BIG DATA

LA INTELIGENCIA ARTIFICIAL HOY Y SUS APLICACIONES CON *BIG DATA*

Amparo Alonso Betanzos
Daniel Peña
Pilar Poncela
(editores)

funcas

BIG DATA

LA INTELIGENCIA ARTIFICIAL HOY Y SUS APLICACIONES CON *BIG DATA*

Amparo Alonso Betanzos
Daniel Peña
Pilar Poncela
(editores)

funcas

Impreso en España
Edita: Funcas
Caballero de Gracia, 28, 28013 - Madrid

ISBN impreso: 978-84-17609-93-1

ISBN digital: 978-84-17609-94-8
Depósito legal: M-5156-2025
Maquetación: Funcas

Contenido

Presentación 1
Amparo Alonso Betanzos, Daniel Peña y Pilar Poncela

Capítulo I. Modelado basado en agentes para la simulación de políticas de sostenibilidad 5
Amparo Alonso Betanzos, Alejandro Rodríguez Arias Bertha Guijarro Berdiñas y *Noelia Sánchez Maroño*

Capítulo II. Inteligencia artificial y cerebro computacional a través de la fusión de datos 31
Humberto Bustince

Capítulo III. Un sistema de inteligencia artificial rápido y eficiente energéticamente 53
José Duato

Capítulo IV. Grandes modelos de lenguaje: ¿de la predicción de palabras a la comprensión? 73
Carlos Gómez-Rodríguez

Capítulo V. Transformación de la movilidad urbana: aplicaciones y perspectivas de la inteligencia artificial 99
Ibai Laña

Capítulo VI. Interpretabilidad con redes bayesianas 115
Pedro Larrañaga

Capítulo VII. Redes bayesianas como modelos generativos: de los juegos a las finanzas 145
Antonio Salmerón

Presentación

En los últimos años, la inteligencia artificial (IA) ha emergido como uno de los temas más debatidos y transformadores en la sociedad. Impulsada por innovaciones como la IA generativa, y sus herramientas generadoras de texto o imágenes (como ChatGPT, o DALL-E), que incorporan también otras posibilidades, como voz o video, la IA ha captado la atención de gobiernos, empresas y ciudadanos por igual. Su potencial de cambio es innegable y, en consecuencia, ha hecho surgir ciertas preocupaciones en diversos ámbitos.

Los gobiernos y las instituciones políticas están luchando por encontrar un equilibrio entre fomentar la innovación impulsada por la IA y proteger a la sociedad de sus posibles riesgos. Por ejemplo, el Parlamento Europeo aprobó en junio de este año 2024 un reglamento pionero[1] para establecer normas armonizadas sobre la inteligencia artificial, un paso clave hacia la regulación de una tecnología en constante evolución. A nivel académico, las universidades se ven también en la obligación de adaptarse rápidamente y establecer normas sobre el uso ético de la IA y no sólo por parte de los estudiantes, que la emplean como una ayuda en sus trabajos, sino también para crear conciencia ética en su uso por parte de los investigadores y profesores[2].

La IA ha llegado para quedarse en la vida cotidiana de los ciudadanos, que cada vez usan más sus herramientas y aplicaciones para planificar y gestionar distintas actividades como, por ejemplo, buscar rutas adecuadas en sus desplazamientos, encontrar contenidos o ítems que satisfagan sus gustos personales, y traducir y componer textos y documentos. Su amplio uso ha propiciado el interés de crear una norma mundial sobre la ética de la IA, con un primer paso dado por la Unesco en noviembre de 2021, con recomendaciones aplicables a sus 194 Estados miembros[3].

Un ejemplo de la importancia actual de la IA es que el Premio Nobel de física se ha concedido este año a John J. Hopfield y Geofrey E. Hinton "por descubrimientos e invenciones fundamentales que permiten el aprendizaje automático con redes neuronales artificiales", y el de química 2024 se ha dividido entre David Baker, por "el diseño de proteínas con computación", y Demis Hassabis y John Jumper, por "la predicción de la estructura de las proteínas mediante el uso de inteligencia artificial".

[1] https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX:32024R1689

[2] https://www.ciencia.gob.es/InfoGeneralPortal/documento/cedef913-3842-4f90-9ffc-032d256125f0

[3] https://unesdoc.unesco.org/ark:/48223/pf0000381137_spa

El término "inteligencia artificial", nacido en el ámbito de las ciencias de la computación, ha evolucionado para abarcar una amplia variedad de aplicaciones automáticas, integrando elementos de ciencia de datos, estadística, matemática aplicada e ingeniería de señales y electrónica. Estas disciplinas se combinan para crear algoritmos capaces de procesar y analizar grandes volúmenes de datos, aprovechando la capacidad de cómputo de los sistemas modernos. Las aplicaciones tradicionales de la IA incluyen tareas como clasificación, predicción y toma de decisiones, pero, gracias a los avances recientes en diversas áreas de la computación, también se han abierto nuevas posibilidades, como la generación automática de imágenes, textos y sonidos a través de modelos de IA generativa.

Este libro explora cómo la IA y el *big data* están transformando todos los sectores, y cómo las aplicaciones de esta tecnología están reconfigurando nuestras vidas, en formas que aún estamos empezando a comprender. Para ello, publicamos aquí una versión ampliada de las ponencias presentadas en las jornadas de las que el libro toma el nombre, que fueron organizadas por los tres editores de esta monografía en Funcas el 16 de octubre de 2024. El lector puede ver en el canal de Funcas en YouTube las presentaciones orales de las contribuciones que se presentan en este libro, que se han organizado en capítulos por orden alfabético de autores. A continuación, describimos brevemente sus contenidos:

En el capítulo I, **Amparo Alonso Betanzos, Alejandro Rodríguez Arias, Bertha Guijarro Berdiñas** y **Noelia Sánchez Maroño** presentan tres ejemplos de modelado basado en agentes inteligentes para prever el resultado de distintas políticas de sostenibilidad en una universidad o en una ciudad. Describen, en primer lugar, la herramienta de modelización basada en agentes, donde se modela el comportamiento de un conjunto de agentes, por ejemplo, personas que actúan dentro de un entorno, un centro, una ciudad o una región, en interacción en red con otros agentes con base en unos objetivos definidos. Esta herramienta se utiliza para analizar el efecto de distintas políticas para disminuir las emisiones en CO2 en la Universidad de la Coruña, para estudiar la aceptabilidad de políticas de innovación social, como las supermanzanas en una ciudad, y para investigar la aceptación por la población de medidas de contención del COVID durante la epidemia.

La inteligencia artificial está teniendo mucha importancia en el campo de la salud y **Humberto Bustince** presenta en el capítulo II algunas herramientas de IA que pueden ayudar a pacientes afectados por problemas neurológicos mediante la neurociencia computacional. Describe algunas investigaciones en este campo para entender cómo nuestro cerebro emite señales para activar nuestro cuerpo y se concentra en cómo manejar los datos disponibles teniendo en cuenta el problema de los errores de medición. Un problema importante es cómo combinar distintas mediciones afectadas por ruido y el autor presenta soluciones desarrolladas dentro de la neurociencia para resolver este problema.

José Duato presenta en el capítulo III un nuevo modelo de aprendizaje automático que, aunque basado en una red lineal, logra comportarse de manera no lineal, emulando las características de las redes neuronales con la conocida función de activación ReLU (*Rectified Linear Unit*).

Antes de describir el modelo, se analizan los avances más relevantes alcanzados en los últimos años en el campo del aprendizaje automático, especialmente aquellos orientados a reducir el consumo energético de los algoritmos, destacando la importancia de este tema en el desarrollo futuro de la inteligencia artificial. Posteriormente, se evalúan las principales ventajas del modelo, como su precisión, interpretabilidad y facilidad para el reentrenamiento, además de destacar una significativa reducción en los tiempos de ejecución, lo que se traduce en un importante ahorro energético.

En el capítulo IV, **Carlos Gómez-Rodríguez** presenta un recorrido histórico sobre la investigación en Procesamiento del Lenguaje Natural (PLN), explicando de manera didáctica cómo se ha llegado al desarrollo de los grandes modelos de lenguaje (*Large Language Models*, LLM), que sustentan herramientas tan populares en la actualidad como ChatGPT, entre otras. A continuación, analiza las utilidades, capacidades y limitaciones de estos modelos, y aborda algunos de los debates más controvertidos relacionados con su uso, como la cuestión de la comprensión del lenguaje desde una perspectiva humana y su elevado consumo energético.

Las aplicaciones de la inteligencia artificial en la movilidad urbana, potenciadas por la amplia disponibilidad de datos, incluso en tiempo real, constituyen el foco del capítulo V, cuyo autor es **Ibai Laña**. En él, se analiza cómo estas técnicas disruptivas permiten mejorar la seguridad vial, optimizar la planificación de rutas, reducir atascos y aumentar la eficiencia en el consumo de combustible, entre otros beneficios. Tras abordar aspectos generales del tema, el autor profundiza en las áreas específicas de uso de la IA en el contexto urbano. Finalmente, se presentan las tendencias tecnológicas más recientes en este ámbito, como las redes neuronales de grafos o la IA generativa, entre otras.

Los dos últimos capítulos abordan diversos aspectos de las redes bayesianas en el contexto de la IA. Una de las críticas, que a menudo se hace a los diversos métodos empleados en IA, es que actúan como cajas negras. No obstante, tanto desde el punto de vista científico como de situaciones de alto riesgo, entender cómo es el proceso en el que se basa la toma de decisiones y qué lleva a decantarse por una u otra elección resulta crucial. Para rebatir lo anterior en el contexto de redes bayesianas, en el capítulo VI **Pedro Larrañaga** aborda su interpretabilidad. El autor explica que estas redes verifican las tres condiciones necesarias para que se puedan considerar como interpretables: simulabilidad, decomponibilidad y transparencia algorítmica. Se ilustra lo anterior mediante el análisis de un caso real en neurociencia computacional.

Finalmente, la monografía se cierra con el capítulo VII de **Antonio Salmerón**, quien ilustra el uso de las redes bayesianas en IA generativa y en modelos de predicción a través de tres ejemplos. En primer lugar, se muestra cómo las redes bayesianas aprenden a jugar al ajedrez, refinando la heurística de búsqueda a medida que la red adquiere experiencia sobre la base de jugadas anteriores. A través del segundo ejemplo se ilustra su uso en genética agrícola. En particular, se emplean redes bayesianas para encontrar la combinación de variedades de tomates que maximizan la probabilidad de generar una nueva variedad con unas determinadas características deseadas. La última aplicación trata el problema de la predicción de la morosidad en créditos particulares. A partir de la información contenida en 44 variables

predictoras, se monitoriza la evolución del riesgo de entrar en mora durante los próximos 12 meses de los clientes en operaciones de crédito.

Los editores queremos agradecer a todos los autores de los capítulos su excelente disposición para prepararlos pensando en un público no necesariamente especialista en el tema. Agradecemos también al director general de Funcas, Carlos Ocaña, el apoyo a esta iniciativa y todas las actividades relacionadas con *big data* y sus aplicaciones. Es un placer contar para la organización de las actividades con Cecilia y Esperanza, que cuidan todos los detalles organizativos con mimo y eficacia, y con Myriam González, responsable de la cuidadosa edición de este libro. Para todos los autores y para el excelente equipo de Funcas nuestro profundo agradecimiento.

Amparo Alonso Betanzos, Daniel Peña y **Pilar Poncela**

Diciembre 2024

CAPÍTULO I

Modelado basado en agentes para la simulación de políticas de sostenibilidad*

Amparo Alonso Betanzos
Alejandro Rodríguez Arias
Bertha Guijarro Berdiñas
Noelia Sánchez Maroño

En este artículo exploramos el uso de modelos basados en agentes (MBA) combinados con técnicas de inteligencia artificial (IA) para simular y analizar sistemas complejos. Los MBA permiten representar sistemas compuestos por múltiples entidades autónomas, cada una con sus propias características y comportamientos, lo cual facilita el estudio de dinámicas emergentes a nivel micro y macro. Al integrar técnicas de aprendizaje automático se potencia la capacidad de los agentes para tomar decisiones adaptativas, analizar datos y mejorar la precisión de las simulaciones. Este enfoque ha demostrado ser particularmente efectivo para resolver problemas donde la interacción entre agentes y su entorno es clave, como en la propagación de enfermedades, la adopción de innovaciones, o la gestión de recursos en organizaciones. Estos modelos son aplicables en diversos dominios, como la sostenibilidad, la salud pública o la economía, y tienen un importante potencial para la simulación de políticas, permitiendo mejorar la toma de decisiones en entornos complejos y cambiantes.

Palabras clave: modelado basado en agentes, aprendizaje automático, simulación de políticas.

* Los sistemas descritos en este capítulo han sido desarrollados en el marco de los proyectos financiados por la Unión Europea: LOw CArbon at Work y SMARTEES (Social Innovation for Modelling Approaches to Realizing Transition to Energy Efficiency and Sustainability), así como del proyecto CEDCOVID (Ciencia e Ingeniería de Datos para la Evaluación, Predicción Poblacional y Personalizada de la Evolución de la Enfermedad COVID-19), financiado por la Xunta de Galicia. Expresamos nuestro más sincero agradecimiento a nuestros compañeros de los distintos proyectos por su valiosa colaboración y el esfuerzo dedicado a estas iniciativas.

1. INTRODUCCIÓN

Los modelos basados en agentes (MBA) son una herramienta clave para simular y estudiar sistemas complejos normalmente integrados en contextos sociales y económicos determinados (Railsback y Grimm, 2019). Estos modelos simulan el comportamiento de agentes y su interacción con el entorno, donde los agentes pueden representar individuos, organizaciones, o incluso ecosistemas, que poseen cierta autonomía, con características y comportamientos individuales, actuando bajo ciertas reglas que determinan sus acciones y decisiones. Esta técnica ha ganado relevancia en diversos campos, debido a que permite la simulación de interacciones a nivel micro y el estudio de sus efectos emergentes a nivel macro. Este enfoque es especialmente valioso en el modelado de políticas sociales o económicas relacionadas con la sostenibilidad, ya que los problemas suelen ser multifacéticos, afectando a los individuos, a la sociedad y al medioambiente. El uso de estos modelos permite a los investigadores y a los responsables de la elaboración de políticas evaluar el impacto potencial de distintas intervenciones en comunidades y sistemas complejos antes de implementarlas en la vida real.

Al incorporar la diversidad de actores, sus decisiones y las interacciones entre ellos, los modelos basados en agentes son unas herramientas para analizar cómo diversos actores (ciudadanos, instituciones, empresas, organizaciones, entre otros) se relacionan dentro de sistemas complejos en contextos socioeconómicos. Esto permite una visión más dinámica y realista del impacto potencial de las políticas implementadas, favoreciendo el diseño de estrategias más efectivas y equitativas para la sociedad.

Una ventaja importante de este método de modelado es su flexibilidad, con capacidad para representar a los individuos mediante sus características particulares, lo que permite abordar los aspectos de heterogeneidad entre individuos o tipos de actores. Además, facilita la inclusión de representaciones estructuralmente complejas, dinámicas y también heterogéneas de exposición o influencia social y ambiental, permitiendo incluir elementos espaciales que afectan a los agentes y a sus interacciones. Otra característica relevante es su capacidad para modelar la interacción entre los elementos del sistema y la adaptación, permitiendo la captura de interacciones entre actores y entornos, como las influencias bidireccionales entre las normas sociales y los comportamientos individuales, entre otros. El MBA permite analizar efectos emergentes a nivel macro que no son fácilmente previsibles a partir de las decisiones individuales. Por ejemplo, en la adopción de energías renovables, el modelo puede representar cómo las preferencias de los consumidores, la disponibilidad tecnológica y las políticas gubernamentales interactúan para influir en los niveles de aceptación de las distintas opciones de consumo. Además, las simulaciones permiten evaluar el impacto de diversas políticas alternativas en plazos medio-largos, y comparar también los resultados de diferentes escenarios en un entorno virtual. De esta manera, se puede observar cómo las variaciones en incentivos, regulaciones o cambios culturales impactan la sostenibilidad. Esto no solo mejora la toma de decisiones, sino que también facilita la identificación de políticas más resilientes, capaces de adaptarse a un entorno global dinámico. Como ejemplo en el ámbito bancario, un MBA podría ser útil al permitirnos capturar la diversidad de perfiles y comportamientos de los clientes, lo cual facilita el análisis de efectos emergentes a nivel macro que no pueden

preverse fácilmente a partir de las decisiones individuales. Así, en el contexto de la concesión de créditos, el modelo puede simular cómo influyen en la toma de decisiones los distintos perfiles de clientes, la situación económica, la competencia y las políticas de riesgo. Además, las simulaciones permiten evaluar el impacto de diferentes estrategias de crédito o cambios regulatorios en horizontes medio-largos y comparar los resultados en diversos escenarios simulados. De este modo, es posible observar cómo los cambios en las tasas de interés, los incentivos fiscales o la educación financiera impactan en la estabilidad y el crecimiento de la cartera de crédito. Esto no solo mejora la toma de decisiones, sino que también ayuda a identificar estrategias de crédito y políticas de riesgo más resilientes, adaptables a fluctuaciones económicas y a un entorno financiero dinámico.

Un importante desafío relacionado con el MBA es la validación de los modelos, cuyo objetivo es garantizar que su funcionamiento se parezca al del sistema real. Este proceso incluye no solo la comparación de los resultados globales del modelo, sino también la evaluación detallada del comportamiento individual de los agentes a nivel micro. Entre los diversos métodos de validación disponibles, uno de los más comunes debido a la falta de suficientes datos de calidad es la validación por expertos (Collins *et al.,* 2024; Balci, 1998). Este enfoque implica presentar el modelo y su comportamiento a especialistas en el campo, quienes analizan y evalúan si los resultados del modelo son consistentes con el conocimiento teórico o empírico del sistema que se está modelando, y confirman su precisión y coherencia con la realidad observada. Por ejemplo, en nuestro caso veremos cómo una vez implementados los modelos, hemos comprobado si éstos son capaces de reproducir de manera adecuada los resultados del proceso de implantación de políticas similares en el pasado. Para ello, hemos contado con la opinión experta de diversos *stakeholders* (Oficina de Medioambiente de la UDC, Concejalía de Medioambiente de Vitoria, etc.).

Por otra parte, la inteligencia artificial (IA) puede contribuir dotando a los agentes individuales de capacidades de aprendizaje, adaptación y toma de decisiones que van más allá de las reglas o ecuaciones del modelado basado en agentes clásico, además de contribuir también con un modelado más realista y complejo de los comportamientos de sociedades y grupos humanos debido a las interacciones más complejas que se pueden modelar usando IA (Campos *et al.,* 2025). Los modelos de aprendizaje automático que se pueden incorporar a los MBA permiten modelar el comportamiento cognitivo de los agentes individuales y sus interacciones con el entorno y con el grupo social considerado, así como su evolución en un cierto intervalo temporal. En resumen, es particularmente útil e interesante cuando el sistema a estudiar implica múltiples agentes que interaccionan de forma compleja, a partir de procesos de toma de decisión individuales, y aparecen propiedades emergentes que es necesario estudiar y que no son fácilmente predecibles o explicables mediante las técnicas tradicionales de simulación. Esta simbiosis entre IA y MBA es relativamente novedosa, y nos permite realizar simulaciones más precisas en áreas como los sistemas económicos y sociales complejos donde se han mostrado inadecuados hasta ahora los métodos de modelado más tradicionales.

En este capítulo veremos tres ejemplos diferentes de modelado de políticas sociales en el área de sostenibilidad, en los que se desarrollarán diferentes modelos MBA con combina-

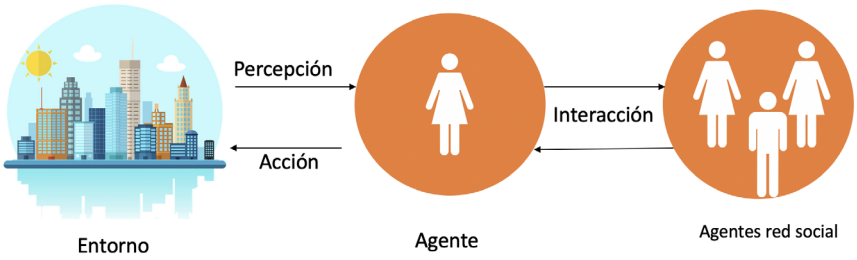
ción de técnicas de IA para modelar la toma de decisiones de los agentes individuales. Hasta ahora, las aproximaciones tradicionales no han sido capaces de tratar la complejidad social de la transición energética. La sostenibilidad es compleja, porque puede percibirse en diferentes contextos, como el medioambiental, el social, el psicológico o el económico, de modo que constituye un problema que involucra la toma de decisiones humanas en entornos multifacéticos que interactúan en formas múltiples. El modelado de procesos relacionados con la sostenibilidad es una tarea especialmente relevante y particularmente urgente en este momento de transición energética en la que nos encontramos. Este enfoque es especialmente valioso en el modelado de políticas sociales, ya que permite a los investigadores y formuladores de políticas evaluar el impacto potencial de distintas intervenciones en comunidades y sistemas complejos antes de implementarlas en la vida real.

2. MODELOS BASADOS EN AGENTES. ESTRUCTURA GENERAL

Los modelos basados en agentes tienen tres componentes principales, como se puede ver en la figura 1:

Figura 1.

Los tres componentes principales de un modelo basado en agentes



Fuente: Elaboración propia.

- *Agentes*, que representan una entidad autónoma con características y comportamientos individuales. Los agentes van a incorporar modelos de comportamiento que podrían estar basados en teorías psicosociales que formarán parte de su modelo de toma de decisiones, pero este modelo también debe reflejar el comportamiento real de los individuos de la organización que se va a modelar. Para ello, se recogerán datos en un cuestionario que será elaborado por psicólogos y sociólogos, y que se enviará a los individuos o entes cuya respuesta se pretende modelar. Se incorporarán también otros tipos de datos procedentes de fuentes disponibles, como pueden ser datos censales, por ejemplo. En los tres ejemplos que vamos a describir, el modelo de toma de decisiones de los agentes necesita ser explícito y transparente, para poder ser revisado por los expertos en el área social, para comprobar su adecuación a las teorías psicosociales empleadas.

- *Red social*, que está formada por individuos o entes con los que se relacionan los agentes. Esta red se ocupa de representar otro aspecto que es imprescindible modelar, la interacción entre agentes. Esta interacción no es necesariamente ni simétrica ni estática, y por tanto puede influir en la toma de decisiones de los agentes individuales. Usualmente, un agente establece una unión o *link* con otros agentes con los que se relaciona (otras organizaciones o individuos como, por ejemplo, amigos, familiares o compañeros de trabajo). La red social de diferentes agentes tiene distintos tamaños, grados de influencia y tipos de evolución. Existen actualmente cuatro tipos básicos de modelos de redes (Van Eck y Jager, 2010): 1) red regular: cada nodo está conectado a sus cuatro vecinos inmediatos; 2) mundo pequeño (*small world*): la mayoría de los nodos están conectados solo con sus vecinos inmediatos; 3) libre de escala: unos pocos nodos tienen muchas conexiones, y 4) aleatoria: la mayoría de los nodos tienen tres o cuatro conexiones.

- *Entorno*, que es el espacio en el que los agentes interaccionan, toman decisiones y realizan acciones, cuyo alcance nos interesa medir.

En los siguientes apartados veremos tres ejemplos diferentes de modelado de políticas sociales. En el primero de los ejemplos modelaremos el efecto de diferentes políticas medioambientales sobre diferentes organizaciones, y cómo mejorar los resultados de estas, usando como medida de evaluación de su efectividad la disminución de las emisiones en CO_2, y un horizonte temporal amplio. El tamaño de la organización no es elevado (alrededor de 2.000 personas), por lo que, en este caso, modelaremos una red social *ad-hoc*. En el segundo modelo, la idea es medir la aceptabilidad de políticas de innovación social en entornos locales, como puede ser el caso de la implantación de modelos urbanos en una ciudad determinada. En este segundo caso, el tamaño del modelo es mucho mayor, ya que pretendemos modelar el comportamiento de una ciudad de mediano tamaño, o de una comunidad concreta (como una isla, por ejemplo), y por lo tanto la red social será elegida como uno de los modelos disponibles en las herramientas de simulación de MBA comunes, como es el caso de NetLogo[1]. Finalmente, en el tercer y último caso modelaremos la aceptación, por parte de la población, de medidas de contención de un virus en una epidemia, y cómo esta cuestión afecta de manera notable a la propagación de esta, ilustrando el interés y la utilidad de este tipo de técnicas que son capaces de modelar sistemas complejos teniendo en cuenta teorías psico-sociales en la toma de decisiones.

3. MBA PARA LA SIMULACIÓN DE LA TOMA DE DECISIONES EN UNA ORGANIZACIÓN. EL PROYECTO LOCAW

Las grandes empresas y organizaciones necesitan modelos cada vez más precisos para monitorizar su estado o cualidades, o bien para simular qué ocurriría si se realizasen ciertos cambios en las mismas. Los MBA son útiles para este propósito, ya que nos permiten modelar

[1] https://ccl.northwestern.edu/netlogo/

directamente a los agentes involucrados (trabajadores, secciones, departamentos...) en su día a día, en lugar de depender de sistemas de ecuaciones extensos y difíciles de interpretar, justificar o explicar. El potencial de los MBA radica en que hacen posible la representación directa de los actores de un sistema social y de sus comportamientos en sus entornos sociales y/o físicos.

Este estudio[2] describe el modelo de toma de decisiones sobre prácticas proambientales de los agentes en una organización, así como el modelo de sus interacciones sociales. En concreto, se detalla cómo se diseñó un MBA para una organización académica, específicamente la Universidade da Coruña (UDC) (Sánchez-Maroño, 2015). La idea es simular el comportamiento de cada trabajador en la organización, teniendo en cuenta sus diferentes grados de autonomía y sus diferencias en valores, asunción de normas, etc. Por comportamiento proambiental nos referimos a decisiones que el trabajador toma en el desempeño de su trabajo, por ejemplo, ¿qué medio de transporte usa para ir a su puesto de trabajo? ¿emplea papel reciclado? ¿apaga las luces al salir del despacho/aula? Por tanto, para desarrollar el modelo, necesitamos tanto un esquema del comportamiento de los agentes que representarán a los trabajadores como de las redes sociales en las que interactúan. Dado el tamaño de la institución, que en ese momento contaba con 2.277 trabajadores entre personal docente e investigador y personal de administración y servicios, el modelo de toma de decisiones de los agentes se construyó en función de las respuestas individuales a un cuestionario elaborado por un equipo de psicólogos y sociólogos. El modelo de toma de decisiones, para el que se utilizaron técnicas sencillas de IA, estaba sujeto a dos restricciones fundamentales: en primer lugar, la decisión concreta del agente debía ser explícita, para que los psicólogos y sociólogos pudiesen comprobar su consistencia teórica, además de que un algoritmo de toma de decisiones transparente es en principio más útil a la organización para facilitar la interpretación de los resultados. La segunda restricción se refiere a que debe simularse la respuesta real de los trabajadores, y de ahí la necesidad de un cuestionario individual. Por otra parte, la estructura de la red social se derivó de un análisis documental, complementado con datos del cuestionario sobre el número de relaciones entre los empleados, ajustando así la información a datos reales. Para representar la relación entre dos agentes, se creaba un *link* entre ellos, de modo que cada agente puede tener varios *links* que forman su red social representada como un grafo, en la que los *links* tienen valores que reflejan la fuerza de la relación, y que pueden cambiar con el tiempo, además de aparecer/desaparecer. Los agentes interactúan mediante esta red social, percibiendo el comportamiento de los demás, quizás modificando el suyo en base a ello e intentando influir en otros agentes con los que están conectados.

3.1. El modelo de toma de decisiones

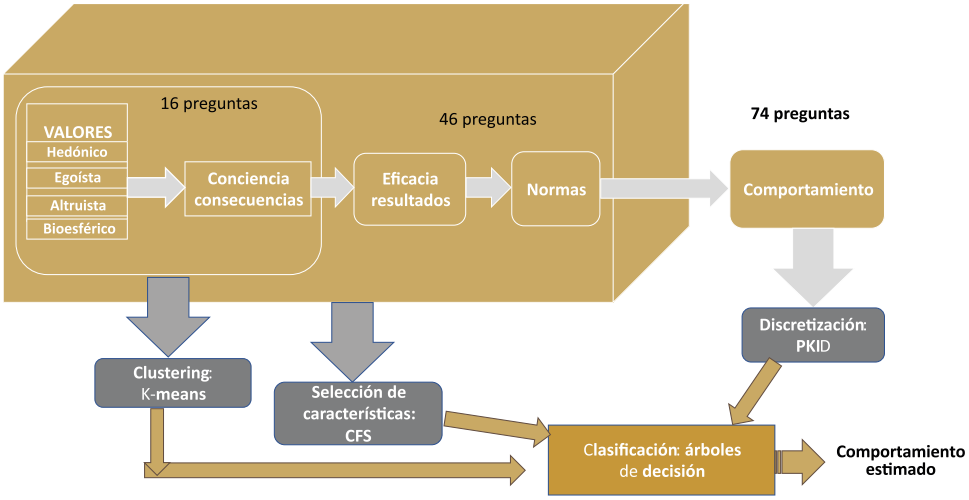
El modelo que se adoptó para modelar el comportamiento proambiental de los agentes es un modelo teórico, que mezcla la influencia de los valores individuales, la conciencia de las consecuencias, la eficacia de los resultados y las normas (Steg y De Groot, 2012). En la

[2] LOw CArbon at Work, 7º Programa marco, http://www.locaw-fp7.com/

figura 2 se muestran las diferentes partes en las que se dividió el cuestionario, así como el número de preguntas relacionado con cada uno de los aspectos involucrados (transmisión de normas, conciencia de las consecuencias, etc). También podemos ver las técnicas (*clustering*, con el clásico *modelo k-means* (Wu, 2012); selección de características, para lo que se empleó el modelo *CFS –Correlation Feature Selection–* (Hall, 1999); y discretización, en la que usamos el modelo *PKID–Proportional k-Interval Discretization–* (Yang *et al.,* 2001)) que se utilizaron para poder obtener los modelos de IA (en este caso, árboles de decisión) que nos permiten estimar el comportamiento individual de cada agente, como se explicará más adelante.

Figura 2.

El modelo de toma de decisiones y su relación con algunas de las preguntas del cuestionario



Fuente: Elaboración propia.

Los valores pueden entenderse como conceptos abstractos o creencias relacionadas con los objetivos de una persona, y sirven como estándares orientadores en su vida. Existen varios tipos, pero en este proyecto se han considerado cuatro orientaciones: egoísta, hedónica, altruista y bioesférica. Por ejemplo, una persona puede reducir el uso del automóvil debido a su coste (egoísta), o porque pone en peligro la salud de las personas (altruista).

En el proyecto LOCAW se emplearon herramientas cuantitativas y cualitativas, como grupos focales y entrevistas, además de un cuestionario basado en el modelo de valores-creencias-normas para adquirir los datos necesarios para desarrollar el modelo. Este cuestionario incluyó bloques de preguntas sobre valores (véase figura 3), motivaciones (eficacia, visiones del mundo y normas) y comportamientos, tanto en el trabajo como en el hogar, para analizar la relación entre ambos contextos. En este último aspecto, se incluyeron 74 preguntas

Figura 3.

Extracto del cuestionario. Preguntas sobre valores*

	Opuesto a mis valores -1	No importante 0	1	2	Importante 3	4	5	Muy importante 6	De máxima importancia 7
1. *Igualdad*: igualdad de oportunidades para todos	●	●	●	●	●	●	●	●	●
2. *Respeto por la tierra*: armonía con otras especies	●	●	●	●	●	●	●	●	●
3. *Poder social*: control de los otros, dominio sobre otros	●	●	●	●	●	●	●	●	●
4. *Placer*: alegría, satisfacción de los deseos	●	●	●	●	●	●	●	●	●
5. *Unidad con la naturaleza*: encajando con la naturaleza	●	●	●	●	●	●	●	●	●
6. *Un mundo en paz*: libre de guerras y conflictos	●	●	●	●	●	●	●	●	●
7. *Riqueza*: posesiones materiales, dinero	●	●	●	●	●	●	●	●	●
8. *Autoridad*: el derecho a liderar o dirigir	●	●	●	●	●	●	●	●	●
9. *Justicia social*: corrección de la injusticia, protección del más débil	●	●	●	●	●	●	●	●	●
10. *Disfrute de la vida*: disfrute de la comida, el sexo, el ocio, etc.	●	●	●	●	●	●	●	●	●
11. *Protección del medio ambiente*: conservación de la naturaleza	●	●	●	●	●	●	●	●	●
12. *Influencia*: Tener impacto sobre personas y circunstancias	●	●	●	●	●	●	●	●	●
13. *Ser de ayuda*: trabajar para el bienestar de los demás	●	●	●	●	●	●	●	●	●
14. *Prevención de la contaminación*: protección de los recursos naturales	●	●	●	●	●	●	●	●	●
15. *Hedonismo*: hacer cosas agradables y placenteras	●	●	●	●	●	●	●	●	●
16. *Ambición*: trabajo duro, aspiraciones	●	●	●	●	●	●	●	●	●

*: Sólo es elegible una de las opciones numéricas.

sobre el uso de energía, materiales, manejo de desechos y el uso de transporte y sus diferentes tipos. Los datos recopilados se usaron para generar los árboles de decisión que explicaban las decisiones ambientales de los agentes en tareas cotidianas. Para mejorar la generalización de estos árboles, se aplicaron técnicas básicas de IA que permiten obtener un conjunto representativo de estos:

- *Clustering.* Fue el método utilizado para poder agrupar las respuestas en los cuatro tipos de valores contemplados en base a las 16 preguntas incluidas en el cuestionario, que fue contestado por 237 personas. Tras aplicar la técnica *k-means*, y en colaboración con los sociólogos y psicólogos, se distinguieron en la UDC seis clústeres diferentes que nos permitían una separación adecuada de los individuos, cuatro de ellos correspondientes a los tipos anteriormente citados; los otros dos son clústeres híbridos, que se correspondían a perfiles bioesférico-altruista y egoísta-hedónico.
- *Selección de características y discretización.* En este modelo tenemos un número alto de comportamientos a modelar (por ejemplo, medio de transporte empleado, uso de papel, calefacción, encendido de luces, etc.), para los que además puede haber más de una pregunta relacionada en el cuestionario. Por este motivo, se decidió usar un método que nos permitiese seleccionar las variables o características más relevantes para cada uno de los comportamientos, y para ello se usó el algoritmo *CFS* (Hall, 1999). El resultado es una matriz que relaciona comportamientos con las entradas que los expertos han considerado teóricamente consistentes. Un ejemplo puede verse en la tabla 1, donde vemos que además de ciertos datos personales, el valor de la variable "localización" afecta al comportamiento de elegir el coche para los desplazamientos, y que algunos valores relacionados con los perfiles altruista y biosférico (respeto a la tierra, igualdad y un mundo en paz) influyen en la adopción de comportamientos proambientales. La muestra también se discretizó para obtener una representación adecuada de los intervalos que se usaron para graduar las respuestas de los individuos.

Tabla 1.

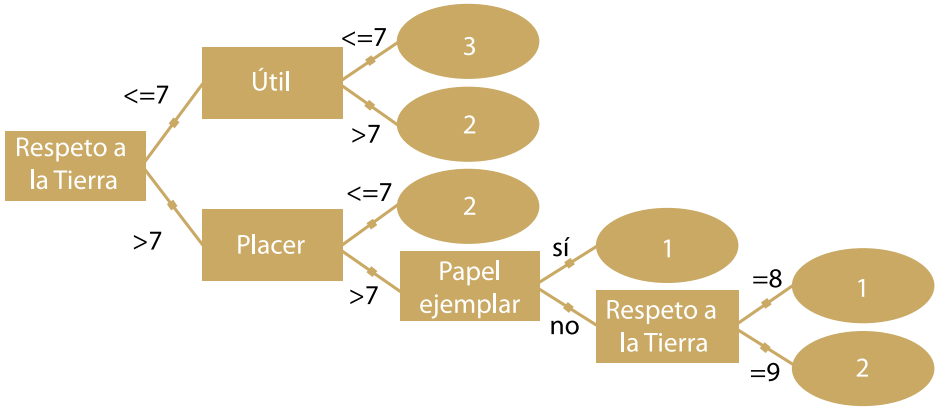
Un extracto de los resultados del proceso de selección de características

Comportamiento	*# Vuelos*	*Uso coche*	*Apagar luces*	*Recicl. papel*
Género	X	X	X	X
Nivel estudio	X		X	X
Nivel organización	X		X	X
Rol ejemplar	X		X	X
Localización		X		
Igualdad			X	
Resp. Tierra			X	X
Paz	X			

- *Clasificación*. Para cada comportamiento, las variables relevantes seleccionadas por el algoritmo *CFS*, junto con el resultado discretizado del paso previo, conforman el conjunto de datos que se utiliza para el entrenamiento del algoritmo C4.5, un algoritmo clásico usado para generar un árbol de decisión para problemas de clasificación (Quinlan, 1993). El conjunto de datos se divide en un 66 % de los datos para entrenar y el 34 % para prueba. El clúster egoísta tiene solo dos muestras en la organización a modelar (UDC), por lo que estas se agregaron al grupo híbrido egoísta-hedónico, según el criterio de los expertos psicólogos. Con estos cinco clústeres y 65 comportamientos a modelar, se generaron 325 árboles de decisión usando C4.5. Para mejorar su rendimiento, algunos árboles se podaron basándose en teorías del campo aportadas por un psicólogo del proyecto. Un ejemplo de uno de estos árboles se muestra en la figura 4. Dado que el número de árboles es muy elevado, las precisiones alcanzadas en el conjunto de prueba varían notablemente, con precisiones que alcanzan el 80,5 % (Sánchez-Maroño *et al.,* 2017). Es resaltable que el problema que tratamos es multiclase, ya que en los cuestionarios las contestaciones de los usuarios tienen asociada una escala de Likert con siete valores diferentes. De ahí la necesidad de usar un algoritmo de discretización, como el *PKID*, que agrupe las respuestas en intervalos concretos que nos permitan mantener una buena precisión teniendo en cuenta que el número de datos (contestaciones al cuestionario) no es demasiado amplio.

Figura 4.

Árbol de decisión para el comportamiento: *¿Con qué frecuencia tienes las luces encendidas en una habitación de casa cuando no hay nadie dentro?*



Nota: Las variables de entrada son las respuestas al cuestionario variando al rango a [1.9]. La salida es la frecuencia en la que se adopta el comportamiento. Como podemos ver, las variables que influyen son el respeto a la tierra del agente, su percepción de la utilidad de la norma de apagar las luces, el placer de disfrutar de las luces encendidas, y si el agente está interesado en mantener un rol ejemplar en la organización.
Fuente: Elaboración propia.

3.2. La red social

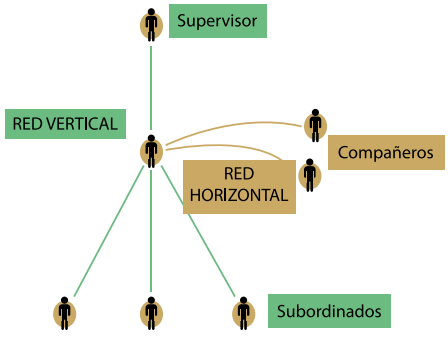
Los agentes interactúan con un subconjunto de otros agentes, formando un grafo en el cual los nodos representan a los agentes y los arcos modelan las relaciones entre ellos. La disposición de estas conexiones se denomina topología de la red social, y describe quién transmite información a quién, además de poder modelar la probabilidad de contacto entre agentes. Aunque existen modelos predefinidos para redes sociales, debido al tamaño manejable del sistema a modelar, se desarrolló un modelo *ad hoc* que incluye dos redes sociales en paralelo, con el objetivo de capturar la complejidad de la organización. Una de las redes refleja la estructura jerárquica de la organización (red vertical), mientras que la otra modela las relaciones entre compañeros (red horizontal). Ambas redes se desarrollaron utilizando el conocimiento de las organizaciones y los trabajadores. La red vertical puede considerarse un modelo libre de escala, ya que pocos agentes (como el rector de la universidad y los directores de departamento) tienen muchos enlaces, mientras que sus subordinados no. En cambio, la red horizontal no se ajusta a estos esquemas, ya que el número de enlaces varía significativamente de un agente a otro y un agente puede estar vinculado a otro que no es necesariamente su vecino. La red jerárquica sirve de base para construir la red horizontal, ya que muchos de los enlaces horizontales se generan a partir de estructuras de departamentos, pero también incluye enlaces "externos" a otros departamentos o incluso empresas, reflejando conexiones espontáneas que pueden surgir en diferentes circunstancias (véase figura 5).

Estas redes facilitan cambios de comportamiento de distintas maneras. Por ejemplo, si el director de un departamento universitario prohíbe imprimir correos electrónicos, esta regla se transmite verticalmente a sus subordinados, lo que implica que no tienen la opción de imprimir. Sin embargo, si el mismo director comienza a reciclar papel colocando un contenedor en un área común (sin que sea obligatorio), algunos investigadores cambiarán su comportamiento por influencia del director a través de la red horizontal, independientemente de si están en el mismo departamento, sólo porque comparten el mismo espacio.

Figura 5.

Las redes sociales en el modelado de la UDC

(a) Esquema conceptual red vertical y horizontal UDC



(b) Esquema red vertical

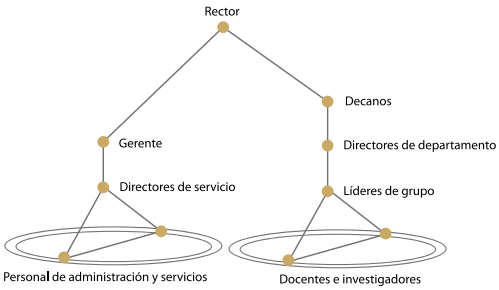
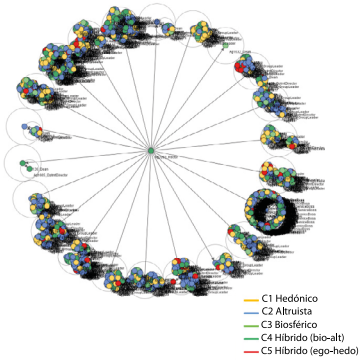


Figura 5. (*continuación*)

Las redes sociales en el modelado de la UDC

(c) Modelo completo red UDC



Nota: En la subfigura (a) se muestra el concepto general de ambas, en la subfigura (b) vemos el esquema general de la red vertical, y en la subfigura (c) vemos ambas redes con sus agentes, usando diferentes colores para los tipos de perfiles de valores de los individuos, como se indica en la etiqueta. En esta subfigura (c) el elemento central es el rector, que tiene conexiones con cada uno de los centros de la Universidad. Estos centros, sus departamentos, grupos, y finalmente los individuos que los forman aparecen como cada uno de los elementos de la rueda. En el caso de los individuos un código identifica su perfil.

Las redes son la estructura que se usa para transmisión de reglas y normas de la organización, tanto a través de la red vertical o jerárquica como de la red horizontal, en la que se puede dar la adopción de comportamientos entre agentes compañeros en la organización. De manera general, los agentes tienen un comportamiento determinado por el árbol de decisión obtenido para el grupo de valores (perfil biosférico, altruista, etc.) al que pertenecen y también por las respuestas del agente a las demás preguntas del cuestionario relacionadas con otros aspectos, como la transmisión de normas. No obstante, cada agente se relaciona con sus compañeros y, como resultado, puede cambiar su comportamiento a lo largo del tiempo. La influencia de los compañeros de un agente variará según las afinidades entre cada par de agentes, de modo que un agente se verá más afectado por aquellos agentes que sean similares a él. El modelo matemático propuesto para la evolución de la red social de los agentes se puede consultar en más detalle en Sánchez-Maroño *et al.,* 2015.

3.3. Simulaciones de diferentes políticas de sostenibilidad

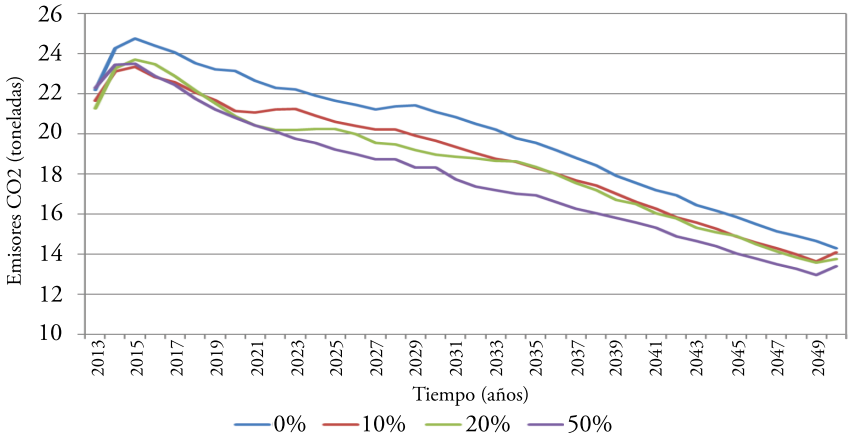
La política de la Unión Europea (UE) tiene entre sus objetivos principales la protección del medio ambiente, además de minimizar los riesgos para el clima, la salud humana y la biodiversidad. El Pacto Verde Europeo aspira a hacer de Europa el primer continente climáticamente neutro del mundo. Con las nuevas regulaciones de la UE, los gobiernos nacionales han aprobado leyes y políticas para reducir o compensar las emisiones de ciertas organizaciones, con el fin de alcanzar los objetivos nacionales y europeos. Estas regulaciones han impulsado a las organizaciones a implementar mecanismos para reducir sus emisiones de gases de efecto invernadero que, sin embargo, hasta el momento, no han logrado las necesarias reducciones.

Para mejorar los esfuerzos hacia la sostenibilidad, es necesario identificar las barreras y los impulsores de cambios sostenibles en las prácticas laborales diarias. El lugar de trabajo, donde se encuentran y negocian las demandas de rentabilidad económica y sostenibilidad ambiental, influye en los hábitos de consumo energético y en las emisiones. Dado que las personas pasan gran parte de su vida en el trabajo, este es un espacio donde se negocian identidades y se promueven o desincentivan comportamientos sostenibles. Utilizando distintas técnicas (grupos focales, entrevistas en profundidad con miembros clave de las organizaciones y encuestas por cuestionario y escenarios de retroproyección) se definieron distintas políticas a implantar en la organización para reducir la huella de carbono cuya efectividad debería ser evaluada mediante el MBA. Las intervenciones diseñadas afectan a las tres partes principales del modelo basado en agentes: los agentes individuales, la red social y el entorno (es decir, la organización). Las políticas abarcaron los tres temas principales (movilidad, energía y residuos) estudiados en el proyecto LOCAW, y fueron evaluadas por separado, en combinación, aisladas en el tiempo y mantenidas en el tiempo para poder obtener conclusiones adecuadas y explorar cómo hacerlas más efectivas. Mostraremos los resultados de algunas de ellas:

- *Incrementar el personal bioesférico.* Dentro de los diferentes perfiles, una posibilidad sería incrementar el más proclive a la sostenibilidad, que sería el personal bioesférico. Asumiendo distintos porcentajes de incremento de ese personal, obtendríamos el resultado que podemos ver en la figura 6. Como se puede apreciar, el descenso en emisiones de CO_2 es prácticamente el mismo para el caso de aumentos del 20 y 10 %. Un incremento importante del 50 % logra resultados algo mejores, pero su efectividad disminuye en el tiempo, debido a la influencia de la red social.

Figura 6.

Simulación de resultados para el caso de que se intente aumentar la proporción de empleados con un cierto perfil, en este caso el bioesférico

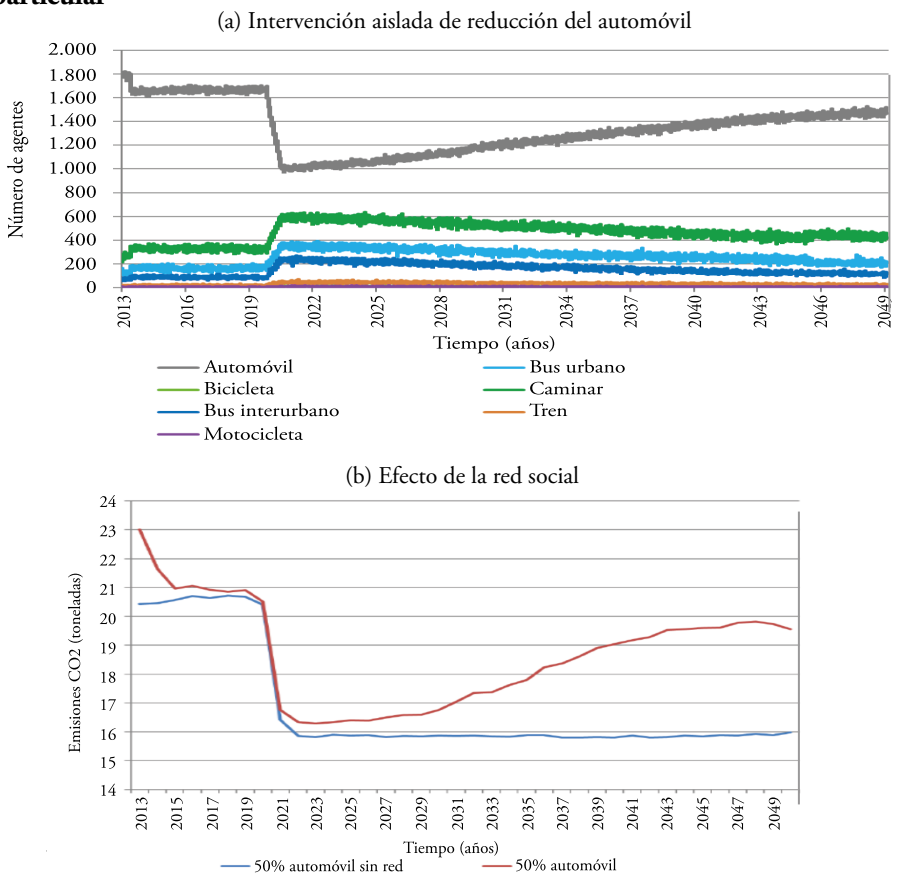


Nota: La situación que corresponde a la etiqueta 0 % es la situación de partida en la organización, y las demás líneas corresponden a aumentos del porcentaje de empleados en el perfil.

- *Reducir el uso del automóvil particular.* En la UDC, el transporte es el aspecto que más perjudica a la sostenibilidad, ya que el 80 % del personal utiliza el automóvil particular para sus desplazamientos. En este caso, se han simulado varias alternativas, que consisten en la reducción del uso del coche, que podría llevarse a cabo de varias formas, por ejemplo, reduciendo la disponibilidad de plazas de aparcamiento o instaurando una política de cobro de espacios de aparcamiento. Las simulaciones se testearon en forma de una acción única aislada en el tiempo, de varias reducciones mantenidas en un tiempo de actuación más largo y, finalmente, mediante la combinación de varias políticas de movilidad, como podría ser incrementar el uso de medios de transporte alternativos, como por ejemplo, favorecer el uso de la bicicleta. En la figura 7 podemos

Figura 7.

Simulación de resultados para el caso de que se reduzca el uso del automóvil particular

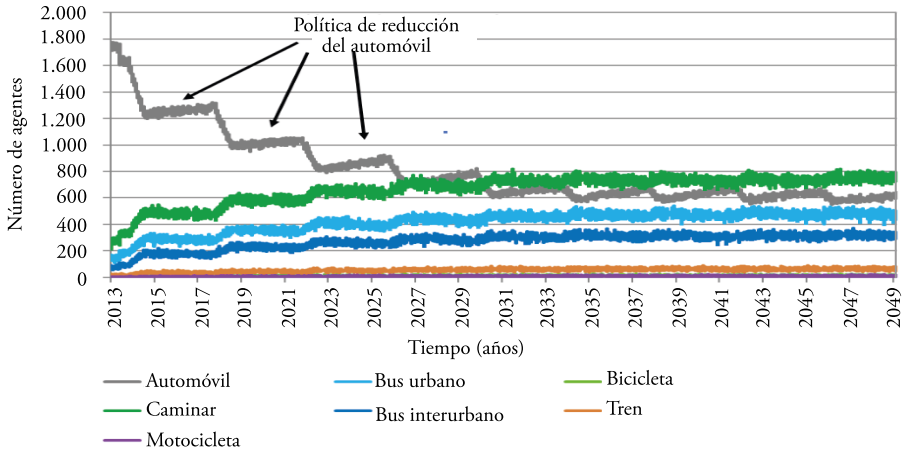


Nota: La subfigura a) muestra el efecto de una intervención aislada para reducir el uso del automóvil particular en el año 2020. La subfigura b) el efecto de la red social.

ver un efecto interesante, que se relaciona con la política de reducción del uso del coche. Como podemos ver, la política en su implantación consigue una reducción de uso del coche (figura 7 superior), pero este se va recuperando con el tiempo por la influencia de la red social (el conocido efecto del boca a boca); prescindiendo de este efecto, la política tendría un efecto perdurable en el tiempo (figura 7 inferior). En cambio, si las políticas de reducción se mantienen en el tiempo y se efectúan de forma periódica, la simulación nos llevaría a que, tras cuatro reducciones en períodos de cuatro años, se conseguiría que el coche dejase de ser la primera opción de transporte, como se puede ver en la figura 8.

Figura 8.

Simulación de resultados para el caso de que se reduzca el uso del automóvil particular (Intervención repetida en el tiempo)



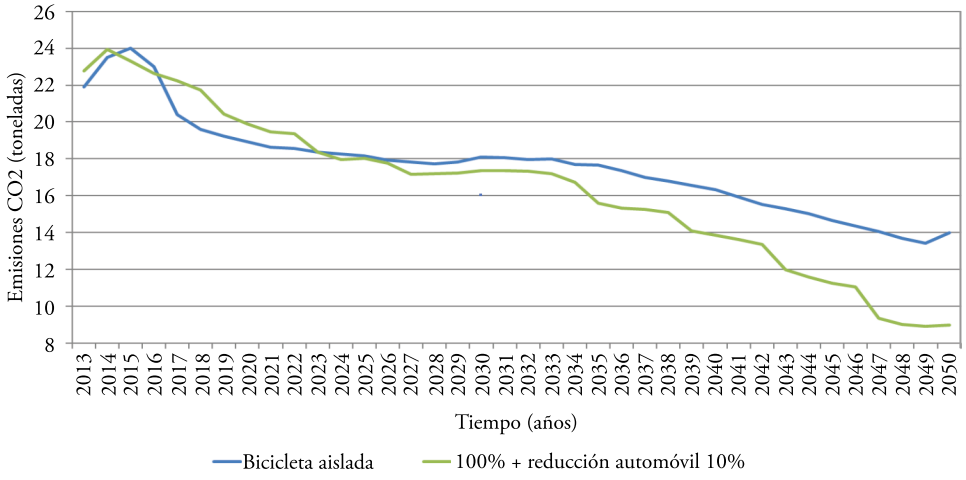
Fuente: Elaboración propia.

Otras opciones se han centrado en estudiar el porcentaje de reducción del uso del coche, o de promoción de otros vehículos, como la bicicleta, obteniendo como resultado que se consiguen reducciones de CO_2 similares con políticas más agresivas que con políticas menos agresivas (por ejemplo, con una reducción de uso del 30 % o del 50 %), y que los mejores resultados se obtienen combinando políticas de distintos tipos, como por ejemplo reducir el uso del coche y promover el uso de la bicicleta (véase la figura 9).

En conclusión, se obtuvo que las políticas mantenidas en el tiempo obtienen mejores resultados que las intervenciones aisladas, y que además es recomendable la combinación de varias políticas, ya que se obtienen mejores resultados que aplicando cada una de ellas separadamente. Para los responsables del diseño de políticas, es también resaltable la importancia de medir adecuadamente la intensidad de estas, ya que las políticas de intensidad media pueden funcionar mejor que las políticas más agresivas para conseguir una transición a la soste-

Figura 9.

Simulación de resultados para incentivar la bicicleta con y sin incentivar la reducción del automóvil particular



Fuente: Elaboración propia.

nibilidad con más éxito. Finalmente, hay que tener en cuenta la influencia de las redes sociales a largo plazo, en especial en organizaciones como las universidades, en las que las redes horizontales tienen más influencia que las verticales. Esta situación no se dio en el modelado de otras organizaciones de corte más jerárquico (Sánchez-Maroño *et al.,* 2014).

4. MBA PARA LA SIMULACIÓN DE LA ACEPTABILIDAD DE POLÍTICAS DE INNOVACIÓN LOCAL. EL PROYECTO SMARTEES

En este nuevo ejemplo[3], el objetivo es estudiar la evolución de la aceptabilidad ciudadana de ciertas innovaciones sociales, con el objetivo de mejorarla y estudiar su escalamiento y replicabilidad. En este caso modelaremos la aceptación de la implantación de supermanzanas en las ciudades. Las supermanzanas pueden definirse como una zona urbana cerrada al tráfico, donde las vías interiores son exclusivas para residentes, transporte público, bicicletas y vehículos de emergencia. La adopción con éxito de este tipo de innovaciones depende en gran medida de la aceptación y la participación de la población en su desarrollo e implementación. En este caso concreto, la población sobre la que se trabaja es mucho mayor que en el ejemplo anterior, ya que necesitamos modelar el comportamiento de una ciudad o comuni-

[3] Proyecto SMARTEES, Social Innovation Modelling Approaches to Realizing Transition to Energy Efficiency and Sustainability. https://local-social-innovation.eu/

dad, por lo que tendremos dos tipos de red social, una red basada en el concepto de círculo social, que utilizaremos como red de vecinos, y una red de amigos formada utilizando una red aleatoria. En cuanto a los agentes, tendremos como principal agente a los ciudadanos, pero en el entorno también consideramos las interacciones que éstos tienen con otro tipo de agentes que llamaremos nodos críticos, y que representan entidades como el ayuntamiento, la prensa, o diversas asociaciones y organismos que toman parte en el proceso, como pueden ser asociaciones de vecinos. Al igual que en el caso anterior, los datos para el MBA se obtendrán a partir de datos disponibles en los ayuntamientos y comunidades, como por ejemplo los datos censales, y también de cuestionarios que se realizan a los ciudadanos. Usaremos como caso base el modelado de la aceptación del modelo de supermanzanas en Vitoria-Gasteiz, que implementó de manera pionera el modelo, para estudiar su posible ampliación a diferentes zonas de esta misma ciudad, y para estudiar su replicabilidad a otros entornos, como es el caso de Poblenou en Barcelona.

Vitoria-Gasteiz desarrolló un Plan de Movilidad y Espacio Público, que incluyó la implementación de la supermanzana central en varias fases, comenzando en 2006 y terminando en 2013. Durante esta fase, se registraron las acciones del Ayuntamiento, las noticias publicadas en los medios y las reacciones de la ciudadanía. Esta información es clave para calibrar y validar un modelo basado en agentes.

El modelo simula cómo interactúan diferentes actores del proceso, como vecinos, asociaciones de comerciantes, ciclistas, y el propio Ayuntamiento, utilizando los datos recopilados durante la implementación real. También cómo varían de opinión los vecinos como consecuencia de estas interacciones. Una vez ajustado, el modelo permite predecir los efectos de distintas políticas, ayudando a identificar las estrategias que podrían provocar una mayor aceptación ciudadana del proyecto de implantación de supermazanas.

Este enfoque ayuda a la administración a planificar mejor la creación de nuevas supermanzanas en otras zonas de la ciudad. Además, con los datos adecuados (como cuestionarios o estadísticas demográficas), el modelo puede adaptarse para simular la implantación de supermanzanas en otras ciudades, optimizando el proceso en diferentes contextos.

4.1. El modelo de toma de decisiones

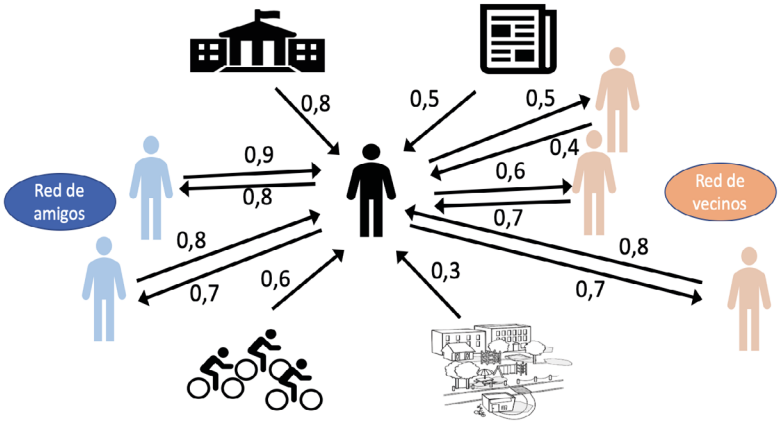
Los agentes ciudadanos son el elemento central en el modelo, y están caracterizados en función de sus variables sociodemográficas y de otras variables necesarias para su modelado. En concreto, el modelo psicosocial que se ha utilizado en este caso es el modelo de toma de decisiones HUMAT (Antosz, 2019). HUMAT considera tres tipos básicos de necesidades, a las que cada agente dará su importancia: las necesidades experienciales (que se refieren al confort y a aspectos económicos), las necesidades de pertenencia (es decir, de sentirse parte de un grupo), y los valores (que se refieren a metas sociales o bioesféricas). Cada agente toma una decisión sobre si aceptar o no la innovación social concreta en función de cómo satisface

sus necesidades esa aceptación y de la influencia que sobre él ejercen los demás agentes de la red social, que son agentes que pertenecen a su red de vecinos o amigos.

Entre los agentes ciudadanos las interacciones que tienen lugar son comunicaciones bidireccionales y en las que el modelo sopesa la confianza que un agente tiene en el otro, que no es necesariamente un valor simétrico. Sin embargo, las interacciones con los nodos críticos son únicamente unidireccionales, van desde el nodo crítico a los ciudadanos, y reproducen un plan de comunicaciones que el nodo crítico ha llevado a cabo en un tiempo determinado, con cierta cobertura e intensidad, y con una intención a favor o en contra de la innovación social. Ante estas comunicaciones, el agente ciudadano evalúa sus alternativas de comportamiento con el modelo HUMAT. Pero es posible que la alternativa de comportamiento (aceptar o rechazar la innovación social) elegida por el ciudadano no esté alineada con todas sus necesidades, de modo que puede generar ciertas disonancias cognitivas asociadas con sus necesidades individuales, que intentará solucionar a través de interacciones con su entorno. La figura 10 ilustra esquemáticamente esta red de interacciones, mostrando los distintos agentes del modelo y sus enlaces, para los que los pesos entre 0 y 1 representan los niveles de confianza entre ellos. El funcionamiento detallado de esta red se encuentra descrito en Rodríguez-Arias (2024).

Figura 10.

El esquema representa las comunicaciones entre cada agente ciudadano del sistema y los agentes de su red de vecinos y amigos, así como con los nodos críticos



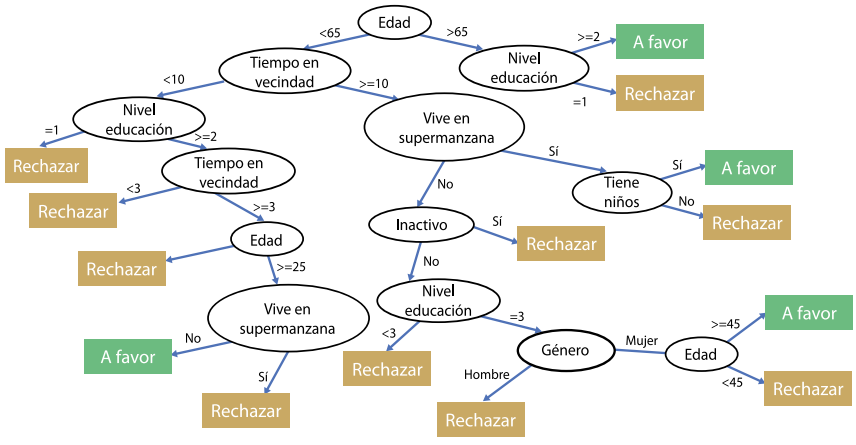
Nota: Los agentes ciudadanos se representan mediante siluetas humanas, mientras que los nodos críticos se representan con imágenes identificativas. Los enlaces indican las redes sociales que conectan a los ciudadanos, cada uno acompañado de un valor numérico entre 0 y 1 que refleja el nivel de confianza del agente origen hacia el agente destino señalado por la flecha.

El modelo representa una población mucho más grande y heterogénea que en el ejemplo anterior. Para la recogida de datos de ciudadanos se utilizó un cuestionario que fue elaborado por un grupo de psicólogos y sociólogos, en el que se recogían variables sociodemográficas (como por

ejemplo edad, género, número de amigos, confianza en ellos, lugar de residencia, nivel de estudios, etc.) y variables referidas a las necesidades de los ciudadanos en relación al modelo a usar (importancia de la calidad del aire, de la salud, de la seguridad, importancia del confort como sería el caso de espacios de aparcamiento y coste, interés en la reputación de la ciudad, etc.). El número de cuestionarios completos recibidos fue alto (cerca de 900 respuestas), y cada uno de los cuales configuró un agente en el sistema. Pero claramente, el número es insuficiente para reproducir la población y se optó por generar agentes que reprodujesen ciudadanos con diferentes perfiles, teniendo en cuenta los datos censales y los perfiles de ciudadanos "sintéticos" que podemos obtener mediante árboles de decisión entrenados con las respuestas reales al cuestionario, en función de la decisión final de los ciudadanos sobre si aceptar o rechazar la innovación planificada, en este caso la supermanzana. Este proceso está descrito en profundidad en Alonso-Betanzos *et al.* (2021). Como indicamos, los árboles de decisión se utilizan para crear perfiles de ciudadanos sintéticos a partir de datos reales. Supongamos que en una zona de la ciudad viven *n* personas, pero solo una parte de ellas, *m*, ha respondido a un cuestionario. Con la información de estos *m* ciudadanos, construimos un árbol de decisión que identifica las características que influyen en su decisión de aceptar o rechazar la supermanzana. Por ejemplo, el árbol puede mostrar que las personas mayores de 65 años, con más de 10 años viviendo en la zona y un nivel de educación básico, tienden a rechazar la supermanzana (como se ilustra en la figura 11). Luego, utilizando los datos censales de las personas que no respondieron al cuestionario ($m - n$), aplicamos el árbol para estimar cómo decidirían en función de sus características. Estos árboles fueron evaluados por los expertos sociólogos para confirmar que su precisión en clasificación resultaba lo suficientemente alta para considerar plausibles las conclusiones obtenidas. En la figura 11 se muestra el árbol que nos permite crear agentes "sintéticos" que reproducen comportamientos de la población real de acuerdo con las variables demográficas y psicosociales que se les asignen.

Figura 11.

Ejemplo de árbol de decisión para la creación de una muestra suficientemente amplia de la población



Fuente: Elaboración propia.

Una vez que se dispone del modelo, se reconstruyen todas las comunicaciones públicas que, por parte de los nodos críticos y con el fin de informar de los cambios y avances del proyecto de supermanzanas, tuvieron lugar en Vitoria-Gasteiz durante la implementación del primer proyecto, y se trasladan al modelo de acuerdo con los parámetros afectados por dichas comunicaciones (a favor, en contra, amplitud, duración, dirigida a confort, dirigidas a salud, etc.). La idea es poder reproducir el proceso pasado lo más fielmente posible, para calibrar el modelo de forma adecuada antes de realizar simulaciones de escenarios alternativos. En primer lugar, se lleva a cabo una calibración local que elimina combinaciones de parámetros del modelo que generan simulaciones significativamente alejadas del proceso real. La validación del modelo se realiza mediante talleres con representantes de las entidades clave, identificadas como nodos críticos. Durante estos talleres, se presentan diversas simulaciones a los expertos, quienes las evalúan y filtran hasta seleccionar aquella que mejor representa la realidad. Una vez que se logró un modelo estable que, en opinión de los expertos, parecía reproducir fielmente lo ocurrido en el proceso original, se pasó a introducir en el mismo nuevas políticas que podrían mejorar la aceptabilidad.

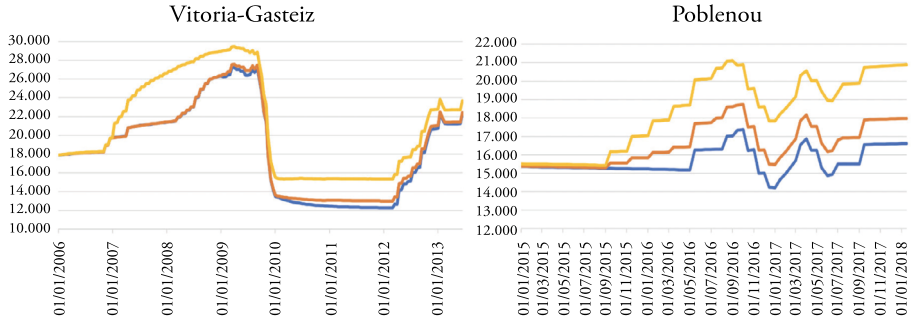
4.2. Simulaciones de nuevas políticas de comunicación a la ciudadanía

En este caso, las personas responsables de diseñar las políticas de comunicación con los ciudadanos querían utilizar el MBA para evaluar el impacto de diversas políticas alternativas que pudieran mejorar la aceptación ciudadana de sus proyectos. La idea era explorar diferentes enfoques e intensidades en las comunicaciones con los ciudadanos para identificar las estrategias más efectivas. En el primer ejemplo, se analiza un escenario alternativo para un evento que ocurrió en noviembre de 2009, en el que la aceptabilidad del proyecto de supermanzanas en Vitoria-Gasteiz descendió considerablemente debido al impacto negativo de una nueva política de aparcamiento comunicada por el ayuntamiento con una estrategia que resultó inefectiva y se centraba en los beneficios de la movilidad sostenible y en el aumento de conciencia medioambiental. Como estrategia alternativa consideramos una estrategia de comunicación en la que el foco es abordar las necesidades específicas de los residentes, en dos escenarios alternativos. El escenario A refleja una campaña de comunicación que se enfoca al confort de los residentes, y el escenario B es un escenario con una campaña más agresiva que se anticipa al rechazo de los residentes, aumentando el alcance de población de la campaña original realizada por el ayuntamiento, pasando de un alcance bajo a uno alto, y donde se extiende además el período de comunicación, y la frecuencia de los mensajes duplicando su número y manteniendo al ayuntamiento activo durante todo el período. Como podemos ver en la figura 12, el escenario 1A apenas mejora los resultados de aceptabilidad de la situación original en Vitoria-Gasteiz, mientras que, en el caso de Poblenou, en el que se ha desarrollado un MBA análogo, la mejora es mayor. El escenario 1B mejora la aceptabilidad aún más en el caso de Poblenou, y en el caso de Vitoria se produce una mejora algo mayor que con el escenario 1A.

Si en cambio optamos por una estrategia orientada a una sensibilización medioambiental y a un entorno ciudadano más participativo, obtenemos los resultados que se muestran en la figura 13. Como vemos, esta estrategia no es útil en absoluto en Poblenou, en donde incluso con

Figura 12.

Simulación del nivel de aceptación del proyecto con la implantación de políticas de comunicación diferentes a la situación original (en azul) para Vitoria-Gasteiz y Poblenou

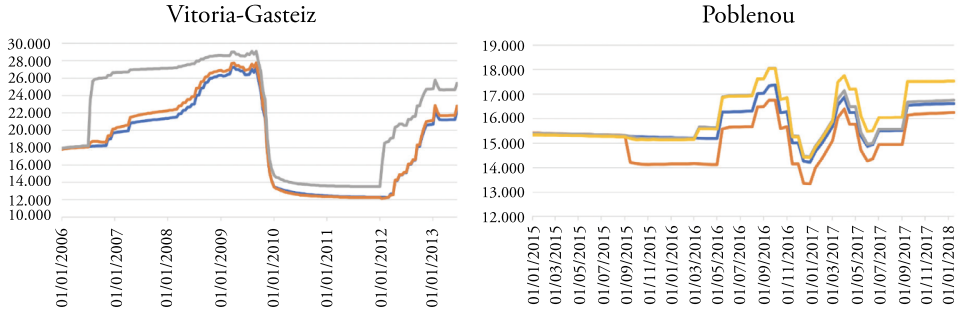


	Vitoria - Gasteiz	Poblenou
Línea de base (azul)	66.30 ± 0.32	60.90 ± 0.31
Escenario 1A (naranja)	66.86 ± 0.32	65.94 ± 0.39
Escenario 1B (amarillo)	70.59 ± 0.33	76.59 ± 0.30

Nota: En el eje x se representa el tiempo, y en el eje y el número de agentes que aceptan la política. El escenario 1A (naranja) enfoca la comunicación hacia favorecer las necesidades de confort, y en el 1B (amarillo) se mantiene el enfoque, pero se intensifica notablemente la estrategia de comunicación.

Figura 13.

Simulación del nivel de aceptación del proyecto con la implantación de políticas de comunicación diferentes a la situación original (en azul) para Vitoria-Gasteiz y Poblenou



	Vitoria - Gasteiz	Poblenou
Línea de base (azul)	66.30 ± 0.32	60.90 ± 0.31
Escenario 2A (naranja)	67.88 ± 0.16	59.57 ± 0.29
Escenario 2B (amarillo)	75.71 ± 0.18	61.44 ± 0.42
Escenario 2C (amarillo)	-	64.27 ± 0.70

Nota: En el eje x se representa el tiempo, y en el eje y el número de agentes que aceptan la política. El escenario alternativo se enfoca a una estrategia de sensibilización ambiental, y a una aproximación participativa de los ciudadanos, intensificando las campañas en los escenarios B y C en cuanto a la estrategia de comunicación.

una intensificación aún mayor que en el caso anterior, la mejora es marginal. Sin embargo, en este caso la estrategia obtiene mejores resultados que la anterior para Vitoria-Gasteiz.

Como conclusión, a pesar de que el MBA desarrollado es exportable a otros entornos ciudadanos, nos permite captar las diferencias entre las diferentes poblaciones y buscar las políticas más adecuadas a cada caso.

5. MBA PARA LA SIMULACIÓN DE LA EXPANSIÓN DE UN VIRUS EN FUNCIÓN DE LA ACEPTACIÓN DE LAS NORMAS DE CONTENCIÓN. EL PROYECTO CEDCOVID

En este último ejemplo[4], veremos cómo modelar la aceptación de las normas por parte de una población en una epidemia puede ayudar en gran parte a la contención de esta. De hecho, varios estudios científicos (Horton, 2020), subrayan la importancia de la confianza de los ciudadanos en sus gobiernos, así como la solidaridad ciudadana, para explicar las diferencias entre los contagios y la mortalidad en los diferentes países durante la reciente pandemia de COVID-19.

Por lo tanto, en este tipo de estudios de propagación de un virus, además de la evolución que se extrae del modelo epidemiológico básico, es de vital importancia prever el comportamiento de los individuos frente a posibles medidas adoptadas para mitigar esta expansión viral. El principal problema de los modelos epidemiológicos clásicos es que la única información que define a un individuo es la relacionada con su estado epidemiológico, por lo que todos aquellos que se encuentran en el mismo estado se consideran idénticos (Korolev, 2020; Menda *et al.,* 2021). Para abordar el problema, nuestra propuesta es integrar un modelo epidemiológico clásico, en este caso el *SEIRD –Susceptible, Exposed, Infectious, Recovered and Dead en inglés–* (Anderson y May, 1992), con un MBA, de forma que cada agente en el modelo represente a un individuo con sus propias características, tanto en términos de variables sociodemográficas como de su estado epidemiológico. Las necesidades psicológicas de cada individuo y la importancia que les otorga determinan, en gran medida, si cumplirá o no con las medidas impuestas. De esta manera, ambos modelos se complementan por una parte tendríamos la propagación del virus en el modelo *SEIRD*, y por la otra el MBA nos permite incluir propiedades individuales que pueden afectar la expansión viral, desde acciones individuales hasta rasgos genéticos específicos.

Dado que los psicólogos y sociólogos consideran que la aceptación de las normas y políticas a aplicar en una pandemia dependen también de que se cumplan las necesidades de los individuos, se ha decidido adaptar el modelo HUMAT, que hemos descrito en el apartado anterior, para la toma de decisiones de los agentes, con ciertos cambios menores. En primer lugar, los agentes son individualizados utilizando las características sociodemográficas (como género, edad, tipo de familia, si es un trabajador esencial, si reside en el campo, salario, etc.), algunas de las cuales están relacionadas con aspectos de las políticas que se implementaron (los trabajadores esenciales debían seguir en su centro de trabajo, por ejemplo). Además,

[4] Ciencia e ingeniería de datos para la evaluación, predicción poblacional y personalizada de la evolución de la enfermedad COVID-19, https://citic.udc.es/proyectos-id/?proyectoId=1062

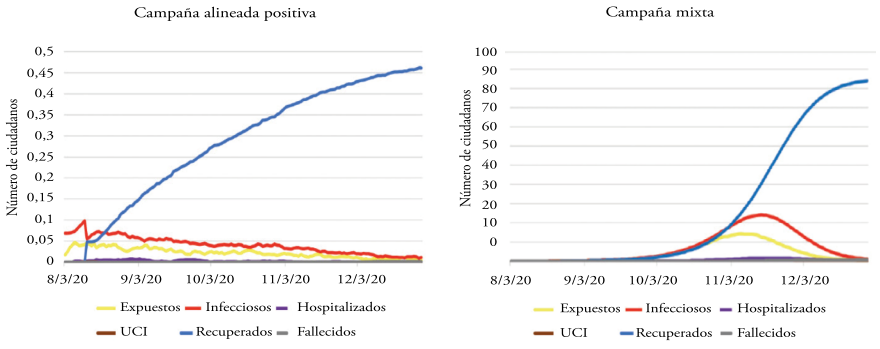
los psicólogos juzgaron como aspectos relevantes para la caracterización individual de los agentes la representación de sus necesidades, como su necesidad de bienestar y confort, así como la necesidad de pertenencia al grupo, que define la necesidad del agente de ser similar a otros miembros de su círculo y que tendrá un peso especial en las acciones que realice el agente. Finalmente, también se tendrán en cuenta las necesidades de interacción social de los agentes. El modelo concreto se describe en mayor detalle en (Rodríguez-Arias *et al.,* 2023). Como nodos críticos se han incluido el gobierno, la prensa, las asociaciones de empresarios y los partidos políticos.

Como ejemplo para las simulaciones hemos elegido la evolución de la pandemia de la COVID-19 en la ciudad de A Coruña. De nuevo, el equipo de psicólogos y sociólogos elaboró un cuestionario que contestaron casi 1.300 personas, cada una de las cuales se modeló como un agente en el MBA. Además, análogamente al caso del modelo en SMARTEES se crearon agentes que representan ciudadanos "sintéticos" creados a partir de los perfiles que se derivan de los árboles de decisión que se construyen sobre las respuestas de ciudadanos reales. Es importante recordar que los árboles actúan como punto de partida para el comportamiento de los agentes "sintéticos", simulando inicialmente el comportamiento de un ciudadano real correspondiente a su mismo nodo hoja, como se explicó en el modelo anterior. Sin embargo, este comportamiento evoluciona a lo largo de la simulación utilizando el modelo HUMAT, de manera similar a como ocurre con los agentes reales".

Como conclusiones generales, cabría destacar que los resultados de las simulaciones indican que la mortalidad hubiese sido mucho mayor sin medidas de contención del virus, y también que el resultado de la evolución es similar en los escenarios de confinamiento completo y en los de medidas menos agresivas, como el uso de mascarillas, distancia social, etc.

Figura 14.

Simulación que compara la evolución del número de ciudadanos en cada estado considerado en el modelo SEIRD según la intención de los mensajes (positivos o negativos) recibidos por los ciudadanos



Esta conclusión es análoga a la del caso del ejemplo del proyecto LOCAW, en el que vimos que se obtienen iguales o incluso mejores resultados con políticas menos agresivas, ya que los agentes son más proclives a cumplirlas. Como aspecto más específico de este caso, se realizó una simulación en la que los nodos críticos estaban alineados en cuanto al tipo de mensaje a enviar, mientras que en contraposición en otra simulación había ciertos actores en desacuerdo, propagando noticias negativas sobre las medidas (como la peligrosidad e ineficacia de las vacunas, remedios no científicos o inexistencia de la pandemia). Como podemos ver en la figura 14, la evolución es mucho mejor en el escenario alineado positivo que en negativo.

6. CONCLUSIONES

En conclusión, el modelado basado en agentes en inteligencia artificial se perfila como una herramienta potente y muy versátil para poder analizar y entender sistemas complejos. Este enfoque permite la creación de simulaciones detalladas de interacciones entre agentes autónomos, que están situados en un entorno específico que podemos también simular, dando lugar a la aparición de comportamientos emergentes que surgen de las interacciones entre esos agentes, y que serían muy complejas de apreciar en otros modelos de simulación. Una opción interesante que hemos visto en los tres casos que hemos detallado es que con estos modelos es posible la generación de datos sintéticos que replican patrones de comportamiento observados en sistemas reales. En muchas áreas, como sería el caso de la economía, esta capacidad es particularmente valiosa porque facilita la exploración de posibles escenarios y el ensayo de políticas en un entorno controlado, sin riesgos reales para el sistema. Además, el modelado basado en agentes aumenta la libertad en la toma de decisiones, ya que permite adaptar los agentes a distintos perfiles psicológicos y sociales, obteniendo resultados que consideran factores humanos profundos y no meramente racionales.

Esta técnica abre nuevas posibilidades para ecosistemas sociales y económicos al permitir no solo predecir comportamientos futuros sino también evaluar el impacto de innovaciones, políticas y cambios regulatorios de manera mucho más precisa. En última instancia, los modelos basados en agentes ofrecen una visión integral y fundamentada de las dinámicas sistémicas, lo que los convierte en una herramienta de gran valor para quienes buscan aplicar la inteligencia artificial no solo en la ciencia y la tecnología, sino también en la toma de decisiones estratégicas en sectores como el financiero, el social o el ambiental.

Referencias

Alonso-Betanzos, A., Guijarro-Berdiñas, B., Rodríguez-Arias, A., y Sánchez-Maroño, N. (2021). Generating a synthetic population of agents through decision trees and socio- demographic data. En *International Work-Conference on Artificial Neural Networks* (128-140). Cham: Springer International Publishing.

Anderson, R. M., y May, R. M. (1992). *Infectious Diseases of Humans: Dynamics and Control.* Oxford University Press.

Antosz, P., Jager, W., Polhill, G., Salt, D., Alonso-Betanzos, A., Sánchez-Maroño, N., Guijarro-Berdiñas, B., y Rodriguez, A. (2019). *Simulation model implementing different relevant layers of social*

innovation, human choice behaviour and habitual structures. https://local-social-innovation.eu/fileadmin/user_upload/Deliverables/SMARTEES-D7.2_Simulation_model_DR1.pdf

Balci, O. (1998). Verification, validation, and testing. *Handbook of simulation, 10*(8), 335-393.

Campos, P., Rao, A., y Joaquim, M. (2025). *Machine-Learning perspectives of agent-based models: Applications to economic crises and pandemics with Phyton R, NetLogo and Julia.* Springer International Pub.

Collins, A., Koehler, M., y Lynch, C. (2024). Methods that support the validation of agent- based models: An overview and discussion. *Journal of Artificial Societies and Social Simulation, 27*(1).

Hall, M. A. (1999). Correlation-based feature selection for machine learning, Ph.D. Thesis, Permanent. https://hdl.handle.net/10289/15043

Horton, R. (2020). Offline: Science and the breakdown of trust. *The Lancet, Vol. 396*, Issue 10256, 945.

Korolev, I. (2020). Identification and estimation of the SEIRD epidemic model for COVID-19. *J. Econom., Vol 220*(1), 63-85. doi: 10.1016/j.jeconom.2020.07.038. Epub 2020 Jul 30. PMID: 32836680; PMCID: PMC7392128.

Menda, K., Laird, L., Kochenderfer, M. J. *et al.* (2021). Explaining COVID-19 outbreaks with reactive SEIRD models. *Scientific Reports, vol. 11*, 17905. https://doi.org/10.1038/s41598- 021-97260-0

Quinlan, J. Ross. (1993). *C4.5: programs for machine learning.* Morgan Kaufmann Publishers.

Railsback, S. F, y Grimm, V. (2019). *Agent-based and individual-based modeling: A practical introduction*, 2nd edition. Princeton University Press.

Rodríguez-Arias, A., Alonso-Betanzos, A., Guijarro-Berdiñas, B., y Sánchez-Maroño, N. (2023). *Agent-Based Model: Simulating a Virus Expansion Based on the Acceptance of Containment Measures.* https://arxiv.org/pdf/2307.15723

Rodríguez-Arias, A., Sánchez-Maroño, N., B. Guijarro-Berdiñas, B., Alonso-Betanzos, A., Lema-Blanco, I, y Dumitru, A. (2024). An agent-based model to simulate the public accep- tability of social innovations. *Expert Systems*, e13731, https://doi.org/10.1111/exsy.13731

Sánchez-Maroño, N., Alonso-Betanzos, A., Fontenla-Romero, O., Brinquis-Núñez, C., Polhill, J. G., y Craig, T. (2015). Influence of internal values and social networks for achieving sustainable organizations. *Proceedings of the Twenty-First European Conference on Artificial Intelligence (ECAI 2014)*, 1179–1185. IOS Press.

Sánchez-Maroño, N., Alonso-Betanzos, A., Fontenla-Romero, O., Brinquis-Núñez, C., Polhil, J. G., Craig, T., Dumitru, A, y García-Mira, R. (2015). An agent-based model for simulating environmental behavior in an educational organization. *Neural Processing Letters, Vol. 42*, 89-118. DOI 10.1007/s11063-014-9390-5

Sánchez-Maroño, N., Alonso-Betanzos, A., Fontenla-Romero, O., Polhill, J. G., y Craig, T. (2017). Empirically-derived behavioral rules in agent-based models using decision trees learned from questionnaire data. *Agent-Based modeling of sustainable behaviors* (53-76). Springer. DOI 10.1007/978-3-319-46331-5

Steg, L., y De Groot, J. I. (2012). *Environmental values. In: The Oxford handbook of environ- mental and conservation psychology.* Oxford University Press.

Van Eck, P., y Jager, W. (2010). Social network structures in agent-based modelling: finding an optimal structure based on survey data (or finding the network that does not exist). *Proceedings of the 3rd World Congress on Social Simulation WCSS2010, Kassel, Germany. 2010.*

Wu, J. (2012). *Advances in K-means clustering: a data mining thinking.* Berlin, Alemania: Springer-Verlag, ASIN: B010DPZNA8.

Yang, Y., y Webb, G. I. (2001). Proportional k-Interval Discretization for Naive-Bayes Classifiers. En L. De Raedt y Flach, P. (eds.), *Machine Learning: ECML 2001.* ECML 2001. Lecture Notes in Computer Science, vol 2167. Berlin: Springer Heidelberg.

CAPÍTULO II

Inteligencia artificial y cerebro computacional a través de la fusión de datos*

Humberto Bustince

La neurociencia computacional es un área de estudio que trata de entender el funcionamiento del cerebro, utilizando inteligencia artificial, matemáticas, ciencia de datos o física, entre otros. Dentro de la neurociencia computacional, el cerebro computacional investiga cómo actuar sobre objetos físicos con la actividad cerebral. En este trabajo vamos a presentar algunos de los recientes desarrollos en este último campo, centrándonos en cómo manejar los datos y la incertidumbre ligada a ellos. El objetivo es desarrollar técnicas que permitan, por ejemplo, recuperar movilidad a pacientes afectados de enfermedades neurológicas.

Palabras clave: neurociencia computacional, cerebro computacional, fusión de datos, incertidumbre.

* Este trabajo ha sido parcialmente financiado por el proyecto PID2022-136627NB-I00 financiado por MCIN/AEI/10.13039/501100011033/FEDER, UE.

1. INTRODUCCIÓN

En los últimos años, es ya un tópico decir que la inteligencia artificial (IA) está adquiriendo un creciente protagonismo en prácticamente todos los campos de la actividad humana. En particular, los recientes desarrollos de Grandes Modelos de Lenguaje *(LLM)* como el ChatGPT han copado y siguen acaparando titulares en la prensa y dando lugar a numerosas discusiones sobre sus posibilidades y riesgos. Muchas de estas discusiones se mueven en terrenos más próximos a la ciencia ficción que a la realidad científica, contribuyendo en ocasiones a crear confusión, temores infundados o falsas esperanzas. Al mismo tiempo, no se pone el foco en problemas cruciales como la interpretabilidad, la explicabilidad o la necesidad de datos de calidad.

Por otra parte, hay muchos desarrollos que, a pesar de contar con una historia relativamente larga, han ido pasando más desapercibidos para la sociedad. Y ello, a pesar de que su impacto potencial en nuestras vidas resulta inmenso y de que están recibiendo fuertes inversiones económicas. Este es el caso, por ejemplo, de las investigaciones en el campo de la salud ligadas a la noción de medicina personalizada o de precisión (Smith, 2012), que vienen siendo impulsadas desde hace años y están ya transformando la forma en que se entiende y se practica la asistencia sanitaria.

Otro campo que puede ser clave para el futuro de la inteligencia artificial es el dedicado a la comprensión de la inteligencia humana. Dado que esta última depende, en gran medida, de que entendamos cómo funciona su elemento más representativo, el cerebro, no es de extrañar el gran esfuerzo que está atrayendo la neurociencia computacional (Barrenechea *et al.,* 2013; Lindsay, 2021); es decir, el estudio y comprensión de los mecanismos de funcionamiento del cerebro humano.

En este trabajo, vamos a considerar algunos recientes desarrollos en una de las áreas específicas de estudio dentro de la neurociencia computacional: el denominado cerebro computacional. En concreto, nos centraremos en cómo podemos actuar sobre dispositivos externos por medio de la actividad cerebral (Sejnowski y Churchland, 1992). Este desarrollo específico presenta el interés añadido de permitirnos discutir de forma simple algunos aspectos críticos para el desarrollo de la inteligencia artificial:

1. cómo podemos extraer información a partir de los datos disponibles, y

2. qué ocurre cuando existen problemas de incertidumbre o ruido ligados a los datos.

En lo fundamental, vamos a seguir los desarrollos en Ko *et al.* (2019), si bien también incluiremos resultados de trabajos posteriores. En todo caso, el esquema del trabajo es el siguiente. Comenzamos con una breve discusión sobre el concepto de inteligencia artificial. En la sección 3, comentamos las principales nociones relacionadas con la neurociencia computacional y el cerebro computacional. En la sección 4 tratamos los métodos no invasivos para el cerebro computacional, y en la sección 5, que se puede considerar el núcleo de esta propuesta, nos centramos en el problema de los datos. Terminamos con unas breves conclusiones y una lista de referencias.

2. ¿QUÉ ES LA INTELIGENCIA ARTIFICIAL?

Dado que vamos a utilizar el problema del cerebro computacional para entender algunos aspectos relevantes de la inteligencia artificial, es importante comenzar fijando el terreno sobre el que nos movemos. Así pues, empezamos tratando de responder la pregunta que sirve de título a esta sección.

El Reglamento de inteligencia artificial de la Unión Europea de 2024 (EU Artificial Intelligence Act, 2024), define la inteligencia artificial como:

> "Sistemas automatizados diseñados para operar con diversos grados de autonomía y que, en función de objetivos explícitos o implícitos, generan resultados, predicciones, recomendaciones o decisiones que influyen en el entorno físico o virtual".

Esta definición, aunque formalmente correcta, resulta bastante oscura y poco precisa. Por ello, en este trabajo vamos a considerar la que dio el investigador John McCarthy, uno de los padres de la inteligencia artificial, en la convocatoria de la Conferencia de Dartmouth de 1955 (McCarthy, 1955). Esta conferencia se considera el hito fundacional de la inteligencia artificial, y fue precisamente donde se acuñó el término, definido como sigue:

> "inteligencia artificial es la ciencia de construir máquinas para que hagan cosas que, si las hicieran los humanos requerirían inteligencia".

El primer punto a resaltar en esta definición es que establece claramente el carácter híbrido de la inteligencia artificial. Por una parte, es una ciencia. Por tanto, requiere de sólidos fundamentos teóricos y del uso del método experimental para obtener y validar posibles resultados. Y, por otra parte, presenta un carácter fuertemente aplicado. Esta dualidad resulta especialmente clara en el caso del cerebro computacional, como veremos.

¿De qué forma puede desarrollarse la inteligencia artificial? O, dicho de otro modo, ¿cómo se pueden desarrollar esas máquinas inteligentes? Desde sus orígenes, se han utilizado dos posibles aproximaciones o paradigmas para responder a esta cuestión.

- El paradigma simbólico (Turing, 1950), que se centra en el desarrollo de sistemas que actúan de forma racional. Es decir, de algoritmos, sistemas expertos, que están, en general enfocados a la resolución de problemas concretos. Este paradigma fue el dominante desde los años 50 hasta la irrupción de las redes neuronales en los años 90.

- El paradigma conexionista (Medler, 1998), que pretende crear sistemas que imitan el cerebro humano. Es este paradigma el que se halla detrás de los desarrollos de redes neuronales actuales.

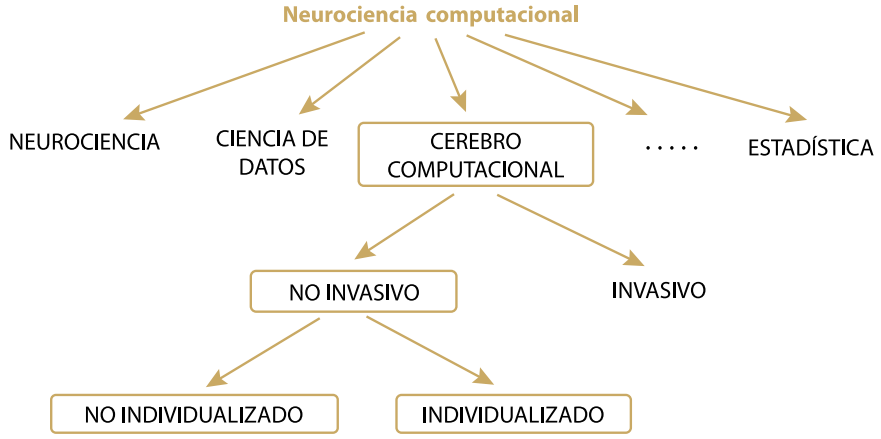
La neurociencia computacional se enmarca en el paradigma conexionista, ya que pretende profundizar en la comprensión del cerebro. Sin embargo, ambos paradigmas no son mutuamente excluyentes, ya que, como vamos a ver en el caso del cerebro computacional, también se va a hacer uso de técnicas y algoritmos propios del paradigma simbólico.

3. LA NEUROCIENCIA COMPUTACIONAL Y EL CEREBRO COMPUTACIONAL

La neurociencia computacional es un campo interdisciplinar que combina la neurociencia, la ciencia de datos, la informática, las matemáticas y la ingeniería para entender cómo funciona el cerebro y utilizar esta comprensión para desarrollar nuevas tecnologías (Trappenberg, 2010). Se trata, por tanto, de un campo muy amplio con numerosas ramificaciones y con un fuerte componente multidisciplinar. En la figura 1 mostramos un esquema muy simplificado de su estructura, con algunos de sus elementos más relevantes.

Figura 1.

Esquema de la neurociencia computacional



Fuente: Elaboración propia.

Como ya hemos indicado, en este trabajo nos vamos a centrar específicamente en el problema del cerebro computacional. Es decir, en el estudio de interfaces cerebro-computador (*BCI*, por sus siglas en inglés), que permitan a los individuos controlar dispositivos externos con su actividad cerebral (Vidal, 1973).

Podemos decir, de forma simplificada, que existen dos métodos fundamentales para abordar los problemas del *BCI*: los invasivos (Polikov *et al.,* 2005) y los no invasivos (Bozinovski y Bozinovska, 2019). Empezamos proporcionado una breve visión de los métodos invasivos, considerando los desarrollos de la empresa Neuralink.

Los métodos invasivos consisten en utilizar dispositivos insertados dentro del propio cerebro para obtener información de la actividad de este. En la actualidad, el más conocido de estos métodos (aunque no el único existente) es Neuralink, desarrollado por la empresa del mismo nombre que fue fundada por Elon Musk en 2016 (https://neuralink.com/).

Según la información proporcionada por la propia empresa, Neuralink obtiene información de neuronas individules, no de grupos de neuronas como sucede en los modelos no invasivos. En concreto, Neuralink utiliza un chip que incluye 64 cadenas flexibles de polímeros. Esto proporciona un total de 1.024 puntos que pueden utilizarse para registrar la actividad cerebral (Neuralink, 2019).

Aunque existen muchas dudas desde el punto de vista científico, la empresa ha informado de que el primer implante de uno de estos chips se llevó a cabo en enero de 2024 y de que, poco después, ya era posible detectar las señales de actividad neurológica del paciente. Más aún, la empresa añade que el dispositivo habría sido implantado en un segundo paciente en julio del mismo año, y que este paciente habría mejorado su capacidad para jugar a videojuegos y estaría entrenándose en el uso de *software* para el diseño de objetos 3D.

Con todo, es necesario mantener una actitud prudente ante estas afirmaciones. La comunidad científica ha expresado su preocupación por la falta de transparencia de estos desarrollos (Regalado, 2020). De hecho, el experimento fue inicialmente rechazado por la FDA (Food and Drug Administration, organismo responsable de autorizar ensayos clínicos y medicamentos en EE. UU.), aunque posteriormente sí dio su aprobación. En cualquier caso, es importante señalar que los pacientes no aparecen registrados en la base de datos de ensayos clínicos de Estados Unidos.

4. MÉTODOS NO INVASIVOS EN EL CEREBRO COMPUTACIONAL

Los métodos no invasivos tratan de obtener información por medio de dispositivos externos, como los electroencefalogramas. Entre estos métodos, son muy populares los basados en imagen motora. Es decir, métodos en los que el usuario imagina determinados movimientos que son transmitidos a los dispositivos externos y en los que las señales recibidas por estos deben utilizarse para recuperar el pensamiento del sujeto.

Dentro de los métodos no invasivos, y teniendo en cuenta el tipo de dispositivos a desarrollar, podemos distinguir dos grandes grupos:

- Los métodos que podemos denominar como no individualizados, en los que se trata de obtener información de diferentes sujetos para poder desarrollar dispositivos que sean, de alguna forma, universales. Esto es, aplicables, *a priori*, a cualquier individuo.

- Los métodos individualizados, que se centran en un sujeto concreto para desarrollar dispositivos específicos para dicho individuo.

Las técnicas utilizadas en los dos métodos son similares. De hecho, la principal diferencia entre ambos se halla en los datos que se utilizan para su desarrollo: datos provenientes de individuos diferentes en el primer caso, y de un solo individuo, en el segundo. Por ello, en este trabajo consideramos los métodos no individualizados.

En concreto, vamos a examinar el problema de identificar si un sujeto está pensando en mover su mano derecha o su mano izquierda, para el caso no individualizado, tal y como se discute en Ko *et al.* (2019). En primer lugar, describimos las principales etapas del tratamiento del problema.

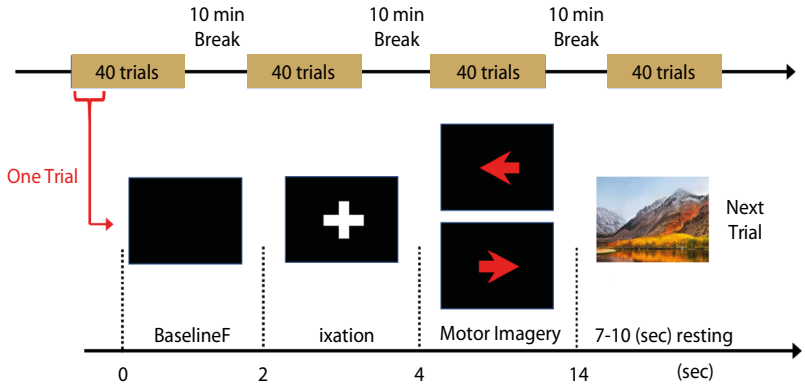
4.1. Captación de datos

El punto de partida, como en todos los problemas que hacen uso de técnicas de inteligencia artificial, son los datos. Tal y como se explica en Ko *et al.* (2019), en este caso los datos se obtienen en laboratorio. Concretamente, se consideran diez individuos con edades comprendidas entre los 18 y los 29 años, y que no presentan transtornos neurólogicos. Cada participante se sitúa frente a un monitor y es conectado a un dispositivo individual que recoge las señales de su electroencefalograma. A cada uno de ellos se le pide pensar varias veces consecutivas en mover la mano derecha o la mano izquierda. Por medio de un electroencefalograma, se captan las señales eléctricas de la actividad neuronal cuando está pensando en el movimiento correspondiente.

Dado que estas señales son eléctricas, es necesario determinar qué frecuencias van a ser relevantes. Dependiendo del experimento, se utilizan varios canales distintos. La figura 2 muestra en resumen el proceso de captación de datos.

Figura 2.

Procedimiento de captación de datos



Fuente: Ko *et al.* (2019).

En total, cada participante realizaba un total de 160 pruebas. Siguiendo la metodología estándar dentro del aprendizaje máquina, los datos asociados a cada una de estas pruebas se distribuyen aleatoriamente en dos conjuntos disjuntos.

- 80 pruebas para entrenamiento, utilizadas para que el sistema aprenda a predecir si el sujeto está pensando en mover la mano derecha o la mano izquierda.

- 80 pruebas para test, que son utilizadas para comprobar la precisión del sistema una vez entrenado.

Es importante reseñar que, para que el problema esté equilibrado, cada uno de los conjuntos de 80 pruebas constaba de 40 ejemplos de pensar en mover la mano derecha y otros 40 de mover la mano izquierda.

4.2. Preprocesamiento de los datos

En general, en los problemas del cerebro computacional, como en la mayor parte de los problemas en inteligencia artificial, no es posible utilizar directamente los datos obtenidos. Es necesario someterlos a un proceso de "limpieza", conocido como preprocesamiento. En este problema en concreto, el preprocesamiento consta de dos etapas distintas:

- En primer lugar se aplica una transformada rápida de Fourier, para reducir la complejidad de los datos y descomponerlos en diferentes componentes de frecuencia.

- A continuación se usa *Common Spatial Pattern*, que es una técnica matemática habitualmente utilizada para separar las señales multivariables en subcomponentes con máxima variación espacial (Pfurtscheller *et al.,* 1999).

De manera informal, el resultado de esta procesamiento es que las señales en cada banda vienen descritas por una serie de números que son la información (los datos) que va a utilizar nuestro sistema para predecir en qué está pensando el sujeto. Para cada sujeto y cada ensayo disponemos, por tanto, de cuatro conjuntos de números, correspondientes, respectivamente, a cada una de las bandas consideradas.

4.3. Clasificación

Este es el núcleo del método. Nuestro problema puede verse como un problema de clasificación: existen dos clases (mano derecha y mano izquierda) y para cada conjunto de datos (esto es, para cada individuo y cada prueba), debemos determinar a cuál de ellas pertenece. La pregunta es: ¿cómo podemos llevar a cabo esta clasificación?

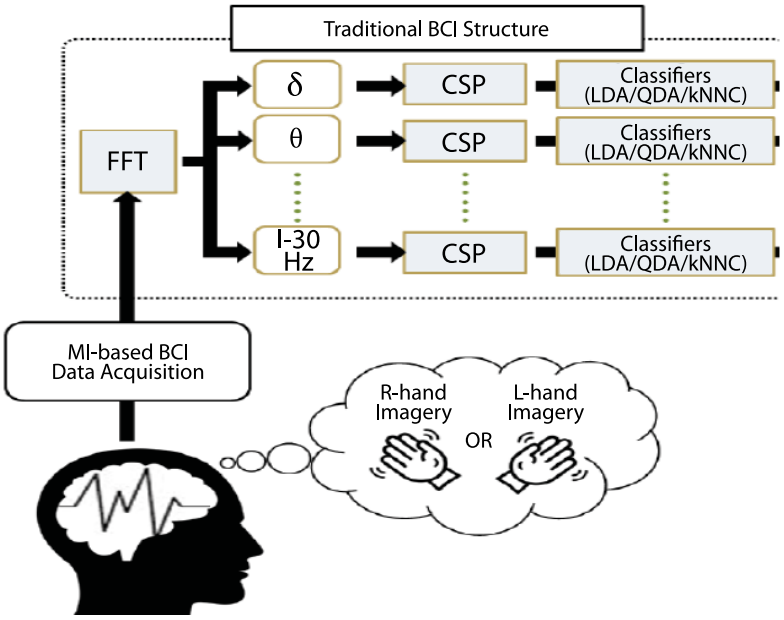
Existen muchos algoritmos de clasificación diferentes. Tres de los más conocidos son el análisis discriminante lineal *(LDA)*, el análisis discriminante cuadrático *(QDA)* y la clasificación por los k-vecinos más próximos. La elección de estos métodos o de otros no es especialmente relevante para nuestra discusión, así que no vamos a entrar en muchos detalles acerca de los mismos.

Lo que sí es relevante es que entrenamos cada uno de los clasificadores con cada uno de los canales de cada uno de los ensayos. Esto es, para cada cada canal, enseñamos al clasificador a distinguir si estamos pensando en mover la mano derecha o la mano izquierda. De esta forma, una vez finalizado el entrenamiento, si introducimos una nueva señal en un canal dado, cada clasificador nos dirá la probabilidad de que el sujeto esté pensando en mover la mano derecha o la mano izquierda. Esto quiere decir que, para cada señal, podemos obtener varias respuestas posibles, una por cada clasificador. ¿Cómo podemos fusionarlas en la clasificación definitiva? Este es el paso clave en el estudio, que se realiza por medio de las denominadas funciones de fusión de datos.

La figura 3 muestra un esquema del proceso completo descrito hasta aquí.

Figura 3.

Esquema del proceso tradicional de clasificación de los datos



Fuente: Ko *et al.* (2019).

5. LA FUSIÓN DE DATOS

5.1. Funciones de fusión de datos

Esta subsección es la más técnica desde un punto de vista matemático, pero es fundamental para poder entender plenamente los desarrollos posteriores del trabajo. Sin embargo,

aquellos lectores interesados en una visión más general del problema pueden omitirla y simplemente referirse a ella cuando se den definiciones concretas.

La fusión de datos es un proceso fundamental en casi todos los campos de la investigación. De una forma informal, podemos definirla como el proceso de obtener, a partir de un conjunto de datos homogéneos o heterogéneos, un nuevo valor que represente a todos ellos de la forma más precisa posible.

Uno de los métodos más utilizados para la fusión de información se basa en el uso de las llamadas funciones de agregación (Grabisch *et al.,* 2009). Una función de agregación es una función creciente que toma valores en el intervalo [0,1] y devuelve otro valor también en el intervalo [0,1], de modo, que, si todos los valores de entrada son 0, el valor devuelto es 0; y, si todos los valores de entrada son 1, el valor devuelto es 1.

Entre los ejemplos más conocidos de funciones de agregación se encuentran las medias (Beliakov *et al.,* 2016), como la media aritmética o la media geométrica, el máximo o el mínimo. Todos estos ejemplos consideran los datos de entrada individualmente. Esto es, no tienen en cuenta en general la posible relación que pueda existir entre datos diferentes, y que puede haber sido determinada de forma experimental. Sin embargo, en muchas ocasiones los datos no son independientes, sino que pueden reforzarse o inhibirse entre sí. Esto es particularmente cierto en el caso que nos ocupa del cerebro computacional, ya que parece razonable suponer que los diversos canales de la señal eléctrica tienen algún tipo de conexión entre ellos que va más allá de la mera correlación. Es decir, que la información de algunos canales es más significativa tomada junto a la de otros, que de forma individual, en especial a partir de determinados umbrales para los valores de la señal.

Para poder considerar la fusión de este tipo de datos que pueden estar relacionados entre sí, se considera toda una familia de funciones de agregación basadas en el concepto de medida difusa. Una medida difusa es simplemente una función que asigna un valor entre [0,1] a cada posible subconjunto de los datos, con algunas restrcciones. Si el subconjunto considerado es vacío, es decir, si no consideramos ningún dato, la medida toma el valor cero. Si, por el contrario, consideramos todos los datos disponibles a la vez, la medida asignada es 1; es decir, máxima. Finalmente, también se impone que si un subconjunto de datos A está incluído en otro subconjunto de datos B, la medida de A no puede ser mayor que la medida de B. De esta forma, podemos utilizar la medida para determinar hasta qué punto datos distintos se refuerzan entre sí.

¿Qué funciones de agregación podemos construir por medio de estas medidas? Aunque existen muchas posibilidades, las dos más utilizadas, y las que consideramos en este trabajo, son la integral Choquet (Choquet, 1953-54) y la integral Sugeno (Sugeno, 1974), que pasamos a describir brevemente.

Consideremos un conjunto de valores $x_1, \ldots, x_n$ en el intervalo $[0,1]$. Vamos a ordenar estos valores de forma creciente. Es decir los reordenamos por medio de una permutación $(\cdot) : \{1, \ldots, n\} \to \{1, \ldots, n\}$ de forma que:

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)} .$$

Por ejemplo, si tomamos n=3 y x_1=0.6 , x_2=0.2 , x_3=0.4 , para reordenar debemos considerar la permutación (1)=2 (2)=3 y (3)=1 , con lo que nos queda:

$$x_{(1)} = x_2 = 0.2 < x_{(2)} = x_3 = 0.4 < x_{(3)} = x_1 = 0.6 .$$

tal y como queríamos. Este ordenamiento nos va a permitir ir considerando subconjuntos teniendo en cuenta qué datos son mayores o menores y la medida va a tener en cuenta este tamaño relativo de los datos, permitiendo dar importancia a los términos mayores. Para ello, vamos a definir, para cada i=1,..., n, $A_{(i)}$ como el conjunto de índices (originales) de los elementos mayores o iguales que $x_{(i)}$. Es decir, con el ejemplo anterior:

$$A_{(1)} = \{1,2,3\}, \quad A_{(2)} = \{1,3\}, \quad A_{(3)} = \{1\} .$$

El objetivo de estas notaciones, que pueden parecer un poco complejas, es poder introducir las dos funciones siguientes. En primer lugar, definimos una medida m como una función que, dado un subconjunto de $\{1,...,n\}$, le asocia un valor entre 0 y 1 y que cumple las dos propiedades siguientes:

1. $m(\{1,...,n\}]=1$ y $m(\emptyset)=0$.
2. Si $A \subseteq B$ entonces $m(A) \leq m(B)$.

Sin embargo, lo que no se exige en general es que la medida de una unión de dos conjuntos disjuntos sea la suma de sus medidas correspondientes. Este hecho nos permite tener en cuenta el posible reforzamiento de la información al tener en cuenta varios datos distintos. En efecto, si en un conjunto tenemos solo dos elementos a y b , está definición nos permite decir que la medida del conjunto $\{a, b\}$ nunca va a ser más pequeña que las de a y b por separado, pero puede ser mayor. Usando un símil futbolístico, si tenemos dos delanteros de calidad, su efectividad es mayor si además resulta que son capaces de colaborar de forma adecuada.

Utilizando este concepto de medida, podemos considerar las dos definiciones siguientes:

- La integral Choquet de un conjunto de números $x_1,..., x_n$ en [0,1] con respecto a la medida m se define como:

$$C_m(x_1,\ldots,x_n) = \sum_{i=1}^{n}(x_{(i)} - x_{(i-1)})m(A_{(i)})$$

donde establecemos que $x_{(0)}$=0.

- La integral Sugeno de un conjunto de números $x_1,..., x_n$ en [0,1] con respecto a la medida m se define como:

$$S_m(x_1,\ldots,x_n) = \max_{i=1}^{n} \min(x_{(i)}, m(A_{(i)})$$

Es importante observar que en los últimos sumandos únicamente se tienen en cuenta para la medida, los mayores valores entre los datos de entrada, de modo que se puede valo-

rar la relación entre estos. Por ejemplo, si los valores más altos se obtienen para dos señales concretas, la medida permite resforzar su influencia en el resultado final. Experimentalmente, se ha comprobado que estas integrales proporcionan muy buenos resultados en problemas diversos, que van desde el procesamiento de imagen (Paternain, 2015) a la clasificación (Barrenechea *et al.,* 2013) o la toma de decisión (Lourenzutti *et al.,* 2017) cuando se comparan con las funciones de agregación más tradicionales (medias, productos...). De hecho, también es importante destacar que esto es debido, en parte, a que funciones como las medias ponderadas pueden recuperarse como ejemplos específicos de estas funciones, luego estamos considerando una familia más general que incluye las funciones usadas tradicionalmente en la literatura.

Sin embargo, en algunas ocasiones, estas funciones no captan adecuadamente la complejidad del problema a considerar. Este hecho ha llevado a investigar la forma de extenderlas o generalizarlas, de modo que, por una parte, se preserven algunas de sus propiedades más relevantes, y, por otra parte, se obtenga una mayor flexibilidad para tratar con problemas complejos. En particular, para el problema concreto del cerebro computacional, se han considerado generalizaciones de ambas integrales. En concreto, se trata de reemplazar el producto y el mínimo (y, por tanto, la linealidad) por funciones más generales que induzcan un comportamiento no lineal en la definición. Como muestra de estas generalizaciones, podemos dar los dos ejemplos siguientes:

1. Respecto a la integral Choquet, podemos hacer uso de la propiedad distributiva del producto y reescribir la integral Choquet como:

$$C_m(x_1,\ldots,x_n)=\sum_{i=1}^{n} x_{(i)}m(A_{(i)})-x_{(i-1)}m(A_{(i)})$$

Ahora, si reemplazamos el producto en cada uno de los términos de la derecha por sendas funciones F_1, F_2 con propiedades adecuadas, obtenemos las $CF1F2$ -integrales (Dimuro, 2020), definidas, para $x_1,\ldots, x_n$ en [0,1], por:

$$C_m^{F_1,F_2}(x_1,\ldots,x_n)=\min(1,\sum_{i=1}^{n} F_1(x_{(i)},m(A_{(i)}))-F_2(x_{(i-1)},m(A_{(i)}))).$$

Obsérvese que introducimos el mínimo al comienzo simplemente para asegurar que el valor obtenido se encuentra entre 0 y 1 . Las funciones F_1 y F_2 consideradas proporcionan mayor flexibilidad. Algunos ejemplos de funciones que podemos utilizar en esta construcción son $F_1(x,y)=\sqrt{x,y}$ y $F_2(x,y)$=max $(0, x+y-1)$, entre otras muchas.

2. De forma similar, podemos reemplazar el mínimo en la definición de integral de Sugeno por otra función M de dos variables y obtener la siguiente generalización (Bardozzo *et al.,* 2021):

$$S_m^M(x_1,\ldots,x_n)=\max_{i=1}^{n} M(x_{(i)},m(A_{(i)}))$$

Como función M en este caso podemos tomar, por ejemplo, el producto o las mismas funciones que en el caso anterior.

Es importante destacar que, al llevar a cabo estas generalizaciones, debemos pagar un cierto precio. En particular, perdemos la monotonía. Esto es, si tenemos un conjunto de datos

A y otro conjunto de datos *B* y todos los datos en *A* son mayores o iguales que los correspondientes datos en *B*, no tienen por qué cumplirse que el resultado de aplicar estas funciones a los datos de *A* sea mayor o igual que el resultado obtenido al aplicarlas a los datos de *B*. Sin embargo, esta propiedad de monotonía no se pierde completamente, ya que sigue siendo cierta si los datos crecen de una forma concreta, siguiendo una dirección o un conjunto de direcciones espocíficas. Esto es lo que se denomina monotonía direccional. Las funciones, como estas, que satisfacen esta forma menos restrictva de monotonía y que toman el valor 0 si todos los valores de entrada son 0 , y el valor 1 si todos los valores de entrada son 1, se denominan preagregaciones (Lucca *et al.,* 2016). Las preagregaciones están atrayendo un considerable interés en los últimos tiempos, ya que permiten mejorar los resultados obtenidos con muchos algoritmos que hacen uso de agregaciones. En este trabajo utilizaremos el término de función de fusión de datos o información para referirnos tanto a funciones de agregación como a preagregaciones.

5.2. La fusión difusa multimodal de datos en el cerebro computacional

El uso de las funciones de fusión de datos introducidas en la subsección anterior tiene como objetivo mejorar los resultados que puede proporcionar cada uno de los clasificadores individualmente. Visto de otra forma, se trata de aprovechar la información obtenida de todos los clasificadores considerados para lograr una mejor predicción en el problema. La pregunta es: ¿cómo podemos llevar a cabo esta fusión de datos?

Existe una amplia literatura sobre los denominados ensembles. Es decir, sobre cómo llevar a cabo esta fusión de los datos provenientes de diversos clasificadores (Bolón-Canedo y Alonso-Betanzos, 2019). En este trabajo, vamos a considerar la aproximación a este problema dada en Ko *et al.* (2019), y denominada Fusión Difusa Multimodal, y que lleva a cabo la fusión de la información en varias etapas, como se describe más abajo. La elección de este método es debida a los buenos resultados obtenidos. Se trata, por tanto, de una elección empírica, y, a día de hoy, se carece de una justificación completa para estos buenos resultados.

La Fusión Difusa Multimodal consta de tres etapas:

1. En primer lugar, para cada clasificador, obtenemos el resultado que dicho clasificador proporciona para cada uno de los canales por separado.

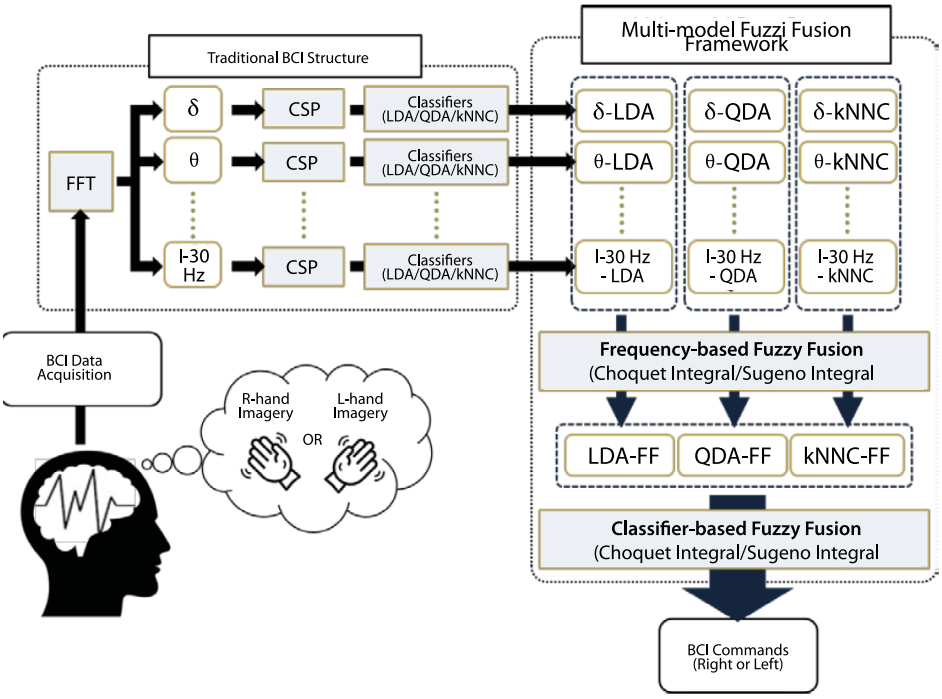
2. A continuación, también para cada clasificador, fusionamos los resultados obtenidos en el paso 1. Es decir, para cada clasificador obtenemos un único resultado (probabilidad de que el sujeto esté pensando en mover la mano derecha y probabilidad de que esté pensando en mover la mano izquierda). Al completar este paso, tenemos, por tanto, tantos resultados como clasificadores.

3. Finalmente, fusionamos los resultados de todos los clasificadores considerados, para obtener una respuesta definitiva.

En la figura 4 mostramos el esquema completo de este algoritmo, añadiendo los nuevos pasos de función de información al esquema tradicional mostrado en la figura 3.

Figura 4.

Esquema del proceso completo de clasificación de los datos



Fuente: Ko *et al.* (2019).

Es importante destacar que en las dos últimas etapas, en que se lleva a cabo la fusión de datos, no tiene por qué utilizarse la misma función.

Desde el punto de vista teórico, el método propuesto tiene sentido y parece razonable. Sin embargo, ¿proporciona alguna mejora el hecho de considerar fusión de datos en dos etapas distintas? En Ko *et al.* (2019) se considera la respuesta a esta pregunta. Los resultados de esta comparación que se pueden encontrar en ese trabajo reflejan que, en general se obtienen una mejora de la precisión frente a la obtenida fusionando solo los resultados de todos los clasificadores o la precisión fusionando solo los resultados de las bandas para cada clasificador. Además, la tasa de acierto obtenida se aproxima a un 75 %.

Ahora bien, para poder valorar la calidad del método, es necesario tener en cuenta que este ha sido desarrollado con un conjunto de datos muy específico obtenido en laboratorio y que tanto las funciones como las medidas han sido escogidas *ad hoc.* ¿Qué sucede si se trata de extender el mismo método a otros conjuntos de datos, provenientes del mundo real?

En Fumanal-Idocin *et al.* (2024) se incluye una interesante discusión sobre los resultados de aplicar este mismo método a un problema más complejo: determinar si el individuo está pensando en mover la mano izquierda, la mano derecha, un pie o la lengua. Aunque formalmente el problema es muy similar al anterior, la precisión del método cae drásticamente a valores alrededor de un 60 o 65 % en el mejor de las casos. Esto claramente indica que existe un amplio margen de mejora en el algoritmo y plantea una cuestión: ¿a qué puede deberse esta pérdida de precisión?

Obviamente, esta pregunta no admite una respuesta simple ni única. A partir de los resultados en Fumanal-Idocin *et al.* (2024), vamos a discutir una de las posibles causas del problema y vamos a ver cómo podemos modificar el método de fusión de información para tenerla en cuenta.

5.3. El problema de la incertidumbre

En todo problema de fusión de información, uno de los principales factores que pueden afectar al carácter más o menos óptimo de la solución obtenida es el de la calidad de los datos disponibles. De hecho, no es exagerado afirmar que la base del éxito de cualquier aplicación de la inteligencia artificial se halla, precisamente, en los datos utilizados. No es casualidad que la explosión de desarrollos de inteligencia artificial que hoy vivimos haya venido de la mano del desarrollo de técnicas para poder tratar computacionalmente grandes cantidades de datos.

Cuando consideramos una aplicación concreta, como puede ser el cerebro computacional, hay dos preguntas fundamentales a responder relativas a los datos:

1. ¿De cuántos datos disponemos? Algunos métodos, como las redes neuronales profundas, solo proporcionan resultados óptimos si se dispone de cantidades suficiente mente grandes de datos para entrenarlas. Esto hace que su aplicación en algunos problemas concretos pueda resultar muy compleja o incluso ineficiente.

2. ¿Cómo de buenos son los datos de los que disponemos? En ocasiones, se considera que disponer de datos es equivalente a disponer de largas tablas de valores. Sin embargo, la realidad es más compleja. En primer lugar, existe la posibilidad de sesgos, es decir, que los valores de los que disponemos (si nos centramos en el caso numérico) no reflejen adecuadamente todos los casos de interés, o que sobrerrepresenten algunos ejemplos o clases frente a otros. Pero, más aún, aunque se extreme el cuidado para evitar esta situación, puede ser que los datos no sean totalmente correctos. En particular, si los datos se obtienen a partir de sensores (utilizando esta palabra en su sentido más general) y si utilizamos ordenadores para su tratamiento, nunca vamos a poder garantizar la exactitud de los valores con los que trabajamos. Es decir, nuestros datos van a estar afectados de incertidumbre.

La incertidumbre, entendida como la imposibilidad de conocer el valor exacto de un dato, es inherente a la ciencia de datos, y por ende, a todos los procesos que utilizan datos. En el caso concreto del cerebro computacional, recordemos que los datos se obtienen utilizando electroencefalogramas fijados a la superficie de la cabeza. Esto implica que las señales eléctri-

cas captadas por estos dispositivos están inevitablemente afectadas de una importante cantidad de ruido. Es decir, que los valores que estamos midiendo, no son los reales, sino que son perturbaciones de esos valores reales, y no tenemos forma de conocer cuál es el valor exacto de los mismos. Dado que nuestros algoritmos hacen uso de estos valores, esta incertidumbre debe afectar al resultado final. Por tanto, una posible vía de mejora de los algoritmos es reducir la cantidad de ruido para tratar de obtener datos para nuestro uso los más próximos posibles a los datos reales. Esta es una aproximación seguida en muchos casos. Así, por ejemplo, es posible captar las señales dentro de una jaula de Faraday, para evitar distorsiones debidas a campos electromagnéticos externos, o intentar mejorar la sensibilidad de los sensores utilizados.

Pero en muchas ocasiones, no es posible eliminar completamente el ruido, por lo que debemos asumir que está presente. ¿Qué podemos hacer en este caso? Una salida es tenerlo en cuenta a la hora de fusionar la información. O, más concretamente, incluirlo dentro del proceso de fusión incorporándolo de alguna manera a la representación de los datos que consideramos.

¿Cómo es posible hacer esto? Una forma muy simple es por medio de intervalos. La idea no es en sí misma original ya que, de hecho, el uso de intervalos es habitual en campos como la Estadística. Se trata de una idea natural y simple de desarrollar. Supongamos que hay un dato real, que no conocemos, pero que podemos acotar. Para fijar las ideas, supongamos que el dato real es 0.3, pero que no lo conocemos, porque la medición que nosotros hacemos para obtener ese dato está afectada por ruido. Lo que podemos tratar de hacer es dar una cota inferior y una cota superior para ese valor real que no conocemos. Así, podemos decir que el dato está en el intervalo [0.2, 0.5]. El punto clave es darnos cuenta de que podemos ir un paso más allá y representar el dato, desconocido por ese intervalo. Es decir, el dato con el que vamos a trabajar es, directamente, el intervalo [0.2, 0.5].

¿Cómo obtenemos este intervalo? Aunque existen métodos muy elaborados, es posible hacerlo de una manera muy simple. Podemos, por ejemplo, realizar varias mediciones y considerar como el intervalo resultante aquel dado por el menor y el mayor de los valores medidos. También es posible diseñar funciones específicas que dado un dato concreto, le asocien un intervalo, de forma similar a como se pueden construir intervalos de confianza en Estadística. No vamos a entrar en estos detalles, porque nos alejarían del objetivo de esta exposición.

Sin embargo, en la representación escogida del valor por medio del intervalo, hay un elemento esencial a tener en cuenta. Queremos representar, también, la incertidumbre asociada a ese mismo dato. Para entender este punto, imaginemos que todos los datos con los que trabajamos están en el intervalo [0,1] . Si un dato concreto x viene representado por el intervalo [0.34, 0.35], estamos bastante seguros de cuál es el valor real. Sin embargo, si otro dato y viene representado por el intervalo [0,1] , estamos diciendo que de hecho no tenemos ni idea de cuál es el valor real. En otras palabras, podemos considerar que la amplitud del intervalo que utilizamos para representar un dato concreto (la diferencia entre el extremo superior y el extremo inferior del intervalo) es una medida de la incertidumbre que tenemos respecto al valor real de dicho dato. La construcción de intervalos para reemplazar a un dato teniendo en cuenta la incertidumbre ligada a él puede hacerse de numerosas maneras diferentes. Por ejemplo, es posible realizar diferentes mediciones de la misma señal y considerar el intervalo entre el menor

valor obtenido y el mayor valor obtenido. O puede asumirse que la incertidumbre depende del valor medido. por ejemplo, experimentalmente puede determinarse que, para valores muy próximos a cero podemos asumir, en función de las características de nuestros dispositivos que existe una incertidumbre muy pequeña, y que esta va creciendo a medida que aumentan los valores observados; o viceversa. Lo importante es que, al final, transformamos cada dato en un intervalo.

Desde un punto de vista matemático, la naturaleza de los intervalos es diferente a los de los números reales. En particular, si nos restringimos a subintervalos del intervalo unidad [0,1], hay dos puntos clave que deben tenerse en cuenta a la hora de extender el método que hemos explicado.

- En primer lugar, dados dos números reales distintos cualesquiera, uno siempre es menor o igual que el otro. Sin embargo, esto no es cierto dados dos intervalos, ya que no existe un orden "natural" para compararlos. Esto representa un problema, ya que las integrales difusas que hemos utilizado necesitan ordenar primero los datos de entrada.

- Además, en el caso concreto de la integral Choquet, es necesario considerar diferencias entre dos datos. Pero la diferencia entre intervalos tampoco está definida, en general.

Afortunadamente, es posible superar ambas objeciones gracias a desarrollos teóricos recientes. Así, el concepto de orden admisible (Bustince *et al.,* 2013) define una relación de orden entre intervalos que es total. Es decir, tal que permite comparar dos intervalos cualquiera para decidir cuál es mayor y cuál es menor. Estos órdenes admisibles no son únicos, sin embargo, y diferentes órdenes pueden dar lugar a resultados distintos. Esta es una línea en la que se está investigando en la actualidad.

En cuanto al segundo punto, podemos considerar que la diferencia de dos números es una medida de cómo son de distintos. Por tanto, para llevar a cabo un proceso similar con intervalos, una posibilidad es introducir una función, llamada de dismilitud, que devuelva un valor númerico de cómo de diferentes son los intervalos considerados. Esta idea ha dado lugar a la introducción de la noción de d-integral (Bustince *et al.,* 2021).

En resumen, es posible reproducir el método presentado anteriormente trasladando las correspondientes funciones de fusión de datos al marco intervalar. Este estudio ha sido llevado a cabo en Fumanal-Idocin *et al.* (2021) y en Fumanal-Idocin *et al.* (2022), donde se comprueba que el uso de intervalos, en combinación con una adaptación intervalar de la integral de Sugeno, permite mejorar los resultados significativamente. En concreto, y para comprobar la validez del método, se utilizan datos del dataset CBCIC (Chowdhury y Andreu-Perez, 2021). Este contiene datos de diez pacientes afectados de ictus cerebral que les ha provocado problemas de movilidad en las manos. Para cada paciente se incluyen datos de 80 pruebas pensando en mover la mano derecha o la mano izquierda. Se trata de un dataset estándar dentro de la literatura del cerebro computacional y que ha sido utilizado, por ejemplo, en la competición del Congreso Internacional WCCI 2020 para evaluar los resultados de diferentes algoritmos.

En la tabla 1 se muestran los resultados alcanzados, comparando los mejores resultados obtenidos por el método presentado con intervalos, y los mejores resultados obtenidos con otros métodos del estado del arte. En concreto, comparamos el método basado en el uso de la integral de Sugeno intervalovalorada con otros métodos del estado del arte. Se observa que el método intervalo valorado logra un mayor porcentaje de datos correctamente clasificados (segunda columna) con una mejor precisión (tercera columna).

Tabla 1.

Resultados con el dataset CBCIC

Modelo	*Precisión*	*F1-score*
IV Sugeno	0.8175 ± 0.1342	0.8149 ± 0.1366
EEG Net (Lawhern *et al.,* 2018)	0.6562 ± 0.1232	0.5933 ± 0.1712
Shallow Net (Tibor *et al.,* 2017)	0.7453 ± 0.13289	0.7342 ± 0.1489
Deep Net (Tibor *et al.,* 2017)	0.5331 ± 0.1356	0.4218 ± 0.1282
Multiscale CSP (Hersche *et al.,* 2018)	0.7956 ± 0.1144	0.7911 ± 0.1175
Gradient Boosting (Vijay *et al.,* 2020)	0.5956 ± 0.1203	0.5354 ± 0.1169

Fuente: Fumanal-Idocin *et al.* (2022).

Merece la pena destacar que algunos de los métodos del estado del arte considerado en esta tabla, como EEG Net, Shallow net y Deep Net, son, de hecho, redes neuronales profundas, a las que el método con intervalos considerado supera. Como ya hemos comentado, en el problema del cerebro computacional las redes neuronales profundas suelen carecer de datos suficientes para dar un rendimiento óptimo. Este es el caso, sobre todo, para los métodos individualizados.

5.4. El problema con cuatro clases: escogiendo la mejor función de fusión

Como ya hemos señalado anteriormente, los resultados empeoran significativamente si los sujetos, en lugar de pensar solo en mover la mano derecha o la mano izquierda, pueden pensar también en mover la lengua o uno de los pies. En este caso, incluso el uso de intervalos para considerar la incertidumbre no parece ser suficiente para obtener resultados suficientemente competitivos.

Si queremos considerar este problema, es necesario tener en cuenta un aspecto que, hasta ahora, hemos soslayado. ¿Es nuestro resultado indiferente a la función o funciones escogidas para llevar a cabo la fusión de los datos? Los experimentos muestran claramente que no es así, y que, de hecho, la función escogida puede influir significativamente. Pero entonces, ¿cómo podemos escoger la función más apropiada?

En general, no existe una respuesta a esta pregunta, ya que son todavía necesarios muchos estudios en profundidad que permitan clarificar la naturaleza del problema desde un punto de vista matemático para determinar las propiedades más deseables de la función de fusión. De hecho, el procedimiento más habitual actualmente es el ensayo y error. Utilizamos diversas funciones y nos quedamos con aquella o aquellas que mejor resultado dan.

Sin embargo, en la actualidad se está explorando una vía para intentar avanzar hacia una respuesta más satisfactoria. La idea es recordar que el objetivo de las funciones de agregación, en particular, y de las funciones de fusión, en general, es tratar de obtener un único valor (real o intervalar) que represente de la foma más adecuada posible los datos de entrada considerados. Esto puede interpretarse de la manera siguiente: queremos escoger como resultado de la fusión un valor que sea lo "más parecido" posible a los datos de entrada. La clave está en dar un significado apropiado a la expresión entrecomillada.

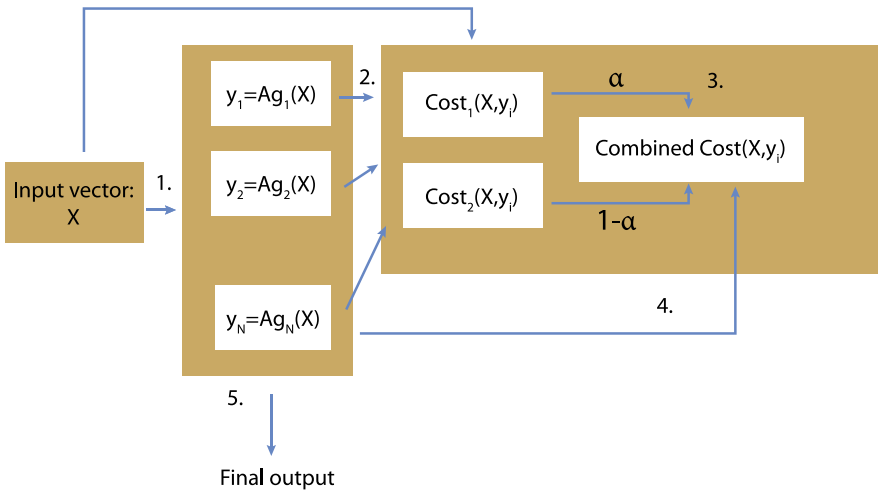
Esta idea está detrás del desarrollo de dos ideas en la literatura de inteligencia artificial: las funciones penalty (Calvo y Beliakov, 2010; Bustince *et al.,* 2017) y las desviaciones moderadas (Altalhi *et al.,* 2019). Sin entrar en detalles excesivamente técnicos, la idea detrás de ambas es similar y sería la siguiente. Supongamos que tenemos un conjunto de valores $x_1,\ldots, x_n$ a fusionar y un conjunto de funciones $M_1,\ldots, M_k$ que podemos utilizar. Cada una de estas funciones nos proporciona un valor $y_1,\ldots, y_k$ como resultado de la fusión. Pues bien, escogemos como valor final el y_j más parecido a los $x_1,\ldots, x_n$. Por fijar ideas, este valor puede ser el que minimiza el error cuadrático medio:

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - y_j)^2 ,$$

aunque en general las funciones penalty y de las desviaciones moderadas proporcionan aproximaciones más ricas a la idea de "mayor parecido". Esquemáticamente, el método se muestra en la figura 5, donde las funciones de comparación aparecen como funciones de coste que se combinan entre sí por medio de un parámetro α para afinar los resultados (Fumanal-Idocin *et al.,* 2024).

Figura 5.

Esquema de uso de funciones penalty y desviaciones moderadas



Fuente: Fumanal-Idocin *et al.* (2024).

Un aspecto a destacar del uso de funciones penalty y desviaciones moderadas es que la selección del valor representativo se realiza cada vez que es necesario llevar a cabo un proceso de fusión. Es decir, a la hora de fusionar los datos de la señal 1 del paciente 1, el método puede escoger el resultado obtenido con una función M_1 , pero al fusionar los datos de la señal 2 del mismo paciente, elegir el resultado obtenido con una función distinta M_2 . De esta forma se pueden tener en cuenta posibles diferencias en la naturaleza de los datos, incluyendo diferentes niveles de incertidumbre. Tal y como se muestra en (Fumanal-Idocin *et al.,* 2024), esta metodología proporciona resultados cercanos al estado del arte cuando se considera el problema de las cuatro clases discutido anteriormente.

Hay que tener en cuenta que esta es una línea de investigación en desarrollo, y que para poder explotarla plenamente se necesita un mejor conocimiento de la naturaleza de las señales electroencefalográficas para poder determinar como debe medirse de la forma más adecuada el parecido. Además, desde el punto de vista computacional, este es un método muy costoso, ya que, para obtener un resultado óptimo, es deseable disponer de un amplio conjunto de funciones de fusión de información, que es necesario utilizar una y otra vez para evaluar los datos disponibles.

6. CONCLUSIONES

En este trabajo nos hemos centrado en el problema de identificar en qué está pensando un individuo a través de las señales de un encefalograma. En concreto, hemos considerado el problema de determinar si está pensando en mover la mano derecha o la mano izquierda, y el problema más complejo de si está pensando en mover una de las manos, la lengua o un pie. Este problema nos ha permitido presentar dos de los puntos clave en el desarrollo de la inteligencia artificial actual: la forma de fusionar datos para obtener información útil, y el problema de la calidad de esos mismos datos. En concreto, en el cerebro computacional, los datos están fuertemente afectados de incertidumbre, y es necesario tener en cuenta esta incertidumbre.

Por supuesto, el problema considerado admite otros abordajes, como pueden ser entender las señales del electroencefalograma como una serie temporal o diseñar redes neuronales profundas *ad hoc*. Así mismo, las técnicas que hemos discutido en estas páginas son, en general, aplicables a un amplio rango de problemas de la inteligencia. El cerebro computacional proporciona un marco muy apropiado para entender su relevancia sin necesidad de desarrollos técnicos muy complejos para el lector no experto. En cualquier caso, se trata de un área de investigación en plena ebullición en la actualidad y es seguro que los próximos años darán a lugar a desarrollos muy relevantes.

Referencias

Altalhi, A. H., Forcén, J. I., Pagola, M., Barrenechea, E., Bustince, H., y Takac, Z. (2019). Moderate deviation and restricted equivalence functions for measuring similarity between data. *Information Sciences*, 501, 19–29.

Bardozzo, F., De La Osa, B., Horanská, L., Fumanal-Idocin,J., delli Priscoli, M., Troiano, L., Tagliaferri, R., Fernandez, J., y Bustince, H. (2021). Sugeno integral generalization applied to improve adaptive image binarization. *Information Fusion*, 68, 37–45.

Barrenechea, E., Bustince, H., Fernández, J., Paternain, D., y Sanz, J. A. (2013). Using the Choquet integral in the fuzzy reasoning method of fuzzy rule-based classification systems. *Axioms, 2*(2), 208-223.

Beliakov, G., Bustince, H., y Calvo, T. (2016). *A practical guide to averaging functions.* Springer.

Bolón-Canedo, V., y Alonso-Betanzos, A. (2019). Ensembles for feature selection: A review and future trends. *Information Fusion,* 52, 1-12.

Bozinovski, S., y Bozinovska, L. (2019). Brain-computer interface in Europe: The thirtieth anniversary. *Automatika, 60*(1), 36–47.

Bustince, H., Beliakov, G., Dimuro, G.P., Bedregal, B., y Mesiar, R. (2017). On the definition of penalty functions in data aggregation. *Fuzzy Sets and Systems*, 323, 1–18.

Bustince, H., Fernandez, J., Kolesárová, A., y Mesiar, R. (2013). Generation of linear orders for intervals by means of aggregation functions. *Fuzzy Sets and Systems,* 220, 69–77.

Bustince, H., Mesiar, R., Fernández, J., Galar, M., Paternain, D., Altalhi, A., Dimuro, G. P., Bedregal, B., y Takac, Z. (2021). d-Choquet integrals: Choquet integrals based on dissimilarities. *Fuzzy Sets and Systems*, 414, 1–27.

Calvo, T., y Beliakov, G. (2010). Aggregation functions based on penalties. *Fuzzy Sets and Systems, 161*(10), 1420–1436.

Choquet, G. (1953-54). Theory of capacities. *Annales de l'Institut Fourier*, 5, 131–295.

Chowdhury, A., y Andreu-Perez, J. (2021). Clinical brain-computer interface challenge 2020 (cbcic at wcci2020): Overview, methods and results. *IEEE Transactions on Medical Robotics and Bionics, 3*(3), 661–670.

Dayan, P., y Abbott, L. F. (2001). *Theoretical neuroscience: computational and mathematical modeling of neural systems.* Cambridge, Mass: MIT Press.

Dimuro, G. P., Lucca, G., Bedregal, B., Mesiar, R., Sanz, J. A., Lin, C. T., y Bustince, H. (2020). Generalized CF1F2-integrals: from Choquet-like aggregation to ordered directionally monotone functions. *Fuzzy Sets and Systems*, 378, 44–67.

EU Artificial Intelligence Act. (2024). Ley de inteligencia artificial de la UE (2024). https://artificialintelligenceact.eu/es/

Fumanal-Idocin, J., Takac, Z., Fernández, J., Sanz, J. A., Goyena, H., Lin, C. T., Wang, Y.-K., y Bustince, H. (2021). Interval-valued aggregation functions based on moderate deviations applied to motor-imagery-based brain–computer interface. *IEEE Transactions on Fuzzy Systems, 30*(7), 2706–2720.

Fumanal-Idocin, J., Takac, Z., Horanská, L., Asmus, T., Dimuro, G., Vidaurre, C., Fernandez, J., y Bustince H. (2022). A generalization of the Sugeno integral to aggregate interval-valued data: an application to brain computer interface and social network analysis. *Fuzzy Sets and Systems*, 451, 320–341.

Fumanal-Idocin, J., Vidaurre, C., Fernández, J., Gómez, M., Andreu-Perez, J., Prasad, M., y Bustince, H. (2024). Supervised penalty-based aggregation applied to motor-imagery based brain-computer-interface. *Pattern Recognition*, 145, 109924.

Grabisch, M., Marichal, J. L., Mesiar, R., y Pap, E. (2009). *Aggregation Functions*. Cambridge, U.K.: Cambridge University Press.

Hersche, M., Rellstab, T., Schiavone, P. D., Cavigelli, L., Benini, L., y Rahimi, A. (2018). Fast and accurate multiclass inference for MI-, BCIs using large multiscale temporal and spectral features. En *2018 26th European Signal Processing Conference (EUSIPCO)*, 1690–1694.

Ko, L.-W., Lu, Y. C., Bustince, H., Chang, Y.-C., Chang, Y., Fernandez, J., Wang, Y.-K., Sanz, J. A., Dimuro, G. P., y Lin, C.-T. (2019). Multimodal fuzzy fusion for enhancing the motor imagery-based brain computer interface. *IEEE Computational Intelligence Magazine, 14*(1), 96–106.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., y Lance, B. J. (2018). EEGnet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering, 15*(5), 056013.

Lindsay, G. (2021). *Models of the mind: How physics, engineering and mathematics have shaped our understanding of the brain.* Bloomsbury Sigma.

Lourenzutti, R., Krohling, R. A., y Reformat, M. Z. (2017). Choquet based TOPSIS and TODIM for dynamic and heterogeneous decision making with criteria interaction. *Information Sciences*, 408, 41-69.

Lucca, G., Sanz, J. A., Dimuro, G.P., Bedregal, B., Mesiar, R., Kolesárová, A., y Bustince, H. (2016). Preaggregation functions: construction and an application. *IEEE Transactions on Fuzzy Systems, 24*(2), 260–272.

McCarthy, J., Minsky, M. L., Rochester, N., y Shannon, C. (1955). *A proposal for the Dartmouth summer research project on artificial intelligence.* http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf

Medler, D. A. (1998). A Brief History of Connectionism. *Neural Computing Surveys,* 1, 61–101.

Neuralink, Musk, E. (2019). An integrated brain-machine interface platform with thousands of channels. *bioRxiv* 10.1101/703801.

Paternain, D., Fernández, J., Bustince, H., Mesiar, R., y Beliakov, G. (2015). Construction of image reduction operators using averaging aggregation functions. *Fuzzy Sets and Systems*, 261, 8–111.

Pfurtscheller, G., Guger, C., y Ramoser, H. (1999). EEG-based brain-computer interface using subject-specific spatial filters. *Engineering applications of bio-inspired artificial neural networks*, Lecture Notes in Computer Science, 1607/1999, 248-254.

Polikov, V. S., Tresco, P. A.,y Reichert W. M. (2005). Response of brain tissue to chronically implanted neural electrodes. *Journal of Neuroscience Methods, 148*(1) 1–18.

Regalado, A. (2020). Elon Musk's Neuralink is neuroscience theater. *MIT Technology Review.* www.technologyreview.com/2020/08/30/1007786/elon-musks-neuralink-demo-update-neuroscience-theater/

Sejnowski, T. J., y Churchland, P. S. (1992). *The computational brain.* Cambridge, Mass: MIT Press.

Smith, R. (2012). Stratified, personalised, or precision medicine. *British Medical Journal.* https://blogs.bmj.com/bmj/2012/10/15/richard-smith-stratified-personalised-or-precision-medicine/

Sugeno, M. (1974). Theory of fuzzy integrals and its applications. *Doctoral Thesis.* Tokyo Institute of Technology.

Tibor, S. R., Tobias, S. J., Josef, F. L. D., Martin,G., Katharina, E., Michael, T., Frank, H., Wolfram, B., y Tonio, B. (2017) Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping, 38*(11), 5391–5420.

Trappenberg, T. P. (2010). *Fundamentals of Computational Neuroscience*. Oxford University Press Inc.

Turing, A. M. (1950). I.—Computing Machinery and Intelligence. *Mind LIX*, 236, 433–460

Vidal J. J. (1973). Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering, 2*(1), 157–180.

Vijay, M., Kashyap, A., Nagarkatti, A., Mohanty, S., Mohan, R., y Krupa, N. (2020). Extreme gradient boosting classification of motor imagery using common spatial patterns. En *2020 IEEE 17th India Council International Conference (INDICON)*, 1–5.

CAPÍTULO III

Un sistema de inteligencia artificial rápido y eficiente energéticamente*

José Duato

Las redes neuronales son una herramienta de inteligencia artificial muy potente, pero adolecen de ciertas limitaciones que dan lugar a una falta de transparencia y a un elevado consumo de energía, especialmente durante el entrenamiento. En este capítulo, tras revisar las principales aportaciones de las matemáticas y la tecnología para optimizar los procedimientos asociados y reducir el consumo de energía, se propone un nuevo sistema de inteligencia artificial para atacar los problemas mencionados. Dicho sistema se basa en un modelo lineal, en el cual cada salida se expresa como una suma ponderada de las entradas, siendo los pesos ajustables. La novedad estriba en que cada entrada, en lugar de tener un solo peso asociado, dispone de varios pesos cuya aportación se suma. Además, dichos pesos no están siempre activos. De hecho, del conjunto de pesos para cada entrada solamente se activa un subconjunto de los mismos para cada muestra. Esta flexibilidad permite aplicar el nuevo sistema a problemas complejos, alcanzando una precisión similar a la de una red neuronal. Sin embargo, el sistema propuesto genera una sencilla explicación para cada muestra y consigue, además, reducir mucho los tiempos de reentrenamiento. Se presenta el diseño del nuevo sistema de inteligencia artificial y se evalúa su rendimiento y precisión.

Palabras clave: inteligencia artificial, consumo de energía, redes neuronales.

* Agradecimientos: A Enrique Quintana-Ortí (UPV), Manel Dolz (UJI) y, sobre todo, José Ignacio Mestre (UJI) por sus inestimables aportaciones en el desarrollo del sistema de inteligencia artificial descrito en este capítulo.

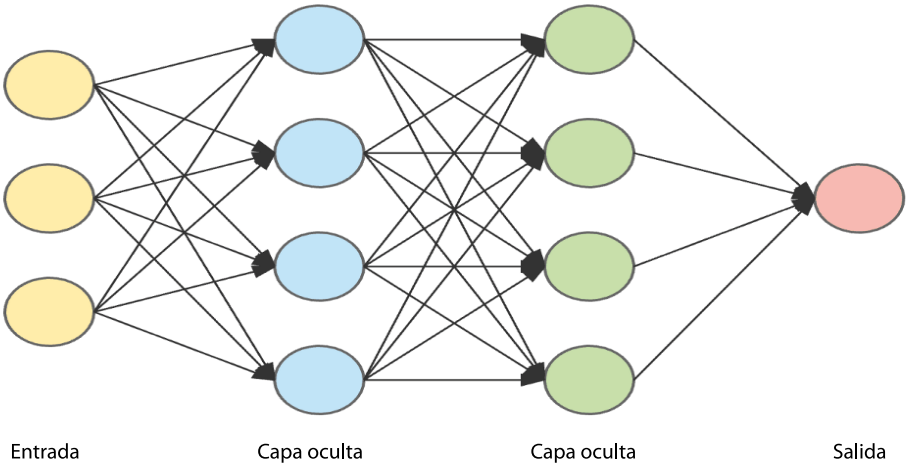
1. INTRODUCCIÓN

Las redes neuronales (Nielsen, 2019) son una herramienta de inteligencia artificial ampliamente utilizada en la actualidad para un gran número de aplicaciones. Es una herramienta muy potente, capaz de capturar la información contenida en grandes volúmenes de datos, establecer relaciones y aplicarlas posteriormente para realizar inferencia, es decir, realizar predicciones para nuevos datos no utilizados durante el entrenamiento.

La capacidad de establecer relaciones requiere que una red neuronal sea capaz de reproducir el comportamiento de cualquier función multivariable. Hay múltiples formas de conseguir esta flexibilidad, pero en el caso de las redes neuronales se ha optado por concatenar múltiples capas de unos elementos denominados neuronas artificiales (o, simplemente, neuronas) cuyo comportamiento se puede definir mediante un sencillo modelo matemático. Dicho modelo matemático consiste en aplicar una función no lineal predefinida, denominada función de activación, a la suma ponderada de las entradas de la neurona más un sesgo. Mientras que la función de activación es fija, los pesos aplicados a las entradas y el sesgo son ajustables. Las neuronas se organizan en varias capas, normalmente con patrones de interconexión regulares entre las mismas, como puede verse en la figura 1. Tradicionalmente se ha distinguido entre: a) la capa de entrada, que no contiene neuronas sino solamente los valores de las componentes de cada muestra, b) las capas ocultas, que contienen neuronas cuya función de activación puede variar de una capa a otra pero se utiliza la misma dentro de cada capa, y c) la capa de salida, con una neurona por cada salida de la red, que puede incorporar alguna función especial (por ejemplo, para convertir los valores de las diferentes salidas de la red neuronal en probabilidades, es decir, valores entre cero y uno y cuya suma total vale uno).

Figura 1.

Red neuronal con dos capas ocultas



Fuente: Elaboración propia.

Para entrenar una red neuronal se define una función de coste, que es una función que penaliza las desviaciones de los valores de salida calculados por la red para una muestra dada respecto de los valores esperados (por ejemplo, el valor medio de los cuadrados de dichas desviaciones). Seguidamente se aplica un algoritmo de optimización que, a partir de un conjunto de muestras de entrenamiento, calcula los valores de los pesos y sesgos de las neuronas que minimizan el valor de la función de coste para las muestras de dicho conjunto. Es necesario validar la precisión que es capaz de alcanzar una red neuronal entrenada, usando para ello un conjunto de muestras de validación, diferentes de las muestras de entrenamiento. Es importante que la precisión alcanzada con las muestras de validación sea similar a la alcanzada con las muestras de entrenamiento. En caso contrario, estaríamos ante un caso de sobreentrenamiento, situación que se da cuando los pesos de la red neuronal se han ajustado excesivamente a las muestras de entrenamiento, lo que obligaría a repetir el entrenamiento e incluso a buscar una configuración de red neuronal más adecuada al problema a resolver.

La gran potencia de las redes neuronales proviene de la combinación de tres factores. En primer lugar, el uso de funciones no lineales en cada una de las neuronas, lo que permite modelar comportamientos altamente no lineales. En segundo lugar, la implementación de un gran número de capas de neuronas (en particular, en las denominadas redes neuronales profundas), permite adaptar el comportamiento de la red neuronal a cualquier grupo de funciones, por complejas que sean. Desde otro punto de vista, se podría considerar que cada capa de neuronas toma pequeñas decisiones, que se basan en las decisiones de la capa anterior. Esta concatenación de pequeñas decisiones construye el equivalente a decisiones mucho más complejas. Y en tercer lugar, la capacidad de entrenamiento a partir de un conjunto de datos de entrenamiento, mediante la minimización de una función de coste. Este proceso se complica porque el número de operaciones aritméticas a realizar para entrenar una red profunda con un conjunto enorme de muestras de entrenamiento es elevadísimo, y puede llegar a tardar semanas e incluso meses en computadores muy potentes.

Debido a la no linealidad de las redes neuronales, el proceso de entrenamiento es iterativo. En cada iteración se reduce el valor de la función de coste. Los métodos más habituales se basan en calcular el vector gradiente (Nielsen, 2019), el cual indica el sentido y magnitud en los que debe modificarse cada peso de la red para progresivamente reducir el valor de la función de coste. A pesar de las numerosas optimizaciones introducidas en el proceso de entrenamiento, este proceso converge muy lentamente. Con frecuencia se requieren varias decenas de épocas, donde en cada época se procesa una vez el conjunto de entrenamiento completo. Uno de los motivos principales de esta lenta convergencia es el denominado problema del gradiente evanescente, que consiste en que algunas componentes del vector gradiente tienen valores cercanos a cero, por lo que los pesos correspondientes apenas se modifican en cada iteración. Este problema empeora con la profundidad de la red neuronal.

El problema de la lentitud, coste y consumo energético asociado que conlleva el entrenamiento se ve notablemente agravado en aquellas aplicaciones en las que se generan nuevos datos de forma frecuente y se requiere reentrenar la red neuronal para incorporar el conocimiento contenido en los nuevos datos. Las redes neuronales sufren el denominado olvido

catastrófico, que consiste en la pérdida no deseada de conocimientos ya adquiridos cuando se entrena solamente con nuevas muestras. Este comportamiento no deseado deriva del método de entrenamiento utilizado, consistente en minimizar una función de coste, ya que las muestras que no se utilicen durante el proceso de entrenamiento no influirán en los valores que se calculen para los pesos. Y este problema se agrava con el paso del tiempo, ya que los modelos utilizados son cada vez de mayor tamaño (Amodei y Hernandez, 2018).

Finalmente, otro problema completamente distinto se deriva del comportamiento de las redes neuronales como una caja negra. No se puede saber el efecto de cada peso de una red. Como consecuencia, una red neuronal no proporciona ninguna explicación de por qué se infiere una predicción determinada para una muestra dada. Este problema, que resulta insignificante en algunas aplicaciones, es de vital importancia en otras, tales como los recomendadores (por ejemplo, para conceder o denegar un préstamo) o los sistemas de ayuda al diagnóstico clínico. En estas aplicaciones, no es posible utilizar un sistema de inteligencia artificial si éste no proporciona una explicación clara del motivo por el que genera una determinada predicción.

La lentitud del entrenamiento no solo supone un tiempo de espera y un coste económico elevados. El entrenamiento con conjuntos de datos de gran tamaño combinado con redes neuronales profundas, una convergencia muy lenta y un gradiente que se desvanece generan un enorme consumo de energía. Por ejemplo, aunque estrictamente no sea una red neuronal, el entrenamiento del GPT-3 utilizado en ChatGPT requirió 1,287 gigavatios hora (The Brussels Times Newsroom, 2024). Esta cantidad es mayor que la energía generada en una planta nuclear durante una hora. Los tiempos de entrenamiento se pueden acortar mediante clústeres muy grandes equipados con potentes aceleradores tipo GPU. En el caso de ChatGPT se redujo el tiempo de entrenamiento a varios días. Si se hubiera ejecutado en una sola GPU, hubiera tardado más de un millón de horas, es decir, más de cien años.

2. REDUCCIÓN DEL CONSUMO DE ENERGÍA

Antes de plantear nuevas propuestas para reducir el consumo energético de las redes neuronales, conviene revisar los principales avances que se han producido a lo largo de varias décadas de investigación y desarrollo. Estos avances se han producido principalmente en tres frentes. Por una parte, tenemos los avances en los algoritmos matemáticos empleados, siendo los más relevantes aquellos que consiguen reducir el número de operaciones aritméticas a realizar y/o contribuyen a reducir el movimiento de datos, ya que además de reducir el consumo también mejoran las prestaciones. Por otra parte, tenemos los avances combinados en tecnología de fabricación de chips y en arquitectura de computadores, que han dado como resultado los sofisticados aceleradores de cálculo para inteligencia artificial que existen en la actualidad.

2.1. Avances en los algoritmos

Dado que las neuronas artificiales incorporan una función de activación no lineal y que el objetivo del proceso de entrenamiento es ajustar los pesos de las neuronas para que el valor de la función de coste alcance su valor mínimo para un determinado conjunto de muestras de entrenamiento, los investigadores se decantaron por sencillos métodos iterativos, tales como el método del descenso del gradiente. Éste es un algoritmo de optimización iterativo de primer orden para minimizar cualquier función multivariable diferenciable.

La idea de este algoritmo es avanzar, en cada iteración, en dirección opuesta al gradiente de la función en el punto actual, ya que ésta es la dirección en la que se reduce más rápidamente el valor de la función de coste. Se han realizado muchísimos avances en la mejora de los algoritmos de entrenamiento, pero sin duda las dos contribuciones más relevantes han sido el algoritmo de retropropagación (*backpropagation*) para el cálculo del gradiente y la versión estocástica del método de descenso del gradiente.

El algoritmo de retropropagación parte de la observación de que las expresiones matemáticas para las componentes del gradiente para una determinada capa de la red neuronal son idénticas a las expresiones para la capa posterior de la red, pero añadiendo un par de factores en cada nueva componente. Este algoritmo ha permitido que al calcular el gradiente, en lugar de calcular cada componente desde cero, se puedan reutilizar los cálculos de las componentes de una capa para calcular las componentes de la capa anterior (que es la siguiente que se calcula, ya que se procesan las capas en orden inverso). En redes neuronales profundas, este algoritmo puede llegar a reducir el número de operaciones aritméticas a menos de la centésima parte respecto al algoritmo original.

En teoría, en cada iteración se deberían incluir todas las muestras de entrenamiento en el cálculo del gradiente de la función de coste en ese punto. Sin embargo, se ha podido comprobar que basta con una pequeña parte (denominada lote o *batch*) de dicho conjunto de muestras, seleccionado aleatoriamente, para conseguir buena precisión en el cálculo del gradiente. Esta variante del algoritmo de descenso del gradiente, denominada estocástica, consigue también reducir el número total de operaciones aritméticas a menos de la centésima parte respecto a la versión no estocástica. Afortunadamente, esta reducción puede combinarse con la conseguida por el algoritmo de retropropagación, dando lugar a una reducción combinada que puede superar los cuatro órdenes de magnitud (es decir, reducir el número de operaciones en un factor de 10.000).

Hay muchas otras optimizaciones que consiguen resultados menos vistosos, pero no por ello menos importantes, ya que todos contribuyen a mejorar la eficiencia energética. Por citar algunas de estas optimizaciones:

- Proceso por lotes. En lugar de procesar secuencialmente las muestras de un lote, se pueden procesar todas a la vez. Al procesar lotes de muestras, los productos matriz por vector que deben ejecutarse durante el entrenamiento de una muestra se con-

vierten en productos matriz por matriz. Y este cambio es aprovechado por los aceleradores actuales para conseguir unas prestaciones mucho más elevadas y un consumo de energía mucho menor.

- Funciones de activación más sencillas, como por ejemplo la ReLU (Rectified Linear Unit) (ver figura 5). Estas funciones y sus derivadas son más rápidas de calcular y consumen menos energía que otras funciones tradicionales tales como la sigmoide.
- Transferencia de conocimiento. En algunas aplicaciones es posible utilizar modelos entrenados para una aplicación en otra aplicación diferente. De este modo, un modelo puede basarse en conocimientos previos para dominar nuevas tareas, y se puede seguir entrenando el modelo a pesar de tener datos limitados.
- Métodos de poda. Permiten eliminar muchas neuronas sin perder apenas precisión, especialmente cuando la red neuronal estaba sobredimensionada, lo que normalmente redunda en una reducción del número de operaciones a realizar. Pero si la poda no es estructurada, puede resultar difícil aprovecharla debido a la arquitectura interna de los aceleradores actuales.
- Métodos de inicialización de los pesos. Una inicialización de los pesos que reduzca la probabilidad de obtener valores muy pequeños en el cálculo de las componentes del gradiente normalmente conseguirá reducir el número de iteraciones necesarias durante el entrenamiento.
- Hiperparámetros variables dinámicamente. Los hiperparámetros son parámetros globales del sistema o del algoritmo de entrenamiento. Su valor puede ser fijado por el usuario. Algún hiperparámetro, como la velocidad de aprendizaje (que es el factor que se aplica al gradiente para determinar cuánto se progresa en cada iteración del entrenamiento) es crucial, pudiendo afectar mucho al número de iteraciones necesarias durante el entrenamiento, y pudiendo hacer incluso que el entrenamiento no converja ni termine nunca. Una optimización consiste en variar dinámicamente el valor de este hiperparámetro para conseguir los mejores resultados.

2.2. Avances en la tecnología

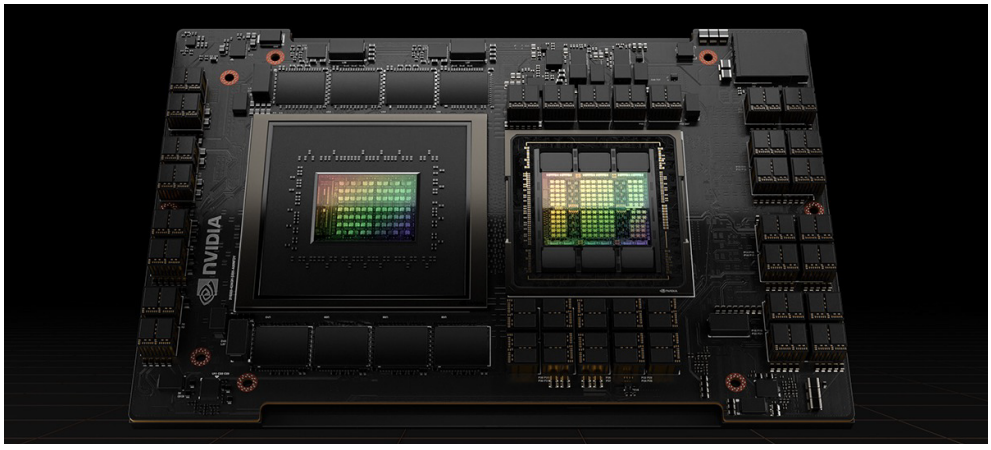
Aunque los avances en los algoritmos han sido trascendentales para conseguir unos requisitos de potencia de cálculo y unos niveles de eficiencia energética que hacen viables los sistemas de inteligencia artificial actuales, su contribución queda ensombrecida por la descomunal magnitud de los avances en la tecnología. Para tener una idea de conjunto de la aportación de la tecnología y su evolución a lo largo de las últimas ocho décadas, basta con comparar el supercomputador actual con mayor eficiencia energética con el primer ordenador electrónico.

El ordenador más eficiente es el que encabeza la lista del Green500 en junio de 2024 (Green500, 2024), el cual es capaz de ejecutar 72,7 GigaFLOPS por vatio. Está basado en el superchip de NVIDIA denominado GH200 Grace Hopper Superchip (NVIDIA GH200

Grace Hopper Superchip, 2024), cuya imagen sin disipador de calor puede verse en la figura 2. Si comparamos este ordenador con el primer ordenador electrónico, el ENIAC, consigue una mejora de la eficiencia energética en un factor de 36 billones ($3{,}6\times10^{13}$). Esta impresionante mejora, que es más de mil millones de veces mayor que la alcanzada por las principales mejoras en los algoritmos, se ha conseguido combinando tecnología (escala de integración de los chips, familia lógica CMOS) y arquitectura (por ejemplo, los aceleradores tipo GPU).

Figura 2.

NVIDIA GH200 Grace Hopper Superchip



Fuente: Extraída de https://www.nvidia.com/es-es/data-center/grace-hopper-superchip/

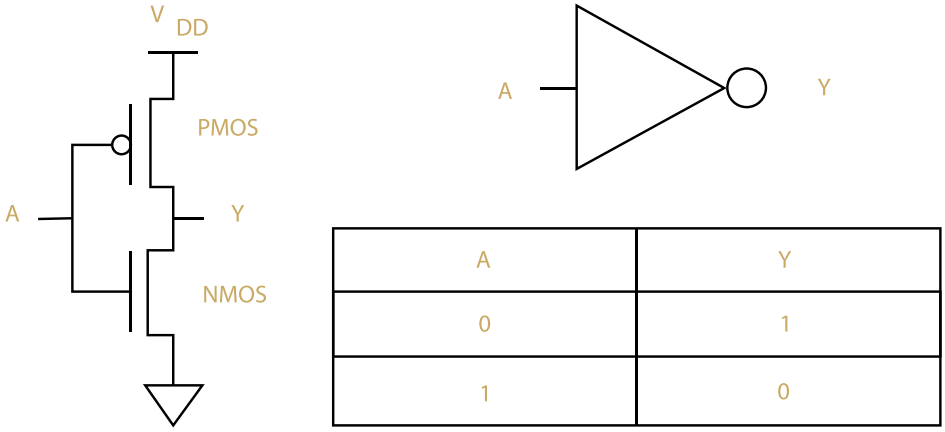
Por otra parte, la elevada escala de integración de los transistores ha permitido pasar de un tamaño de transistor de 5 mm a 3 nm, lo que ha supuesto un factor de reducción en el tamaño superior a un millón de veces. Como los transistores se fabrican sobre chips de dos dimensiones, la integración ha permitido empaquetar hasta 200.000 millones de transistores, como es el caso del GH200 Grace Hopper Superchip (y mucho más, hasta 4 billones de transistores, si se utiliza una oblea entera de silicio, como es el caso de los chips WSE-3 de Cerebras). La menor distancia entre transistores ha permitido además reducir el consumo de energía para mover datos y subir la frecuencia de reloj de MHz a GHz.

Un componente importante de la tecnología es la familia lógica utilizada en los chips. En la década de los 80 existían muchas familias lógicas con diferentes características, desde la rápida ECL, pasando por la popular TTL, las PMOS y NMOS utilizadas en los primeros microprocesadores, hasta la lenta CMOS de muy bajo consumo. De ellas, únicamente queda la CMOS, que es la que consigue los consumos más bajos de energía, y que ha mejorado muchísimo sus prestaciones a lo largo de cuatro décadas. Como puede verse en la figura 3, gracias al uso de transistores MOS complementarios, cuando un transistor conduce, el otro no lo hace, consiguiendo así unos consumos bajísimos. No obstante, a pesar de su bajo con-

sumo, cuando se integran 200.000 millones de transistores en un chip, y a pesar de implementar múltiples optimizaciones para minimizar el consumo de energía, dicho chip puede llegar a requerir 500 W de potencia.

Figura 3.

Puerta lógica CMOS: inversor



A	Y
0	1
1	0

Fuente: Extraída de https://www.geeksforgeeks.org/cmos-logic-gate/

Es importante mencionar que cuando hablamos de tecnología, muchas veces nos olvidamos de la aportación de la arquitectura de los computadores, que es la ciencia que se ocupa de la organización de los componentes y sus interconexiones, tanto dentro un chip como de los chips entre sí, para conseguir objetivos diversos tales como elevadas prestaciones, bajo consumo, buena fiabilidad, etc. Para hacernos una idea de la importancia de la arquitectura del chip, basta con comparar dos chips actuales (que usan tecnologías de fabricación similares) con arquitecturas muy diferentes. Si nos fijamos en un procesador (CPU) de uso general, encontramos que uno de los más rápidos (AMD Ryzen 9 7950X3D 16-Core) es capaz de ejecutar 7,29 GigaFLOPS por núcleo (lo que conseguiría un pico de 116,64 GigaFLOPS con los 16 núcleos), mientras que un acelerador tipo GPU diseñado específicamente para inteligencia artificial alcanza 7,1 TeraFLOPS, es decir, más de 60 veces más rápido. Pero no solamente es mucho más rápido sino que también es mucho más eficiente energéticamente, en una proporción bastante similar.

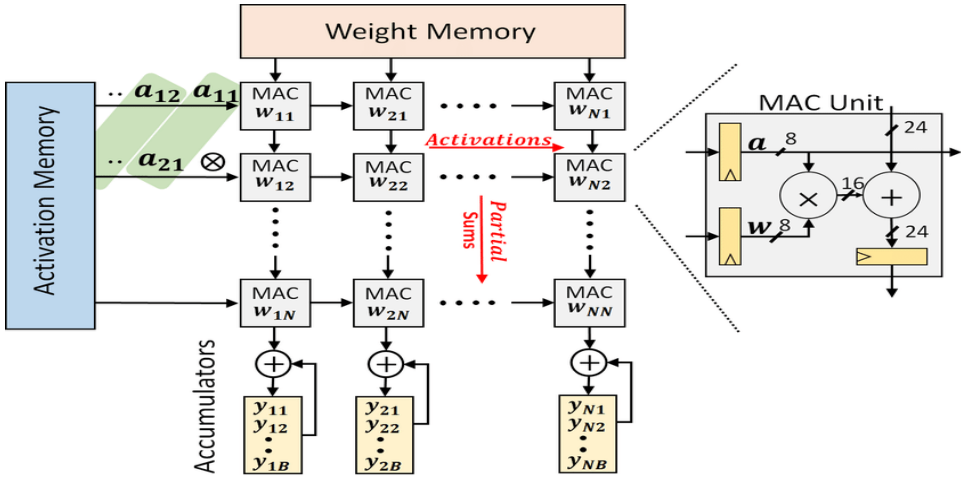
El punto de partida para que una GPU consiga mucha más potencia de cálculo que una CPU es que dedica un porcentaje mucho más grande de los transistores a las unidades aritméticas, con lo que en una GPU hay un número mucho más grande de estas unidades. Pero con ello no se conseguiría ni mayor eficiencia energética ni tampoco mayor potencia de cálculo, ya que la memoria no tendría suficiente ancho de banda para poder alimentar con datos dichas unidades aritméticas. Así pues, se requiere una arquitectura diferente.

Los aceleradores actuales (sean GPUs u otras variantes) son energéticamente mucho más eficientes que una CPU gracias a que reutilizan muchas veces cada dato leído de memoria. Su arquitectura deriva del concepto de procesador sistólico, en el cual:

- Las unidades aritméticas de procesamiento se organizan de forma estructurada, con frecuencia formando una estructura bidimensional en la que cada unidad se conecta únicamente a sus vecinas (en la figura 4, las vecinas de la derecha y de abajo).
- Los datos avanzan de forma sincronizada, de modo que cada dato se mueve de una unidad a la siguiente a cada ciclo de reloj. Varias filas y varias columnas de datos se mueven a la vez.
- Cada vez que un dato que avanza en vertical coincide con otro que avanza en horizontal, se opera con dichos datos en la unidad aritmética correspondiente (en la figura 4, una multiplicación seguida de una suma).

Figura 4.

Procesador sistólico para redes neuronales



Fuente: Extraída de https://arxiv.org/html/2402.18595v2

Existen muchas variantes de estos procesadores sistólicos. En inteligencia artificial se utilizan fundamentalmente para realizar productos de matrices (por ejemplo, la matriz de pesos de las neuronas de una capa por la matriz de activaciones, que son las salidas de la capa anterior para las diferentes muestras de un lote), motivo por el cual cada unidad aritmética ejecuta una multiplicación seguida de una suma. Por otra parte, la figura muestra una variante en la cual se precarga la matriz de pesos en registros internos de las unidades aritméticas, y posteriormente se introduce ciclo a ciclo la matriz de activaciones por la izquierda y el resultado del producto va saliendo, también ciclo a ciclo, por la parte inferior.

Las elevadas prestaciones y eficiencia energética se consiguen porque: 1) cada dato leído de memoria se procesa muchas veces sin tener que traerlo de nuevo de memoria; y 2) cada dato solamente tiene que viajar de una unidad a la vecina para poder realizar las siguientes operaciones con dicho dato.

Además del uso de procesadores sistólicos, se utilizan otras técnicas adicionales para aumentar la velocidad de procesamiento y reducir el consumo de energía aún más. Por ejemplo, se utilizan operaciones aritméticas de muy baja precisión, llegando a representar un número en coma flotante con 16 bits (FP16) e incluso con solo 8 bits (FP8). Los aceleradores actuales incorporan soporte hardware para estos formatos.

Otro componente en el que se han desarrollado importantes mejoras en la tecnología y la arquitectura es la memoria. Se han desarrollado técnicas de fabricación de chips con integración 3D. Consiste en apilar chips dentro de un mismo encapsulado, aumentando así el número de transistores que se pueden usar. Se aplica a las memorias, ya que éstas disipan poco calor, y permite poner los chips más cerca unos de otros y reducir así mucho el movimiento de datos. En la actualidad, se están integrando en un mismo encapsulado un acelerador tipo GPU junto con una pila (3D) de chips de memoria, consiguiendo reducir aún más la distancia recorrida por los datos y aumentar notablemente el ancho de banda de memoria.

2.3. Margen de mejora de la tecnología

Tras repasar las impresionantes mejoras desarrolladas en la tecnología y en la arquitectura de los chips cabe plantearse si se puede hacer algo más. La respuesta es afirmativa, pero el margen de maniobra es ya muy escaso. La escala de integración de los chips está tocando el techo. Además, una mayor integración ya no reduce el consumo de energía sino que lo aumenta (por ejemplo, debido a las corrientes de fugas como consecuencia de que las capas aislantes de los transistores son muy finas). Por otra parte, los aceleradores están muy optimizados para minimizar el movimiento de datos y reutilizar datos al máximo. Finalmente, para fabricar transistores aún más pequeños, solamente queda el carbono por encima del silicio en la tabla periódica. Investigadores del MIT (Massachusetts Institute of Technology) han demostrado que los transistores de efecto de campo (FET) fabricados con nanotubos de carbono son más eficientes energéticamente que los transistores FET de silicio y que pueden ser fabricados en grandes cantidades en las plantas de fabricación de chips de silicio, usando obleas de tamaño estándar (Becky Ham, 2020). Pero de momento, aún no se están comercializando.

3. OPORTUNIDAD DE AHORRO DE ENERGÍA

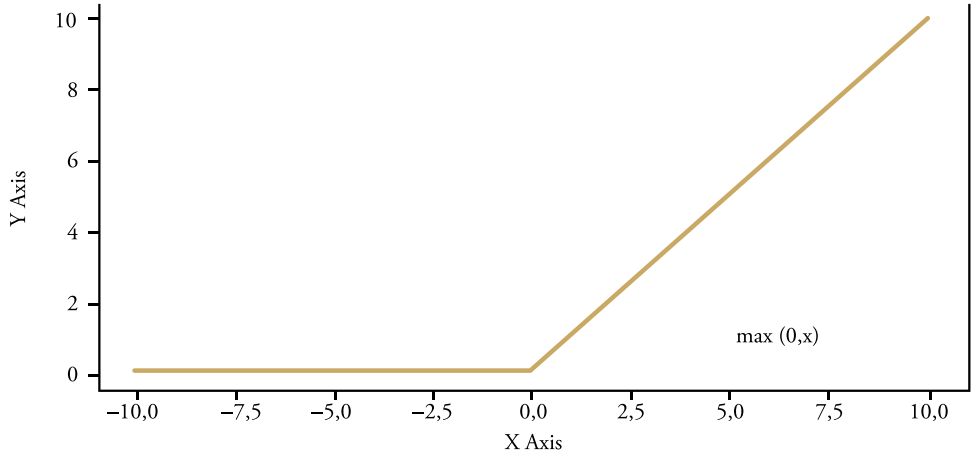
Una línea de mejora poco explorada consiste en centrarse en aplicaciones que requieren un reentrenamiento frecuente y diseñar métodos de reentrenamiento incremental que no sufran olvidos catastróficos. En aplicaciones que requieran un reentrenamiento frecuente,

este enfoque tendría un potencial de ahorro superior a 1.000 veces. Lamentablemente, el reentrenamiento incremental no funciona en redes neuronales debido al olvido catastrófico. Esto se debe a que las muestras que no se procesan durante la optimización de la función de coste no contribuyen al entrenamiento de los pesos de la red neuronal.

Para buscar una solución a este problema, consideremos una red neuronal cuyas neuronas implementan la función de activación ReLU, representada en la figura 5. Para valores de entrada negativos o nulos la función ReLU tiene una salida nula y para valores positivos se comporta como la función identidad. Cuando la salida de la función ReLU es nula, la neurona correspondiente no aporta nada a ninguna salida de la red neuronal y diremos que dicha neurona está inactiva. En cambio, cuando la ReLU se comporta como la función identidad hace que la neurona correspondiente tenga un comportamiento lineal. Recordemos que, en tal caso, la neurona simplemente calcula la suma ponderada de las entradas más el sesgo. En ese caso, diremos que la neurona está activa.

Figura 5.

Función de activación Rectied Linear Unit (ReLU)



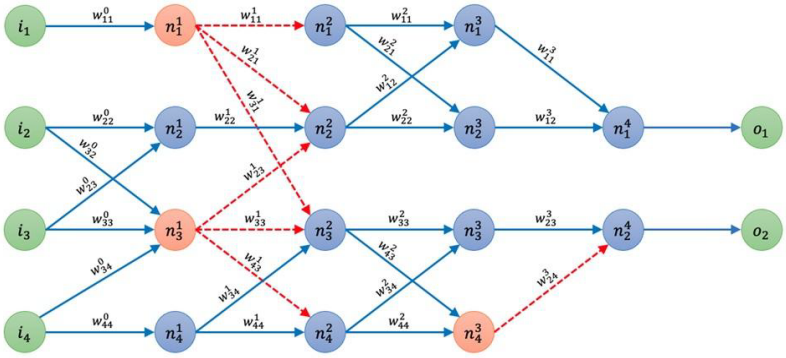
Fuente: Elaboración propia.

Así pues, cuando aplicamos una muestra a las entradas de una red neuronal, ciertas neuronas se activarán y otras quedarán inactivas. La figura 6 muestra esta situación para una pequeña red de ejemplo, en la que las neuronas activas se representan en color azul y las inactivas en rojo. Si queremos calcular el valor de una salida de la red neuronal (por ejemplo, o_1) para la muestra aplicada, hay que tener en cuenta que es la salida de la neurona n_1^4, que está activa, por lo que $o_1 = w_{11}^3 a_1^3 + w_{12}^3 a_2^3 + b_1^4$, donde a_1^3 y a_2^3 son las salidas de las neuronas n_1^3 y n_2^3, respectivamente, y b_1^4 es el sesgo de la neurona n_1^4. Si ahora sustituimos de forma recursiva el valor de cada salida de una neurona activa por su expresión matemática, obtenemos la expresión mostrada en la figura 6. El valor de la salida o_2 se calcularía de forma análoga.

Como puede observarse en la figura 6, cuando hacemos inferencia para una muestra dada, es decir, cuando aplicamos dicha muestra a la red, cada salida puede expresarse como una combinación lineal de las entradas más varios términos independientes. Por ejemplo, el rectángulo azul muestra los términos que relacionan la entrada i_2 con la salida o_1 y el rectángulo verde muestra los términos independientes aportados por la neurona n_1^2. Como detalle adicional, podemos observar que los coeficientes de cada término de la expresión para una salida de la red se pueden calcular como producto de los pesos a lo largo de la ruta desde la entrada hasta la salida correspondiente, o como producto de los pesos a lo largo de la ruta desde una neurona hasta la salida correspondiente (multiplicado por el sesgo de dicha neurona) en el caso de los términos independientes. En resumen, el valor de una salida de la red se puede calcular como la suma de las aportaciones de las diferentes rutas activas para la muestra dada, donde una ruta activa es aquella que tiene todas sus neuronas activas desde el punto de inicio hasta la salida correspondiente. El punto de inicio de una ruta puede ser una entrada o una neurona de una capa oculta o de salida.

Figura 6.

Cálculo del valor de una salida de una red neuronal con ReLU para una muestra dada



$$
\begin{aligned}
o_1 \;=\;& b_1^4 \\
&+\; w_{11}^3 b_1^3 + \boxed{w_{11}^3 w_{11}^2 b_1^2} + w_{11}^3 w_{12}^2 b_2^2 + w_{11}^3 w_{12}^2 w_{22}^1 b_2^1 + \boxed{w_{11}^3 w_{12}^2 w_{22}^1 w_{22}^0 i_2} + w_{11}^3 w_{12}^2 w_{22}^1 w_{23}^0 i_3 \\
&+\; w_{12}^3 b_2^3 + \boxed{w_{12}^3 w_{21}^2 b_1^2} + w_{12}^3 w_{22}^2 b_2^2 + w_{12}^3 w_{22}^2 w_{22}^1 b_2^1 + \boxed{w_{12}^3 w_{22}^2 w_{22}^1 w_{22}^0 i_2} + w_{12}^3 w_{22}^2 w_{22}^1 w_{23}^0 i_3
\end{aligned}
$$

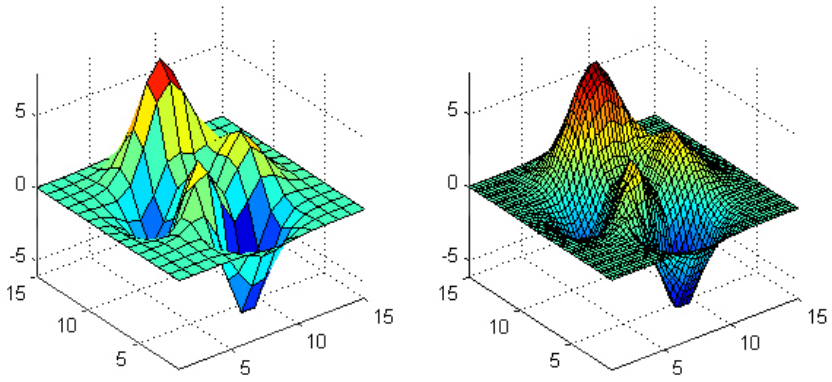
Fuente: Elaboración propia.

Es más, ante muy pequeñas variaciones en los valores de entrada respecto a los correspondientes a la muestra aplicada, el estado de activación de las neuronas de la red no cambiará, por lo que las expresiones de las salidas tampoco cambiarán y la red neuronal se comportará como un modelo lineal. En cambio, cuando las variaciones en los valores de entrada sean de mayor magnitud y cambie el estado de activación de una o varias neuronas, el resultado será otro modelo lineal diferente, análogo al mostrado pero con otros valores para los coeficientes del modelo lineal. Este comportamiento es bien conocido y se dice que una red neuronal cuyas neuronas implementan la función de activación ReLU genera salidas lineales por tra-

mos. De este modo, cualquier salida de la red puede generar funciones tan complejas como sea necesario, para ajustar así la relación entre los valores de entrada y de salida de las muestras. Si se quiere obtener mayor precisión en el ajuste, basta con hacer los tramos lineales más cortos, lo que se consigue utilizando una red neuronal más profunda (ver figura 7).

Figura 7.

Precisión de una función de salida aproximada mediante tramos lineales



Fuente: Elaboración propia.

4. NUEVO SISTEMA DE INTELIGENCIA ARTIFICIAL

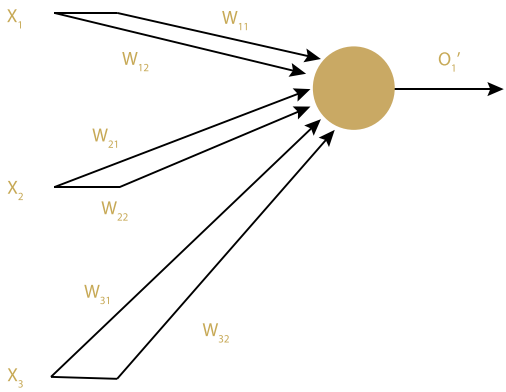
Hemos visto que una red neuronal con la función de activación ReLU selecciona un subconjunto de las neuronas de la red (las que hemos denominado neuronas activas) para cada muestra. En general, dicha selección variará de una muestra a otra. En base a estas observaciones se puede desarrollar un nuevo sistema de inteligencia artificial basado en un modelo lineal, pero que se comporta como un sistema no lineal y emula con precisión las redes neuronales basadas en la función de activación ReLU. Para conseguir dicho comportamiento no lineal, el nuevo sistema deberá seleccionar para cada muestra un subconjunto de los parámetros del modelo lineal (Duato *et al.,* 2023). Si las selecciones de parámetros son diferentes para diferentes muestras, dicho sistema también generará salidas lineales por tramos.

Al tratarse de un modelo lineal, dicho sistema puede representarse mediante una red neuronal de una sola capa, con una sola neurona por cada salida del sistema, y cuyas neuronas no tienen función de activación para que tengan un comportamiento lineal. Sin embargo, este diseño tan sencillo no permitiría suficiente flexibilidad como para poder aproximar el comportamiento de una red neuronal profunda de forma precisa. Para dotar a este sistema de mayor flexibilidad a la hora de elegir subconjuntos del mismo para cada muestra, basta con que cada neurona tenga varios pesos asociados a cada entrada, en lugar de un

solo peso, como puede verse en la figura 8. Por simplicidad, en dicha figura se han representado únicamente dos pesos asociados a cada entrada de la neurona. Cada uno de esos pesos se puede seleccionar o no, de forma individual, para cada muestra. Esto da lugar a cuatro combinaciones por entrada. Por ejemplo, para la entrada X_1, podemos no seleccionar ningún peso (en cuyo caso esa entrada no influirá en la salida), seleccionar w_{11}, seleccionar w_{12} o seleccionar ambos pesos (en cuyo caso su aportación se suma). Nótese que la selección de pesos para cada entrada es independiente de la selección para el resto de entradas. Así pues, en la figura 8, a pesar de su simplicidad, se pueden generar 4×4×4=64 combinaciones diferentes de selección de pesos. El número de combinaciones también puede calcularse como 2^p, siendo p el número de pesos que pueden ser seleccionados individualmente (6 en este ejemplo).

Figura 8.

Ejemplo de neurona con 3 entradas y 2 pesos por entrada



Fuente: Elaboración propia.

Por ejemplo, si se usan 10 pesos por entrada, se puede elegir entre 1024 combinaciones diferentes para distintas muestras, y dichas combinaciones son independientes de las que se pueden elegir para el resto de entradas. De este modo se puede conseguir una gran flexibilidad y obtener un comportamiento análogo al de una red neuronal con la función de activación ReLU.

Los principales beneficios del enfoque propuesto son:

- Excelente interpretabilidad, ya que directamente se genera un modelo lineal para cada muestra.
- Se pueden fusionar fácilmente varias copias de un modelo lineal determinado, sin más que calcular la media ponderada parámetro a parámetro. Como factor de ponderación de los parámetros de cada modelo se puede usar el número de muestras utilizadas para entrenar dicho modelo.

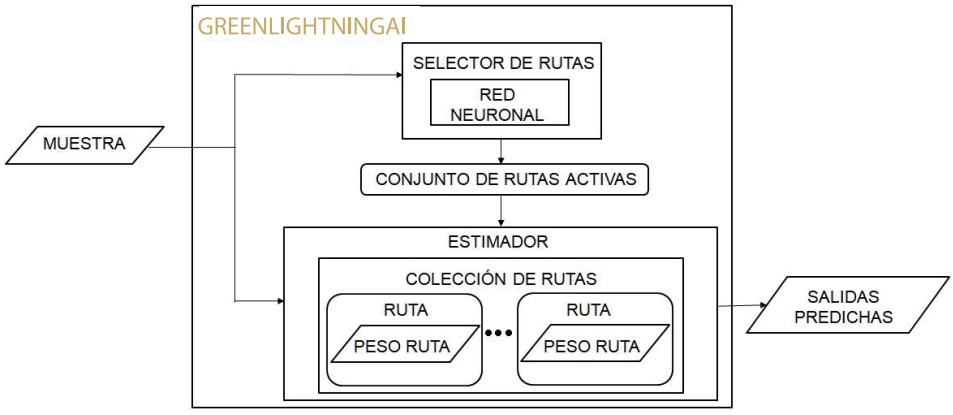
- La fusión de modelos permite realizar reentrenamiento incremental, entrenamiento federado, reentrenamiento incremental federado, etc.

Sin embargo, queda un importante problema por resolver, que es la selección de un subconjunto del modelo lineal para cada muestra. Dicha selección debe cumplir una importante propiedad: Dado un par de muestras, cuanto más similares sean las muestras más similares deben ser los correspondientes subconjuntos de parámetros seleccionados.

Este problema se puede resolver de diversas formas. En una primera versión se ha resuelto utilizando una pequeña red neuronal, con función de activación ReLU, para realizar la selección (Duato *et al.,* 2023)(ver figura 9). Seguidamente se describe la integración de dicha red neuronal con el modelo lineal. En cualquier red neuronal se pueden identificar una serie de rutas posibles desde cualquier entrada a cualquier salida. Es lo que denominamos colección de rutas, e incluye todas las rutas posibles definidas por los patrones de conexión entre las capas de la red. El modelo lineal se dimensiona a partir de esta colección de rutas, de modo que se define un peso entrenable (y solamente uno) por cada ruta posible en la red neuronal que se va a usar para seleccionar subconjuntos del modelo lineal.

Figura 9.

Diagrama de bloques del nuevo sistema de inteligencia artificial



Fuente: Elaboración propia.

Cuando una red neuronal está basada en la función de activación ReLU, para una muestra dada se activan algunas de esas rutas y otras no, como se ha visto en el ejemplo de la figura 6. El conjunto de rutas activas para una muestra dada es lo que hace falta calcular en el sistema propuesto, enviando dicha información al modelo lineal, denominado estimador, el cual seleccionará exclusivamente el subconjunto de pesos del modelo que corresponden a las rutas activas. Seguidamente, dicho subconjunto se puede utilizar para

la operación deseada con dicha muestra, ya sea realizar inferencia o procesar dicha muestra como parte de un proceso iterativo de entrenamiento. El diagrama de bloques del nuevo sistema de inteligencia artificial se muestra en la figura 9.

La red neuronal utilizada para calcular el conjunto de rutas activas para cada muestra, denominada selector de rutas, se puede inicializar entrenándola previamente con cualquier método de entrenamiento. De hecho, es posible entrenar el selector de rutas con una pequeña fracción de las muestras de entrenamiento y con pocas iteraciones. El motivo es que no necesita generar una salida precisa. Basta con que discrimine adecuadamente entre diferentes muestras. Es más, en general no hace falta actualizar el selector de rutas cuando se reentrena el sistema propuesto, por el mismo motivo. Para reentrenar el sistema, basta con reentrenar los pesos del estimador, para lo cual puede usarse cualquier método de entrenamiento habitualmente utilizado para entrenar redes neuronales (ya que en realidad es equivalente a una red neuronal de una capa).

5. VERIFICACIÓN EXPERIMENTAL

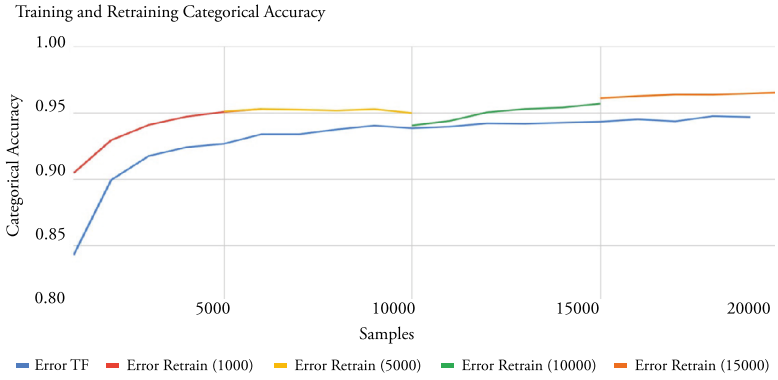
Con objeto de verificar que el selector de rutas no requiere ser reentrenado cuando se reentrenan los pesos del modelo lineal (es decir, del estimador), se ha llevado a cabo un sencillo experimento, comparando una red neuronal de dos capas con 300 y 10 neuronas, respectivamente, con el nuevo sistema de inteligencia artificial configurado de modo que tenga aproximadamente el mismo número de pesos entrenables que la red neuronal. Como conjunto de datos de entrenamiento y validación se ha utilizado el popular MNIST (Lecun *et al.,* 2010), que es un conjunto de imágenes de dígitos manuscritos.

Tanto la red neuronal como el selector de rutas y el estimador del nuevo sistema se han entrenado utilizando el método estocástico de descenso del gradiente. La figura 10 muestra la precisión obtenida entrenando con diferentes números de muestras. La curva azul corresponde a la red neuronal convencional. La curva roja corresponde al nuevo sistema, en el que el selector de rutas ha sido entrenado inicialmente con 1.000 muestras y no se ha vuelto a reentrenar, entrenando únicamente el estimador para los diferentes puntos de la curva. Las curvas amarilla, verde y naranja también muestran resultados en los que se entrena solamente el estimador con tamaños crecientes del conjunto de muestras, habiendo inicializado previamente el selector de rutas con 5.000, 10.000 y 15.000 muestras, respectivamente.

Como puede verse en la figura 10, el nuevo sistema consigue una precisión comparable a la de una red neuronal convencional, a pesar de que el selector de rutas se haya inicializado con un número menor de muestras que las utilizadas para entrenar el estimador. Es más, la reinicialización del selector de rutas con un mayor número de muestras (5.000, 10.000 y 15.000 muestras) no parece aportar ningún beneficio, lo que prueba que basta con entrenar inicialmente dicho selector de rutas con un porcentaje reducido de las muestras (salvo en aquellas aplicaciones en las que las muestras puedan variar mucho a lo largo del tiempo).

Figura 10.

Evaluación de la necesidad de reentrenar el selector de rutas



Fuente: Elaboración propia.

6. EVALUACIÓN DE PRESTACIONES

Con objeto de evaluar los beneficios aportados por el nuevo sistema de inteligencia artificial, se han realizado numerosas comparaciones con redes neuronales convencionales. Las figuras 11 y 12 muestran el tiempo de ejecución relativo y la precisión obtenida en validación cuando comparamos una red neuronal profunda de 152 capas (ResNet152) con un diseño basado en el nuevo sistema de inteligencia artificial con un número similar de parámetros entrenables. El conjunto de datos es ImageNet (Stanford Vision Lab, 2021), un conjunto de más de 1.200.000 imágenes de 1.000 clases de objetos. Nótese que el consumo de energía es aproximadamente proporcional al tiempo de ejecución.

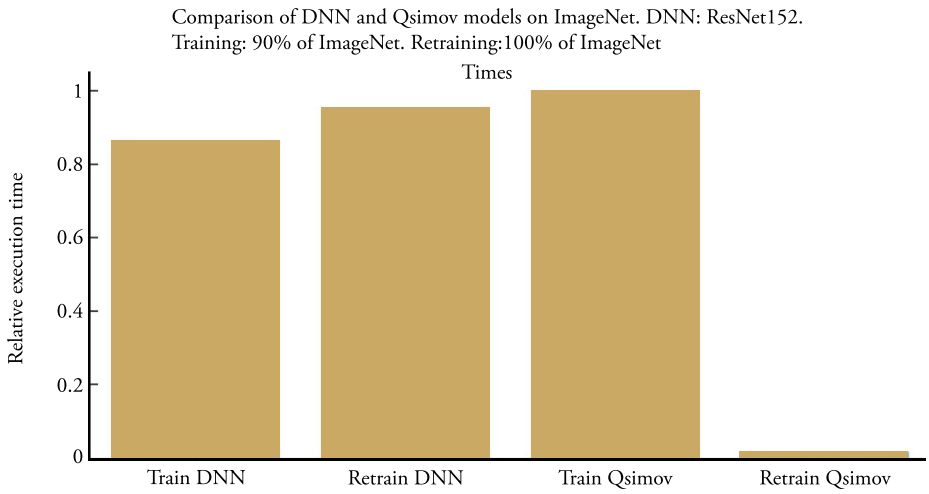
Para evaluar los beneficios del reentrenamiento incremental, que solamente es posible con el nuevo sistema, se ha realizado primero un entrenamiento con el 90 % de ImageNet, seguido de un reentrenamiento con el resto de imágenes. Dicho reentrenamiento solamente ha tenido que procesar el 10 % restante de ImageNet en el caso del nuevo sistema, manteniendo inalterado el selector de rutas y actualizando únicamente los pesos del estimador. Sin embargo, en el caso de la red ResNet152 ha sido necesario procesar el 100 % de ImageNet debido al olvido catastrófico.

Las cuatro barras de cada figura corresponden al entrenamiento de la red neuronal (DNN), el reentrenamiento de dicha red, el entrenamiento del nuevo sistema (Qsimov) y el reentrenamiento incremental del mismo. Como puede observarse en la figura 11, la capacidad de reentrenamiento incremental supone una enorme reducción en el tiempo de procesamiento y en la correspondiente energía consumida. La reducción en el tiempo de ejecución no solo se debe a que el nuevo sistema solamente necesita procesar las muestras nuevas. El diseño del estimador como una red de una sola capa sin función de activación hace que la convergencia del proceso iterativo de entrenamiento sea mucho más rápida que en una red

neuronal profunda. Como puede observarse en la figura 12, a pesar de procesar solamente las muestras nuevas durante el reentrenamiento y actualizar únicamente los pesos del estimador, la precisión conseguida por el nuevo sistema de inteligencia artificial es prácticamente idéntica a la obtenida por la red neuronal.

Figura 11.

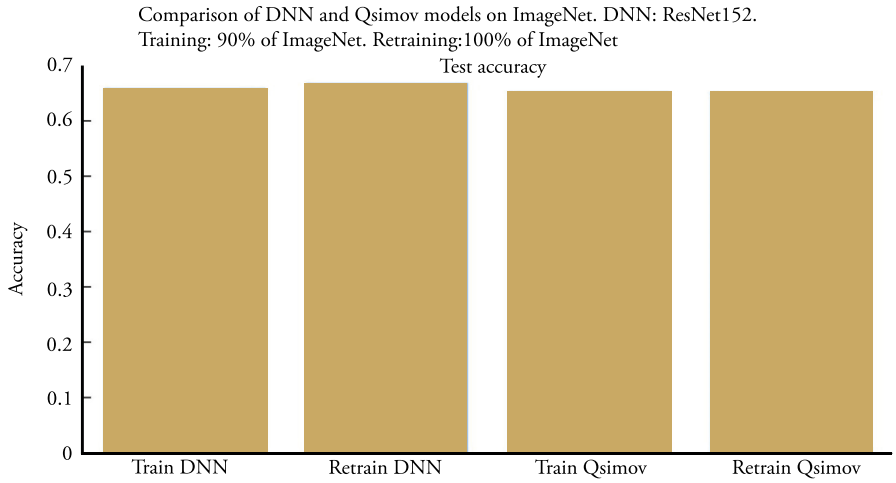
Evaluación comparativa de los tiempos de entrenamiento y reentrenamiento



Fuente: Elaboración propia.

Figura 12.

Evaluación comparativa de las precisiones alcanzadas



Fuente: Elaboración propia.

7. CONCLUSIONES

En este artículo se ha presentado una revisión sucinta de los grandes avances en los métodos matemáticos y en los algoritmos asociados para reducir el número de operaciones aritméticas necesarias para entrenar redes neuronales, con la consiguiente reducción en los tiempos de entrenamiento y en el coste y consumo de energía asociados. También se han revisado las aportaciones realizadas por los avances en la tecnología y en la arquitectura de los computadores, mostrando que su impacto en la reducción del consumo de energía es muchos órdenes de magnitud superior a la reducción lograda por los métodos matemáticos.

Dado que los márgenes de mejora son cada vez más pequeños, se ha propuesto y evaluado un nuevo sistema de inteligencia artificial orientado específicamente a reducir el coste y consumo de energía en aquellas aplicaciones que requieran un reentrenamiento frecuente. Para ello, el nuevo sistema se ha diseñado a partir de un modelo lineal, de modo que permite combinar rápidamente diferentes modelos con la misma arquitectura. Los resultados de evaluación muestran la enorme reducción en los tiempos de ejecución que se consigue al reentrenar con el nuevo sistema, sin que ello afecte a la precisión del modelo.

Referencias

Amodei, D., y Hernandez, D. (2018). Ai and compute. OpenAI. https://openai.com/blog/ai-and-compute

Becky Ham (MIT News correspondent). (2020). MIT News. https://news.mit.edu/2020/carbon-nanotube-transistors-factory-0601

Duato, J., Mestre, J. I., Dolz, M. F., y Quintana-Ortí, E. S. (2023). GreenLightningAI: An Efficient AI System with Decoupled Structural and Quantitative Knowledge. arXiv. https://arxiv.org/abs/2312.09971

Green500, June 2024 List. (2024). Top500. https://top500.org/lists/green500/2024/06/

LeCun, Y., Cortes, C., y Burges, C. J. C. (2010). THE MNIST DATABASE of handwritten digits., https://www.kaggle.com/datasets/hojjatk/mnist-dataset

Nielsen, M. (2019). Neural Networks and Deep Learning. http://neuralnetworksanddeeplearning.com/

Nvidia GH200 Grace Hopper Superchip. (2024). Nvidia. https://www.nvidia.com/es-es/data-center/grace-hopper-superchip/

Stanford Vision Lab, Stanford University, Princeton University. (2021). ImageNet. https://www.image-net.org/

The Brussels Times Newsroom. (2024). ChatGPT consumes 25 times more energy than Google. The Brussels Times. https://www.brusselstimes.com/1042696/chatgpt-consumes-25-times-more-energy-than-google

CAPÍTULO IV

Grandes modelos de lenguaje: ¿de la predicción de palabras a la comprensión?*

Carlos Gómez-Rodríguez

Los grandes modelos de lenguaje, como el conocido ChatGPT, han supuesto una inesperada revolución en el ámbito de la inteligencia artificial. Por un lado, cuentan con multitud de aplicaciones prácticas y un enorme potencial todavía por explorar. Por otro lado, son también objeto de debate, tanto desde el punto de vista científico y filosófico como social: hay dudas sobre los mecanismos exactos de su funcionamiento y su capacidad real de comprensión del lenguaje, y sus aplicaciones plantean dilemas éticos. En este capítulo describimos cómo se ha llegado a esta tecnología y los fundamentos de su funcionamiento, permitiéndonos así comprender mejor sus capacidades y limitaciones e introducir algunos de los principales debates que rodean su desarrollo y uso.

Palabras clave: inteligencia artificial, procesamiento del lenguaje natural, grandes modelos de lenguaje.

* Parcialmente financiado por los proyectos SCANNER-UDC (PID2020-113230RBC21), financiado por MICIU/AEI/10.13039/501100011033; GAP (PID2022- 139308OA-I00) financiado por MICIU/AEI/10.13039/501100011033/ y FEDER, UE; LATCHING (PID2023-147129OB-C21) financiado por MICIU/AEI/10.13039/501100011033 y FEDER, UE; y TSI-100925-2023-1 financiado por el Ministerio para la Transformación Digital y de la Función Pública y "NextGenerationEU" PRTR; así como por la Xunta de Galicia (ED431C 2024/02), y el Centro de Investigación de Galicia "CITIC", financiado por la Xunta de Galicia a través del acuerdo de colaboración entre la Consellería de Cultura, Educación, Formación Profesional e Universidades y las universidades gallegas para el refuerzo de los centros de investigación del Sistema Universitario de Galicia (CIGUS).

1. INTRODUCCIÓN

Los grandes modelos de lenguaje (a menudo conocidos por las siglas en inglés *LLMs*, de *Large Language Models*) han precipitado un cambio de paradigma en el Procesamiento del Lenguaje Natural (PLN), la rama de la inteligencia artificial que busca desarrollar programas que puedan comprender y generar lenguaje humano. La clave del éxito de estos modelos es que, aprendiendo directamente a partir de texto simple, son capaces de responder adecuadamente a todo tipo de consultas, siempre que éstas sean expresables y respondibles mediante texto. Esto incluye la realización de tareas como traducción (Peng *et al.,* 2023), corrección gramatical (Fang *et al.,* 2023), resumen de textos (Pu *et al.,* 2023), respuesta a preguntas factuales (Brown *et al.,* 2020), o incluso escritura creativa (Gómez-Rodríguez y Williams, 2024), entre otras muchas. Mientras que en paradigmas anteriores (Manning y Schütze, 1999; Manning *et al.,* 2014) cada una de esas tareas requería diseñar, entrenar y ajustar un sistema específico para ella, incluyendo costosos procesos de recogida y anotación de datos especializados; los *LLMs* permiten llevarlas todas a cabo bajo el paraguas de un mismo sistema, sin ajuste específico, y con una interfaz natural que ha llevado a millones de usuarios a probar esta tecnología e integrarla en sus vidas en tiempo récord.

No obstante, para hacer un buen uso de estos modelos, es importante tener en cuenta sus limitaciones, siendo algunas de las principales las siguientes:

- Los mejores *LLMs* son sistemas computacionalmente muy costosos, lo cual está propiciando un oligopolio de facto por parte de grandes empresas tecnológicas (Bommasani *et al.,* 2021), enfrentado por modelos más pequeños (incluyendo algunos de código abierto) que por el momento no han logrado cerrar la brecha con los modelos grandes.
- Las respuestas pueden estar sesgadas o ser directamente incorrectas. Se ha llamado "alucinación" al fenómeno por el cual los *LLMs* producen respuestas que pueden estar sintácticamente bien redactadas, e incluso tener coherencia interna, pero no tienen sentido, no responden a la consulta o se alejan de la realidad.
- Se trata de sistemas opacos, donde no contamos con mecanismos fiables para determinar por qué están proporcionando una respuesta y no otra. Esto agrava el punto anterior, dado que el modelo no nos proporciona información para verificar su respuesta, haciendo desaconsejable fiarse de la salida de un *LLM* sin verificación externa.

Además, debemos ser conscientes de que, si bien algunas de estas limitaciones pueden ser fruto de la inmadurez de una tecnología incipiente y resolverse a lo largo de los próximos años, otras podrían ser intrínsecas e inevitables (Xu *et al.,* 2024; Banerjee *et al.,* 2024).

Asimismo, e incluso cuando funcionan correctamente, el uso de los *LLMs* plantea diversos retos sociales y cuestiones éticas derivadas de su uso malintencionado (por ejemplo, para generar información falsa, manipular a personas, o cometer fraude en trabajos y exámenes) o de su sobreuso (por ejemplo, sustituyendo decisiones sensibles que debería tomar una persona).

Para comprender mejor tanto las oportunidades creadas por esta tecnología como los desafíos que plantea, en este capítulo explicaremos las bases del funcionamiento de los *LLMs*, las claves técnicas de su éxito respecto a anteriores sistemas de PLN, y algunas de las más importantes preguntas abiertas y desafíos a los que se enfrenta la investigación en este campo.

Para ello, el resto del capítulo se estructura del siguiente modo: la sección 2 explica por qué los *LLMs* son tan revolucionarios, contextualizándolos mediante un resumen de los diferentes paradigmas que se han sucedido a lo largo de la historia del Procesamiento del Lenguaje Natural, para posicionar el avance que suponen. La sección 3 explica cómo funcionan los *LLMs*; y la sección 4 se basa en dicho funcionamiento para explicar lo que sabemos sobre las capacidades y limitaciones funcionales de los *LLMs*. Por último, la sección 5 resume las conclusiones del capítulo.

2. ¿POR QUÉ SON REVOLUCIONARIOS?

Como hemos mencionado, los grandes modelos de lenguaje (*LLMs*) han provocado un cambio de paradigma en el PLN, situando además este campo de investigación en el candelero. Pero ¿por qué son tan revolucionarios? ¿Qué es lo que los diferencia de anteriores sistemas que trabajaban en lenguaje humano? Para saberlo, conviene hacer una breve perspectiva histórica de los distintos paradigmas que se han sucedido en PLN.

El PLN tiene sus orígenes en los años 50 del siglo XX. Uno de los primeros hitos fue la propuesta del test de Turing (1950), que sugería evaluar la inteligencia de las máquinas a través de su capacidad de mantener una conversación de forma indistinguible de un ser humano. Al mismo tiempo, la Guerra Fría despierta un creciente interés por la posibilidad de traducir textos automáticamente. El experimento de Georgetown-IBM de 1954 (Hutchins, 1954) mostró al mundo un sistema capaz de traducir algunas oraciones entre inglés y ruso. Aunque muy rudimentario, los titulares grandilocuentes como "Electronic brain translates Russian"[1] difundieron entre el público general la idea de que la traducción automática era posible. Enseguida se sucedieron las predicciones optimistas, según las cuales el problema estaría resuelto definitivamente en pocos años, y la lingüística computacional (que se puede definir como la disciplina científica cuya aplicación práctica es el PLN) empezó a recibir financiación, con el consiguiente logro de avances en la tecnología. Desde estos orígenes hasta la actualidad, se podría dividir la evolución tecnológica del PLN en tres etapas, según la manera en que se ha venido "enseñando" a las máquinas a trabajar con lenguaje humano.

La primera de estas etapas abarcaría hasta finales de los años 80. En ella, los sistemas de PLN se basaban casi exclusivamente en reglas escritas a mano por expertos. Noam Chomsky (Chomsky, 1957) describe cómo sistematizar la gramática de un idioma como el inglés mediante reglas sintácticas, que permiten generar e interpretar oraciones. Basándose en este principio, se desarrollan analizadores sintácticos capaces de descomponer las oracio-

[1] https://aclanthology.org/www.mt-archive.info/ChemEngNews-1954.pdf

nes en sus componentes gramaticales, facilitando extraer información estructurada a partir de textos. Del mismo modo, se construyen también sistemas basados en reglas para tareas como traducir textos, responder preguntas o incluso mantener una conversación, como el conocido sistema Eliza (Weizenbaum, 1966). El problema de estos enfoques es que tienen serios problemas de escalabilidad: resulta muy costoso escribir reglas manualmente para cada problema, idioma y dominio de aplicación; y es muy difícil capturar todas las variaciones y excepciones del lenguaje humano, que es cambiante y ambiguo.

Hacia finales de los 80 comienzan a aparecer, y pronto se vuelven dominantes, los sistemas basados en aprendizaje estadístico. Estos no necesitan reglas, sino que se enseña el idioma a las máquinas mediante conjuntos de ejemplos junto con algoritmos de aprendizaje automático que aprenden de ellos. Por ejemplo, para entrenar un traductor automático, se utilizaría un corpus de oraciones en el idioma origen con su traducción al idioma destino (Brown *et al.,* 1990). Este tipo de enfoques son más escalables y baratos que los basados en reglas, dado que no hace falta involucrar a expertos en el diseño del sistema mientras se cuente con datos de buena calidad (como ejemplifica la famosa frase de Fred Jelinek: "Every time I fire a linguist, the performance of the system goes up" [Hirschberg, 1998]).

A lo largo de esta etapa, la calidad de los algoritmos de aprendizaje automático va mejorando según avanzan las investigaciones, especialmente con la irrupción del llamado aprendizaje profundo (*deep learning*), una evolución que devolvió a primera línea las olvidadas redes neuronales, en la primera mitad de la década de 2010. Los sistemas con aprendizaje automático basado en *deep learning* representan las palabras en un espacio continuo de vectores densos (Mikolov *et al.,* 2013), en lugar de como entidades discretas, que se usan como entrada a redes neuronales que aprenden diferentes tareas (Cho *et al.,* 2014). Sin embargo, las mejoras cuantitativas en la precisión de los sistemas logradas por estas tecnologías no cambian la limitación fundamental de esta etapa: se trata de sistemas de propósito específico, es decir, que llevan a cabo una tarea concreta. Entrenando una red neuronal con oraciones en castellano y sus traducciones al inglés, podemos lograr un sistema de traducción automática efectivo para traducir textos de castellano a inglés. Pero será inútil para otros idiomas que no estén en el conjunto de entrenamiento, y más aún para otras tareas (como responder preguntas o resumir textos). Para ello, necesitaríamos entrenar un sistema distinto con un conjunto de datos ajustado a la tarea deseada. Así, si deseamos resumir textos en castellano, tendremos que crear otro sistema diferente, desde cero, entrenándolo con un corpus de textos en castellano con sus resúmenes, y así con cada nueva tarea que se quiera acometer.

Y es precisamente esta limitación la que viene a resolver la tecnología que motiva este capítulo, los grandes modelos de lenguaje, que han inaugurado una aún incipiente tercera etapa del desarrollo del PLN. Desde un punto de vista técnico, los grandes modelos de lenguaje provienen del escalado de los modelos neuronales de la etapa anterior (redes neuronales más grandes gracias a los avances en *hardware*, datos de entrenamiento más grandes, y arquitecturas neuronales mejores, como los Transformers [Vaswani *et al.,* 2017]). Pero desde un punto de vista más amplio, estos modelos, aun proviniendo de una evolución y no una ruptura con los anteriores, suponen un cambio de paradigma, perfectamente resumido por

el título del artículo que introdujo uno de los modelos más transformativos, GPT-2 (Radford *et al.,* 2018), y que se podría perfectamente considerar el hito que inaugura la nueva etapa: *Language Models are Unsupervised Multitask Learners*. En otras palabras, los grandes modelos de lenguaje ya no necesitan datos especializados en una tarea (como oraciones junto con su traducción a otro idioma, o textos con su versión resumida) sino que aprenden a manejar el idioma a partir de cantidades gigantescas de texto, descargado tal cual de Internet u otras fuentes, sin necesidad de que algún humano los adapte a la tarea específica a realizar (*unsupervised*)... pues de hecho, no están restringidos a una tarea concreta sino que el mismo sistema puede llevar a cabo una variedad de tareas que se le pidan (*multitask*). Esto supone una auténtica revolución en la práctica, pues no solamente se obvia la necesidad de datos especializados para cada tarea, idioma y dominio de aplicación al que se quiera aplicar PLN, sino que se cuenta con sistemas de propósito general (como ChatGPT) que interactúan con el usuario final en su idioma, que puede hacer todo tipo de solicitudes sin necesidad de aprender a utilizar un *software* especializado. Por lo tanto, los grandes modelos de lenguaje hacen más versátil y universalmente accesible el PLN, a la vez que nos acercan a la superación del test de Turing.

Este cambio de paradigma, dada su reciente aparición y la vertiginosa velocidad con la que se van sucediendo los nuevos modelos y avances, suscita muchas preguntas, tanto entre los especialistas en el tema como entre el público en general. Por ejemplo, una cuestión en debate es si estos grandes modelos de lenguaje son realmente capaces de entender el lenguaje, en mayor o menor grado, o únicamente simulan respuestas coherentes sin entender en absoluto. Otras cuestiones de actualidad son si estos modelos son o pueden llegar a ser creativos, si podrían llegar a adquirir consciencia, y toda la serie de debates éticos y sociales que se plantean en torno a ellos (su fiabilidad, posibles malos usos para generar desinformación, o hasta qué punto podrían o deberían sustituir a los humanos en ciertos roles, por ejemplo).

Si bien muchas de estas cuestiones no tienen de momento respuestas definitivas que susciten consenso, para entender el debate y evitar posturas simplistas es necesario comprender algunos fundamentos de cómo funcionan los *LLMs*. Esto es lo que describimos en la siguiente sección, evitando deliberadamente el típico enfoque técnico que disecciona sus componentes (pues no es realmente necesario para este fin, haría la explicación menos accesible, y es posible que sea contingente – nada garantiza que, dentro de unos años, las arquitecturas neuronales que hacen funcionar los *LLMs* no sean radicalmente distintas de las actuales [Gu y Dao, 2024]) para centrarnos en lo esencial: *qué hacen* los *LLMs*, y qué es lo que hace que tengan las características que hemos descrito más arriba.

3. ¿CÓMO FUNCIONAN?

La esencia de un *LLM* es un concepto muy sencillo: son modelos que, a partir de cantidades masivas de textos que se usan como datos de entrenamiento, *predicen una continuación plausible de un texto*. Por ejemplo, si a un buen modelo de lenguaje le pasamos el texto "se me

ha roto el", será capaz de continuar con una palabra coherente con el contexto (como puede ser "móvil", "jarrón" o incluso "corazón"). Para poder hacer eso, los *LLMs* se basan en los textos que han visto durante su entrenamiento.

3.1. Generando texto con cadenas de Markov

Para comprenderlo más en detalle, es conveniente remontarnos a una época muy anterior a los orígenes del PLN. En 1906, el matemático ruso Andrey Markov descubre un modelo para describir procesos estocásticos, conocido como cadena de Markov. Una cadena de Markov de orden k es un modelo que va atravesando una serie de estados, de tal modo que la probabilidad de cada nuevo estado depende de los k estados anteriores. Por ejemplo, una cadena de Markov podría asignar probabilidades al tiempo que hará mañana en función del tiempo de ayer y hoy, en función de una tabla como la tabla 1.

Tabla 1.

Cadena de Markov de orden 2 para modelar el tiempo atmosférico

Ayer	Hoy	Probabilidad (mañana soleado)	Probabilidad (mañana nublado)	Probabilidad (mañana lluvioso)
Soleado	Soleado	0.7	0.2	0.1
Soleado	Nublado	0.4	0.4	0.2
Soleado	Lluvioso	0.4	0.3	0.3
Nublado	Soleado	0.4	0.4	0.2
Nublado	Nublado	0.2	0.5	0.3
Nublado	Lluvioso	0.2	0.4	0.4
Lluvioso	Soleado	0.3	0.5	0.2
Lluvioso	Nublado	0.2	0.5	0.3
Lluvioso	Lluvioso	0.1	0.3	0.6

La tabla proporciona una probabilidad de que mañana haga un día soleado, nublado o lluvioso en función del tiempo que ha hecho ayer y hoy. Por ejemplo, si ayer estuvo un día nublado y hoy está lluvioso, la tabla (sexta fila) dice que hay una probabilidad 0.2 de que mañana sea un día soleado, 0.4 de nublado y 0.4 de lluvioso. Del mismo modo, podría-

mos consultar las probabilidades para cualquier otra combinacion de días. En este ejemplo, siempre necesitamos dos días (ayer y hoy) para estimar las probabilidades del tiempo del día siguiente (mañana), porque se trata de una cadena de Markov de orden 2. El orden podría ser también superior o inferior. En el caso de este ejemplo, las probabilidades de la tabla son inventadas, pero en un caso real, nos basaríamos en datos históricos: por ejemplo, si en el pasado después de un día nublado y otro lluvioso vino un día soleado el 20 % de las veces, pondríamos un 0.2 en la correspondiente celda de la tabla. De este modo, las probabilidades se corresponderán a lo que cabe esperar de acuerdo con los datos históricos.

A partir de este modelo, podemos generar una secuencia estocástica, abarcando no solamente el tiempo que va a hacer mañana, sino también los días siguientes. Por ejemplo, supongamos que ayer hizo un día nublado y hoy está lluvioso. Podemos generar un tiempo plausible para mañana haciendo un sorteo aleatorio según las probabilidades de la tabla (es decir, donde haya una probabilidad 0.2 de que salga soleado, 0.4 de nublado y 0.4 de lluvioso). Supongamos que el resultado de este sorteo es un día lluvioso. Entonces, podemos repetir el proceso para pasado mañana: los dos días anteriores serían lluvioso y lluvioso, así que las probabilidades serían las de la última fila de la tabla (0.1 de día soleado, 0.3 de nublado y 0.6 de lluvioso). Este proceso podemos repetirlo indefinidamente para generar todos los días que queramos. A medida que nos alejemos en el futuro, será más difícil que la predicción del modelo se corresponda con lo que sucederá en realidad, pero al menos será una continuación plausible del tiempo registrado los dos últimos días.

El mismo concepto se puede aplicar para generar un texto en lenguaje humano, si consideramos que los estados son palabras[2] en lugar de condiciones meteorológicas. Dado un corpus de textos, las tablas de probabilidades se pueden estimar fácilmente a partir del corpus, contando las apariciones de cada posible *k*-grama (secuencia de *k* palabras) y las palabras que lo siguen.

Por ejemplo, si buscamos el bigrama (2-grama) "unidos de" en el corpus Bruno, un pequeño corpus de un millón de palabras en español con una interfaz web de libre acceso y fácil de usar para el lector que quiera experimentar (Spanish Concordancer - Bruno Spanish) Corpus; y nos fijamos en qué palabra viene después, obtendremos la tabla 2, donde vemos que este bigrama aparece casi siempre seguido de "Norteamérica" o "América" (asociado, como se puede deducir, a referencias a los EE. UU.) aunque también pueden aparecer otras palabras, como "trabajar" (en la oración "El compromiso de Estados Unidos de trabajar con los países en vías de desarrollo..."). La poca variedad de palabras se debe al pequeño tamaño del corpus: si tuviésemos más datos, sin duda podríamos observar más palabras que pueden aparecer después de "unidos de", y nuestras estimaciones de probabilidad serían también más precisas.

[2] En realidad, los grandes modelos de lenguaje no trabajan con palabras, sino que dividen el texto en subpalabras que típicamente se obtienen mediante un algoritmo de codificación de pares de bits (Gage, 1994). Esto se hace de este modo por diversos motivos técnicos, incluyendo una mayor eficiencia y flexibilidad para trabajar con palabras que el modelo no ha visto antes. Por simplicidad y claridad de exposición, en nuestra explicación ignoraremos este aspecto y supondremos que se trabaja con palabras, lo cual también sería una implementación posible (aunque no sea la más eficaz y eficiente) y no cambia la comprensión de los conceptos fundamentales.

Tabla 2.

Frecuencias y probabilidades estimadas que se obtienen para modelar la palabra que viene después del bigrama "unidos de", a partir de un corpus de un millón de palabras en español

w_{i-2}	w_{i-1}	w_i	Frecuencia en corpus	Probabilidad estimada
unidos	de	Norteamérica	7	0.5385
unidos	de	América	5	0.3846
unidos	de	trabajar	1	0.0769

Nota: La notación w_{i-2}, w_{i-1} y w_i hace referencia a las palabras en las posiciones i-2, i-1 e i de un texto.

Para generar texto con nuestra cadena de Markov de orden 2, recopilaríamos datos como los de la tabla 2 para todos los bigramas del corpus (es decir, tendríamos una tabla exhaustiva con una fila para cada posible bigrama y su continuación, al modo de la tabla 1, o si se prefiere, tablas individuales como la tabla 2 para cada combinación de bigramas). A partir de esto, el proceso de generación es sencillo:

- Empezar con un k-grama cualquiera del corpus. Este *k*-grama constituirá el principio del texto generado.
- Repetir hasta que se desee:

  (a) Para las *k* últimas palabras generadas, mirar en las tablas las posibles palabras siguientes y la probabilidad de que aparezcan a continuación de ese *k*-grama.

  (b) Hacer un sorteo para elegir una de estas palabras siguientes, utilizando dichas probabilidades (es decir, cada una de las posibles palabras siguientes tendrá la probabilidad de salir que indique la tabla).

  (c) La palabra seleccionada se añade al final del texto generado.

Así, si por ejemplo el bigrama inicial fuese "las mujeres", nuestro algoritmo de generación consultaría las palabras que aparecen a continuación de ese bigrama en el conjunto de entrenamiento, obteniendo los resultados que se muestran en el bloque superior de la tabla 3. A continuación, escogería aleatoriamente una palabra de entre las posibles continuaciones, con probabilidad proporcional a las estimaciones de la tabla. En el caso de que la palabra así escogida fuese "son", el texto generado sería ahora "las mujeres son" y, en la siguiente iteración del algoritmo, se consultarían las palabras que aparecen a continuación de "mujeres son" en el corpus de entrenamiento, obteniendo los datos del bloque inferior de la tabla 3. De ahí, se esco-

gería de nuevo una palabra al azar (por ejemplo, "muy"), quedando el texto como "las mujeres son muy". Este proceso se puede seguir iterando para generar un texto tan largo como se desee.

Tabla 3.

Frecuencias y probabilidades estimadas que se obtienen para modelar la palabra que viene después de los bigramas "las mujeres" y "mujeres son", a partir de un corpus de un millón de palabras en español.

w_{i-2}	w_{i-1}	w_i	Frecuencia en corpus	Probabilidad estimada
las	mujeres	que	7	0.0414
las	mujeres	son	6	0.0355
las	mujeres	en	5	0.0296
las	mujeres	no	5	0.0296
las	mujeres	se	5	0.0296
las	mujeres	...	...	...
mujeres	son	como	1	0.1667
mujeres	son	mayoría	1	0.1667
mujeres	son	tales	1	0.1667
mujeres	son	un	1	0.1667
mujeres	son	muy	1	0.1667
mujeres	son	fundadas	1	0.1667

Mediante este proceso estocástico, podemos generar texto que parece lenguaje. La tabla 4 muestra el resultado de aplicar este proceso con la Biblia como conjunto de entrenamiento[3], y el efecto de diferentes valores de k. En todos los textos así generados se puede apreciar el léxico y temática bíblica, dado que es la fuente de la que el modelo obtiene las palabras; pero las características sintácticas y semánticas del texto son muy diferentes según el valor de k. Para k=1, el texto carece de coherencia y tiene bastantes errores sintácticos (como "llegué a los pecados que todos"), además de no tener sentido lógico. Al aumentar el valor de k, el texto va cobrando coherencia, los errores sintácticos van desapareciendo (véase k=3, donde la sintaxis está cerca de ser correcta), y se le puede dar al menos un sentido a fragmentos del texto, aunque en su conjunto carece de sentido lógico por no tener consistencia suficiente. El aumento progresivo de coherencia y la disminución de los errores sintácticos a medida que aumenta k se debe a que la cadena de Markov garantiza que, por lo menos, cada (k+1)-grama del texto resulte plausible (dado que, por construcción, sabemos que cada (k+1)-grama del texto aparece alguna vez en el conjunto de entrenamiento).

[3] Santa Biblia: antigua versión de Casiodoro de Reina (1569), revisada por Cipriano de Valera (1602) : otras revisiones, 1862, 1909 y 1960. Revisión de 1960. Disponible en https://raw.githubusercontent.com/kblok/RNN-bible-generator/refs/heads/master/biblia.txt

Tabla 4.

Generación de textos bíblicos mediante cadenas de Markov para distintos valores de *k*

Valor de *k*	Texto generado
1	porque a Jehová había prevaricado con los animales del templo de Isacar, el borde de ellos le deja dormir conmigo, llegué a los pecados que todos, y recogieron doble su tiempo, de Jehová, tu vida, Y su cerrojo, se os anunciamos que Dios había agua salada con los sirvientes del Jordán, como un ángel de Parvaim.
2	Vino a mí palabra de Jehová llenó la tierra. Pero a vosotros con nuestros sacrificios y alabanzas y acción de gracias y la verdad, la cual dio a luz hijo o hija, hermano, o su hija fue sanada desde aquella hora. Y edificó allí altar a Jehová; la grosura del sacrificio de paz os santifique por completo.
3	Yo me acordaré de mi pacto con ellos, dijo Jehová: El Espíritu mío que está sobre tus lomos, y descalza las sandalias de tus pies. Y él puso su diestra sobre mí, diciéndome: No temas; yo soy el que los hirió, sino que se levantó en el campo de Joab está junto al mar de Galilea, y les enseñaba en los días de adversidad.
10	Y las aguas del mar faltarán, y el río se agotará y secará. Y se alejarán los ríos, se agotarán y secarán las corrientes de los fosos; la caña y el carrizo serán cortados. La pradera de junto al río, de junto a la ribera del río, y toda sementera del río, se secarán, se perderán, y no serán más.

Podría pensarse entonces que, para alcanzar un texto totalmente coherente, bastaría con aumentar *k* hasta donde sea necesario. Sin embargo, si hacemos eso, nos encontraremos con otro problema: el agotamiento del espacio muestral. Si observamos el texto generado para *k*=10 en la tabla 4 podremos ver que, efectivamente, es muy coherente. El problema es que se trata de una copia literal de un fragmento del conjunto de entrenamiento, dado que éste no es lo suficientemente grande como para que encontremos en él varias muestras de un 10-grama. Por lo tanto, dado un 10-grama (como puede ser "Y las aguas del mar faltarán, y el río se"), nuestra tabla de posibles continuaciones solamente tendrá una opción ("agotará"), con probabilidad 1, y lo que estará haciendo el modelo estocástico es reproducir el texto bíblico en lugar de generar texto nuevo.

La generación de textos con cadenas de Markov, pues, está muy lejos de la apariencia de inteligencia que se atribuye a los textos generados por *LLMs*, y es de dudosa aplicabilidad práctica por las limitaciones que tiene: con *k* pequeño, los textos no tienen coherencia, y con *k* más grande, enseguida nos encontramos con la barrera de la escasez de datos. Se podría pensar en mitigar esto último con conjuntos de entrenamiento más grandes; pero esto por sí solo tiene un alcance muy limitado: dado que la probabilidad de encontrar un determinado *k*-grama en un texto disminuye de manera exponencial respecto a *k*, ni aunque juntásemos todos los textos escritos a lo largo de la historia de la Humanidad podríamos tener estimaciones de probabilidad aceptables para *k*=10. Basta pensar que, si elegimos una secuencia de *k* palabras consecutivas de este texto o cualquier otro que hayamos escrito, lo más probable

es que no haya sido escrita nunca antes (salvo en casos concretos de texto formulaico, como frases hechas o textos legales). Por lo tanto, por muy grande que sea nuestro conjunto de entrenamiento, es casi seguro que un modelo con k=10 se va a limitar a copiar literalmente.

Sin embargo, estas limitaciones de las cadenas de Markov se pueden atajar si nos salimos de las reglas estrictas que nos impone el modelo, haciéndolo más flexible. Desde el descubrimiento de las cadenas de Markov fueron apareciendo algunas mejoras incrementales en este sentido, como las técnicas de suavizado (Chen y Goodman, 1996), interpolación (Jelinek, 1980) o *back-off* (Katz, 1987). Sin embargo, el avance que realmente permitiría desbloquear modelos generativos de lenguaje con valores grandes de k fueron los *modelos de lenguaje neuronales*.

3.2. De las cadenas de Markov a los modelos neuronales

Los modelos de lenguaje neuronales son, como las cadenas de Markov, modelos que generan una continuación plausible de un texto a partir de lo que han observado en el conjunto de entrenamiento. Sin embargo, en lugar de basarse puramente en cálculos de probabilidades condicionales, la generación se produce mediante redes neuronales. Esto hace que estos modelos puedan generar textos de manera más flexible, soportando así longitudes de contexto (k) más largas. Si bien explicar en detalle el funcionamiento de estos modelos escapa a los objetivos de este capítulo, sí cabe destacar que son principalmente tres los avances que han posibilitado lograr esta flexibilidad de la que las cadenas de Markov carecían: las representaciones densas del lenguaje, los avances técnicos en redes neuronales, y la disponibilidad de enormes conjuntos de datos obtenidos de Internet. A continuación se describe brevemente cada uno de estos avances y su relevancia.

Representaciones densas del lenguaje

Las representaciones densas del lenguaje, conocidas en inglés como *word embeddings*, son vectores numéricos que capturan el significado de las palabras al ubicarlas en un espacio vectorial donde palabras con significados similares están cerca unas de otras. Típicamente, el vector que representa cada palabra está formado por unos pocos cientos de componentes en coma flotante (aproximación informática de los números reales). La popularización de estas representaciones se produjo, sobre todo, a raíz del modelo word2vec de Mikolov *et al.* (2013), que obtiene vectores para las palabras a partir de una sencilla red neuronal. Desde entonces, han ido apareciendo diferentes alternativas (Pennington *et al.,* 2014), incluyendo representaciones contextualizadas, donde el vector de cada palabra varía según el contexto en que aparezca (Devlin *et al.,* 2019).

El uso de representaciones densas del lenguaje tiene varias ventajas sobre considerar cada palabra una entidad discreta. Una de ellas es que las redes neuronales se adaptan mejor a

trabajar con vectores densos de números reales que con entidades discretas. Pero la principal ventaja es, sobre todo, la flexibilidad que añaden al modelar de forma natural la similitud entre palabras. En las cadenas de Markov vistas en el apartado 3.1, así como en otros modelos estadísticos tradicionales usados en PLN, cada palabra se representa como una entidad atómica sin relación aparente con las demás: un modelo de este tipo verá las palabras "abanico", "avión" y "aeroplano" como cadenas de texto diferentes que se traducen interiormente a algún tipo de identificador numérico; pero el modelo no tendrá constancia de que las dos últimas se parecen mucho más entre sí que a la primera. Así, si en una cadena de Markov queremos continuar el texto "perdió el control del aeroplano" y ese *k*-grama exacto no aparece en el conjunto de entrenamiento, pero sí aparece "perdió el control del avión", no podremos aprovechar esta similitud para generar nuestra continuación del texto. En cambio, un modelo neuronal sí podrá hacerlo: en lugar de buscar coincidencias exactas de palabras en una tabla, el modelo neuronal trabaja en un espacio continuo de representaciones vectoriales, así que puede aprovecharse de fragmentos de texto similares, pero no iguales, que haya visto en el entrenamiento. Esto le proporciona flexibilidad adicional frente a la rigidez de una cadena de Markov, y la capacidad de sacar mucho más partido a los datos.

Avances técnicos en redes neuronales

Las redes neuronales son modelos computacionales inicialmente inspirados en la estructura y funcionamiento del cerebro humano, diseñados para aprender la relación entre entradas y salidas deseadas para problemas complejos. Están formadas por unidades de procesamiento llamadas "neuronas", organizadas en capas que se conectan entre sí. En una red neuronal, las neuronas reciben entradas, realizan cálculos y transmiten los resultados a otras neuronas en la siguiente capa a través de conexiones que las unen. Mediante un proceso de entrenamiento, estas conexiones se ajustan usando algoritmos de optimización, permitiendo que la red aprenda patrones a partir de datos y mejore en su tarea específica, como puede ser el procesamiento de textos.

Si bien las redes neuronales son conocidas desde mediados del siglo XX, a finales de siglo era una tecnología bastante olvidada pues, con lo que se sabía hasta entonces, no solían ser la mejor opción en la práctica para la mayoría de problemas, siendo claramente superadas por otras alternativas como las máquinas de vectores soporte. De hecho, como señalan LeCun *et al.* (2015), las redes neuronales fueron "en gran medida abandonadas por la comunidad de aprendizaje automático" porque "se pensaba ampliamente que [la forma en la que aprendían a partir de los datos] era inviable". Su resurgir es un ejemplo de cómo la inversión en investigación básica, incluso sin aplicaciones claras a la vista, es clave para el avance científico: durante la década de 2010, se logran importantes mejoras en las redes neuronales con una serie de descubrimientos que posibilitan el llamado aprendizaje profundo (Goodfellow *et al.,* 2016), un enfoque que transforma el campo de la inteligencia artificial. Gracias a avances en algoritmos, como el uso de capas de redes neuronales más profundas y la implementación de técnicas de regularización y optimización, así como el aumento de la capacidad de cómputo

facilitado por el uso de *GPUs*, estas redes comienzan a alcanzar un rendimiento notable en tareas complejas, incluyendo en procesamiento del lenguaje natural (Goldberg y Hirst, 2017).

Si bien inicialmente las redes neuronales usadas en PLN son arquitecturas ya conocidas de otros problemas que se adaptan para procesar lenguaje (Goldberg y Hirst, 2017), el avance que supondría el pistoletazo de salida para los modelos de lenguaje neuronales que más tarde evolucionarían a grandes modelos de lenguaje es el desarrollo de una arquitectura neuronal específicamente pensada para el procesado de textos: los *Transformers* (Vaswani *et al.,* 2017). Esta arquitectura cuenta con un mecanismo de "autoatención" (*self-attention*) que permite relacionar directamente cada elemento de una secuencia de datos (como una palabra en una oración) con todos los demás, capturando así las dependencias a larga distancia que caracterizan al lenguaje humano. Los modelos basados en *Transformers* enseguida empezaron a mostrar una especial eficacia en todo tipo de tareas de PLN, superando a las arquitecturas neuronales anteriores (Devlin *et al.,* 2019), y están en el núcleo de todos los grandes modelos de lenguaje más conocidos; aunque también estén empezando a explorarse otras arquitecturas alternativas que podrían sustituirlos (Gu *et al.,* 2024).

Grandes volúmenes de datos

El tercer avance clave para lograr modelos más eficaces para continuar de modo plausible un texto es el entrenamiento con cantidades cada vez mayores de datos. Si bien comentamos en el apartado 3.1 que esto por sí solo no sería suficiente para romper las limitaciones de las cadenas de Markov, aunque tuviésemos todo el texto que jamás se ha escrito; este factor sí que ha resultado clave en combinación con las representaciones vectoriales del lenguaje y las mejoras en las arquitecturas neuronales, que permiten precisamente explotar mejor los datos. Concretamente, muchos grandes modelos de lenguaje actuales utilizan conjuntos de entrenamiento de terabytes (Liu *et al.,* 2024), que contienen billones españoles (o trillones americanos) de palabras. Por ejemplo, LLaMa 3, un modelo de lenguaje de Meta, se entrenó sobre 15 billones españoles de "tokens" (fragmentos de palabras) (Dubey *et al.,* 2024), lo cual puede suponer entre 7 y 10 billones españoles de palabras. Para hacerse una idea de lo enorme que es esta cifra, se ha estimado que todos los libros existentes en el mundo en la actualidad podrían sumar unos 16 billones españoles de palabras[4].

3.3. De la predicción de palabras a la realización de tareas

Con los tres ingredientes explicados en la sección anterior, podemos conseguir modelos neuronales de lenguaje que hacen lo mismo que las cadenas de Markov: generar una continuación plausible de un texto, a partir de lo que han visto en su conjunto de entrenamiento, de acuerdo con las probabilidades que se deducen de los datos (continuaciones más típi-

[4] https://www.educatingsilicon.com/2024/05/09/how-much-llm-training-data-is-there-in-the-limit/#all-books

cas o comunes serán más probables, mientras que las más atípicas serán más improbables). Sin embargo, podemos hacerlo sin las limitaciones intrínsecas de las cadenas de Markov: los modelos de lenguaje neuronales nos permiten usar tamaños de contexto de cientos de palabras sin que aparezca el problema de la escasez de datos, mejorando en gran medida la coherencia de los textos generados.

Así, en la tabla 5 se muestran tres textos generados con una versión en español de GPT-2[5], un *LLM* pequeño para los estándares actuales, entrenado sobre 11.5 GB de datos. Los textos fueron generados proporcionando la entrada *La torre de Hércules es,* y permitiendo que el modelo predijese palabras a partir de ahí. Como se puede apreciar, si nuestro objetivo es generar textos en correcto español, lo hemos cumplido con creces: los textos generados son sintácticamente correctos, e incluso semánticamente coherentes. Ahora bien, el contenido de los textos no se ajusta a la realidad: en cada caso se genera una continuación aleatoria que sitúa la torre de Hércules en un lugar diferente, y nunca en su ubicación real, A Coruña, además de proporcionar otros datos que tampoco son veraces. Por supuesto, esto no debería sorprendernos: como hemos reiterado varias veces a lo largo de este capítulo, estos modelos están entrenados simple y exclusivamente para *generar una continuación plausible del texto.* En ningún momento de su proceso de entrenamiento se hace nada para procurar que generen textos factualmente ciertos.

Tabla 5.

Generación de textos con GPT-2 a partir de la cadena "La torre de Hércules es"

La Torre de Hércules es una de las construcciones más antiguas de la ciudad de Alicante. Se encuentra en la Plaza de la Constitución, en el centro de la ciudad. Fue construida en el siglo XVI y es una de las más antiguas de la ciudad...
La Torre de Hércules es una torre de piedra, situada en la ciudad de Palma de Mallorca, a una altura de 86 metros. La torre de Hércules data del siglo XIV. Se construyó siguiendo un modelo inspirado en los edificios de Córcega...
La Torre de Hércules es la construcción del siglo XI más antigua de la que se conserva en Cataluña, y su estructura original ha sido reconstruida en múltiples ocasiones, pero únicamente la tercera y última ha permanecido en pie...

Lo realmente sorprendente, y que supuso un giro inesperado para la comunidad de investigación en PLN (Wei *et al.,*2022), es que simplemente escalando el modelo (entrenándolo con más datos, y haciendo la propia red neuronal más grande, con más neuronas y conexiones), sin ninguna mejora cualitativa en especial, se consigue que empiece a decir la verdad. La tabla 6 muestra varias generaciones de texto a partir del mismo principio que la tabla 5; pero esta vez provienen de BLOOM (Workshop *et al.,* 2023), un modelo más grande y entrenado con unos 200.000 millones de palabras, pero que por lo demás no es esencialmente distinto de GPT-2. Como se puede ver en la tabla, el resultado son textos que

[5] https://huggingface.co/DeepESP/gpt2-spanish

ya no solo son sintácticamente correctos y semánticamente coherentes, sino que además dan en su mayoría datos ciertos (aunque no con fiabilidad total: por ejemplo, no es cierto que la Torre de Hércules esté en un islote).

Tabla 6.

Generación de textos con BLOOM a partir de la cadena "La torre de Hércules es"

La Torre de Hércules es el faro más antiguo del mundo en funcionamiento y el único del mundo romano que sigue dando servicio. La torre se encuentra situada en la península de la ciudad de A Coruña y está construida sobre una base rocosa de 40 metros de altura...
La Torre de Hércules es, sin lugar a duda, uno de los símbolos de la ciudad de La Coruña y uno de los faros en funcionamiento más antiguos del mundo. La historia del faro empieza en el siglo I d.C. cuando los romanos lo construyeron sobre las ruinas de un faro etrusco...
La Torre de Hércules es el faro más antiguo del mundo todavía en funcionamiento, se encuentra situado en la ciudad de A Coruña, Galicia, sobre un islote en la península de la Torre. El faro data del siglo I.…

Por otra parte, si utilizamos el mismo modelo para continuar una pregunta *(¿Qué es la Torre de Hércules?)*, obtendremos los textos de la tabla 7. Como vemos, el resultado es que, a veces, el modelo responde la pregunta, y otras veces no. Este comportamiento es lógico: dado un texto que contiene una pregunta, una continuación plausible del texto es responder dicha pregunta (a menudo tras una pregunta aparece su respuesta, y el sistema habrá visto este patrón en el conjunto de entrenamiento) pero también hay otras continuaciones plausibles que no implican responder (por ejemplo, en el texto de un examen pueden aparecer preguntas seguidas de más preguntas, y sin ninguna respuesta).

Tabla 7.

Generación de textos con BLOOM a partir de la cadena "¿Qué es la Torre de Hércules?"

¿Qué es la Torre de Hércules? La Torre de Hércules (A Coruña) es el faro en funcionamiento más antiguo del mundo. Símbolo de la ciudad...
¿Qué es la Torre de Hércules? La Torre de Hércules es uno de los monumentos más emblemáticos de España, es un faro construido por los romanos en el Siglo I...
¿Qué es la Torre de Hércules? ¿Cuánto mide? ¿Qué altura tiene la Torre de Hércules? ¿Dónde está? ¿En qué lugar se ubica? ¿Cuánto tiempo es necesario para subir? ...

Por lo tanto, mediante el escalado de los modelos de lenguaje neuronales, hemos obtenido un sistema que es capaz de proporcionar información veraz, aunque sin mucha fiabilidad, y que incluso responde preguntas, aunque solamente a veces. Más allá de la curiosidad de generar textos al azar, el modelo podría ser realmente útil, si lográramos dotarlo de consis-

tencia: por ejemplo, nos gustaría que ante una pregunta, siempre intentarse responder, como en los dos primeros ejemplos de la tabla 7, en lugar de generar otras preguntas como en el último. Dicho de otro modo, no solo queremos que el modelo genere una continuación plausible del texto (y a ser posible, veraz); sino que dentro de las posibles continuaciones plausibles, queremos potenciar algunas (como responder preguntas) y desincentivar otras (como hacer más preguntas).

Esto último se consigue con las técnicas denominadas de "ajuste de instrucciones" (*instruction tuning*), que se utilizan para ajustar el modelo de modo que "prefiera" generar el tipo de continuaciones del texto que los humanos queremos. Existe una variedad de técnicas de instruction tuning, pero sobre todo se dividen en dos tipos, que se suelen aplicar en conjunto (Ouyang *et al.,* 2022):

- Ajuste supervisado: se proporcionan al modelo ejemplos del tipo de respuestas que queremos, siguiendo un formato instrucción-respuesta para que el modelo lo imite.
- Aprendizaje por refuerzo con realimentación humana (en inglés, *reinforcement learning with human feedback* o *RLHF*): se proporciona al modelo realimentación humana sobre la calidad de sus respuestas, para que prefiera las respuestas más deseables y se aleje de las menos deseables. Por ejemplo, dadas las respuestas de la tabla 7, generadas por el propio modelo, un evaluador humano marcaría las dos primeras como deseables y la tercera como indeseable. Estos datos se usarían para ajustar el modelo, haciendo que prefiera responder a las preguntas.

Es con estas técnicas cómo se obtienen los *LLMs* tal como los conocen los usuarios finales. Sistemas como ChatGPT o Claude, pues, no son más que modelos neuronales de lenguaje entrenados para proporcionar una continuación plausible de un texto (funcionalmente lo mismo que las cadenas de Markov, pero con mejor desempeño gracias a las tecnologías neuronales y a la mayor capacidad de cómputo y datos) que se someten a un proceso de ajuste para que, dentro de las posibles continuaciones plausibles, prefieran aquéllas que nosotros indicamos como prioritarias.

4. ¿QUÉ CAPACIDADES Y LIMITACIONES TIENEN?

Como hemos visto en la sección anterior, los *LLMs* no están entrenados explícitamente para responder a las peticiones del usuario, ni siquiera para que sus textos reflejen la realidad, sino solamente para generar continuaciones plausibles de un texto a partir de lo observado en los datos de entrenamiento. En este sentido, los modelos pequeños se limitan a generar texto coherente, pero "inventado", que no refleja la realidad ni responde a lo que pide el usuario. Sin embargo, cuando los modelos se hacen más grandes (con más neuronas y conexiones en su red) y se entrenan con mayores volúmenes de datos, espontáneamente el contenido de sus textos se va volviendo más fiable, a la vez que surgen habilidades como la de responder preguntas. A este respecto, es importante resaltar que el ajuste de instrucciones sirve para dar más consistencia a los modelos y hacerlos más previsibles (por ejemplo, como hemos

visto, para que *siempre* respondan a las preguntas que se les hacen, en lugar de solo a veces), y también para mitigar posibles sesgos causados por los datos de entrenamiento (los modelos pueden producir respuestas con sesgos indeseables como machismo o racismo si éstos están presentes en los datos de entrenamiento, como suele ser el caso en conjuntos de datos no filtrados descargados de Internet); pero en ningún caso sirve para añadir capacidades. En los ejemplos vistos en la sección anterior (tabla 7), se puede ver cómo el modelo BLOOM en bruto, sin ningún ajuste, ya es capaz de responder preguntas. El ajuste de instrucciones solo hace que esa capacidad se aproveche de forma más consistente para el usuario final.

Estas nociones sobre el funcionamiento de los *LLMs* nos permiten explicar lo que probablemente sea su mayor limitación tecnológica: el conocido fenómeno de las alucinaciones (Ji *et al.,* 2023), que es como se ha dado en llamar a las situaciones en las que un *LLM* genera texto que suena plausible, pero contiene información factualmente incorrecta o incluso sin sentido. Conociendo el funcionamiento de los *LLMs* podemos deducir que, desde un punto de vista técnico, las alucinaciones no son un fallo del sistema (entendiendo fallo como comportamiento anómalo o no previsto); sino que son una consecuencia directa del funcionamiento normal del sistema: entrenamos modelos para generar texto plausible, y eso es lo que nos dan. Las alucinaciones son especialmente prevalentes cuando el modelo no es capaz de proporcionar una respuesta correcta (por ejemplo, porque se le pregunta por algo que no tiene en su conjunto de entrenamiento). De hecho, Hicks *et al.* (2024) argumentan que las alucinaciones se entienden mejor considerándolas como *bullshit*, en el sentido descrito por Frankfurt en su libro On Bullshit (Frankfurt, 2009). Para Frankfurt, "bullshit" es una forma de comunicación en la que el emisor no se preocupa por la verdad ni la falsedad de lo que dice, sino únicamente por el efecto que sus palabras pueden tener. Según Hicks *et al.* (2024), los grandes modelos de lenguaje serían generadores de *soft bullshit*, que es aquélla que no se emite con intención de manipular. Haciendo un símil humano, sería el tipo de discurso que tienen las personas que no quieren admitir que no saben de un tema, y dicen cualquier cosa para salir del paso cuando se les pregunta sobre él.

Por supuesto, los investigadores en PLN y creadores de *LLMs* están trabajando para reducir las alucinaciones al mínimo posible. El ajuste de instrucciones (Ouyang *et al.,* 2022), visto en la sección anterior, juega un papel importante en ello, pues nos proporciona una forma de incentivar al modelo para que priorice la generación de textos ciertos por encima de los falsos. Otra línea muy activa de investigación es la generación aumentada con recuperación (en inglés, *retrieval-augmented generation* [Lewis *et al.,* 2020]), una técnica en la que se hace que el modelo consulte una base de datos o fuente de información externa para recuperar información relevante antes de generar una respuesta. Esto no solamente permite que el modelo combine su conocimiento interno con información actualizada o específica del contexto, mejorando la precisión de sus respuestas, sino que reduce las alucinaciones porque minimiza las situaciones en las que el modelo no tiene acceso a la respuesta, incurriendo en la soft bullshit que antes mencionábamos. Sin embargo, al usar *LLMs* debemos ser conscientes de que, si bien estas técnicas reducen las alucinaciones lo suficiente para proporcionarnos modelos útiles y aplicables en multitud de situaciones, ninguna de ellas garantiza eliminarlas por completo. Las alucinaciones, como hemos explicado, son una

característica intrínseca de la manera en que están diseñados y entrenados los *LLMs*, y han sido descritas como una "limitación innata" e "inevitable" (Xu *et al.,* 2024) y que va a existir "siempre" (Banerjee *et al.,* 2024). Por lo tanto, al usar LLMs, es importante no otorgar nunca presunción de veracidad a la información que proporcionan, y verificarla por otras fuentes si es importante que sea cierta.

Volviendo a los aspectos positivos de los *LLMs*, lo cierto es que; aunque las alucinaciones existan, en muchos casos estos modelos sí son capaces de proporcionarnos respuestas veraces, hasta el punto de que los mejores *LLMs* alcanzan unas tasas de fiabilidad muy altas siempre que las peticiones que les hacemos no requieran razonamientos complejos, saberes muy especializados, conocimiento de noticias de actualidad, o se enfoquen en algún punto débil de estos modelos, como puede ser la aritmética (los *LLMs* son modelos de lenguaje enfocados a palabras y no tienen acceso directo a qué cifras conforman un número, de ahí que no sean sorprendentes sus fracasos en este aspecto (Gambardella *et al.,* 2024): es pedirles trabajar con información a la que no tienen acceso más que indirectamente, como una persona ciega respondiendo preguntas sobre colores). El hecho es que la cantidad de cosas que *sí* pueden hacer, de manera bastante fiable, es enorme: no solamente demuestran la capacidad de responder preguntas, sino que son capaces de traducir textos, resumirlos, corregirlos, programar en distintos lenguajes de programación, e incluso tienen un buen nivel en escritura creativa.

A modo de inciso, al respecto de esto último, conviene desterrar un mito muy difundido, según el cual los *LLMs* nunca podrían generar textos creativos porque "predicen siempre la palabra más probable" o, como dice Ted Chiang en The New Yorker (Chiang, 2023) hacen "un promedio de las elecciones que otros escritores han hecho, representadas por textos encontrados en Internet; ese promedio equivale a las elecciones menos interesantes posibles, por lo que el texto generado por IA suele ser realmente insulso". Estas descripciones son simplificaciones engañosas que no reflejan el funcionamiento real de los *LLMs*: como hemos visto, éstos funcionan prediciendo la siguiente palabra, pero no es necesariamente ni la palabra más probable, ni un "promedio" que elimine la diversidad o lo atípico. Del mismo modo que las cadenas de Markov, lo que utilizan es una distribución estadística, donde las palabras más probables según los datos de entrenamiento se predecirán más a menudo, y las más atípicas menos a menudo. Pero nada impide que un *LLM* se desmarque con una elección realmente inusual, y de hecho esto incluso se puede fomentar, ajustando un parámetro llamado "temperatura" (Peeperkorn *et al.,* 2024) para aplanar la distribución estadística, disminuyendo la probabilidad de las palabras más probables y aumentando la de las que lo son menos. Y al margen de estas consideraciones, lo cierto es que ya hay resultados que muestran que los *LLMs* pueden generar historias que jueces humanos consideran incluso mejores que las escritas por personas (Gómez-Rodríguez y Williams, 2023). Aunque el resultado depende mucho de factores como longitud de los textos, género literario, idioma empleado, nivel de los escritores humanos con los que se compara y otras condiciones de la tarea, que hace que los resultados de este tipo de experimentos sean diversos (Marco *et al.,* 2024; Chakrabarty *et al.,* 2024); lo cierto es que no hay argumentos teóricos ni empíricos para defender que los textos generados por *LLMs* tengan por qué ser insulsos.

Pero regresando a las habilidades de los *LLMs*, la gran pregunta es: ¿por qué y cómo estos modelos las desarrollan? Lo cierto es que, a día de hoy, no lo sabemos en absoluto. Como se comentó en el apartado 3.3, el hecho de que al hacerse más grandes los modelos de lenguaje dejasen simplemente de generar textos al azar, para proporcionar respuestas con sentido, no fue algo buscado, sino un giro inesperado por los propios investigadores que lo hicieron posible. Se trata de *habilidades emergentes*: capacidades que surgen de manera aparentemente espontánea cuando el sistema alcanza una determinada escala, sin haber sido explícitamente programado o diseñado para ello. Aun hoy, aunque por supuesto hay muchos estudios intentando explicar cómo y por qué surgen estas capacidades (Schaeffer *et al.,* 2024; Du *et al.,* 2024), estamos muy lejos de lograr responder la pregunta. Y esta ignorancia sobre cómo los *LLMs* logran hacer lo que hacen implica, asimismo, una falta de transparencia: hoy por hoy, no podemos explicar por qué un *LLM* está proporcionando una respuesta y no otra, haciendo desaconsejable utilizarlos para cualquier toma de decisiones relevantes, salvo como asistentes que informen a la persona que tome la decisión final.

De hecho, tan poco sabemos hoy por hoy de los *LLMs* que incluso resulta polémico un aspecto que parece que debería ser básico: ¿los *LLMs* entienden (en algún grado) el lenguaje humano? Muchos expertos, como Bender *et al.* (2021), defienden que no lo comprenden en absoluto, basándose sobre todo en su propio funcionamiento. Como hemos visto, los grandes modelos de lenguaje generan palabras basándose en un modelo estocástico, similar en esencia a una cadena de Markov. Son, pues, lo que Bender et al. llaman "loros estocásticos", limitándose a repetir patrones lingüísticos previamente observados en los datos de entrenamiento, sin entender realmente el significado o el contexto en el que se usan. Cuando escogen una palabra dada, no lo hacen basándose en su significado, sino en mera estadística (eligiéndola al azar de una distribución). Tampoco tienen consciencia ni intenciones, con lo cual sus respuestas son lenguaje "falso", vacío de pensamiento o intencionalidad. En un experimento mental (Bender, 2020) que recuerda al clásico argumento de la habitación china de Searle (1985), Bender compara el aprendizaje de los *LLMs* con una persona que, sin saber tailandés ni conocer siquiera su alfabeto, estuviese encerrada en una biblioteca llena de libros en tailandés. El argumento es que, aun disponiendo tiempo ilimitado, sería imposible que esa persona lograra *comprender* realmente la lengua tailandesa: podría tal vez, a partir de patrones observados en los libros, ser capaz de dar respuestas convincentes a oraciones escritas en ese idioma; pero sería un mero ejercicio de reconocimiento de patrones sin comprensión real del lenguaje. Nunca podría deducir lo que significan las palabras porque no tendría referentes del mundo real para poder vincularlas a conceptos y experiencias.

Sin embargo, otros autores no están de acuerdo, y sí ven comprensión real del lenguaje (aun cuando sea diferente a la humana, y pueda tener sus limitaciones) en los *LLMs*. Aunque no sepamos muy bien cómo lo hacen, se ha visto que los *LLMs* más grandes son capaces de responder preguntas que necesitan razonar sobre objetos y propiedades del mundo físico que nunca han observado (Li *et al.,* 2021; Patel y Pavlick, 2022). En la tabla 8 se puede ver una respuesta de ChatGPT ante una petición que, para un ser humano, implicaría hacerse un modelo mental del mundo y pensar en cómo va cambiando el estado de los distintos objetos a medida que suceden los hechos narrados. Como se puede ver, el *LLM* proporciona

la respuesta correcta. Aunque esto no es así el 100 % de las veces, los mejores modelos suelen tener un grado de fiabilidad bastante alto en preguntas como ésta o incluso más complejas, lo cual parece difícil de conseguir sin una cierta capacidad de entender el mundo y, por ende, el significado de las palabras.

Tabla 8.

ChatGPT (con GPT-4) respondiendo una pregunta que parece requerir ciertas capacidades de modelado de mundo

Petición: Estoy en mi habitación. Cojo un vaso lleno de agua, y le pongo un cubito de hielo. Meto el vaso dentro de una caja negra. Llevo la caja negra al salón. Saco el vaso de la caja, vierto su contenido al suelo. Vuelvo a meter el vaso en la caja. Voy a la cocina, y pongo la caja en la encimera. ¿Dónde está el cubito de hielo ahora? ¿Dónde estará dentro de veinte minutos?
Respuesta: Dado que viertes el contenido del vaso al suelo, el cubito de hielo, asumiendo que se encontraba en el vaso, ahora estaría en el suelo del salón. Dentro de veinte minutos, dependiendo de la temperatura de la habitación, es probable que el cubito de hielo se haya derretido, por lo que estaría en forma de agua en el suelo del salón.

Si bien se podría argumentar que tal vez los *LLMs* sean capaces de responder este tipo de preguntas mediante razonamientos superficiales o reconocimiento de patrones, sin un verdadero modelo interno del mundo, hay un experimento que proporciona pruebas sólidas de su capacidad de modelado, al menos en un entorno de juguete: Li *et al.* (2023) entrenan un modelo GPT-2 sobre listas de jugadas de un juego de mesa, Othello. El entrenamiento se lleva a cabo desde cero, es decir, el modelo no ha tenido acceso a ningún otro tipo de información en absoluto, ni siquiera ha estado expuesto a lenguaje humano. Lo único que ha visto son listas de jugadas (como "C3" o "D4") procedentes de partidas. En principio, del mismo modo que la persona encerrada en la biblioteca tailandesa no tiene información que le permita vincular las palabras que aparecen en los libros con objetos o conceptos; este modelo no tiene información que le permita saber que las listas que recibe se corresponden con un juego de tablero, cómo es dicho tablero o el número de jugadores. Sin embargo, el experimento de Li *et al.* muestra cómo el sistema, solamente a partir de las listas de jugadas, es capaz de jugar correctamente (sugiriendo jugadas legales más del 99 % de las veces) y generar una representación interna del tablero (64 neuronas que representan cada una de las casillas del tablero 8x8 del juego, cosa que los autores pudieron comprobar manipulando los bits de esta representación interna y comprobando su efecto en la partida).

Parece, pues que, de alguna manera, los grandes modelos de lenguaje son capaces de inferir *significados* a partir de las cadenas de texto que reciben: aunque, como el prisionero de la biblioteca tailandesa, no dispongan de información sobre la correspondencia entre esas cadenas y entidades reales; consiguen hacerse una idea del funcionamiento del mundo exterior (que en el trabajo de [Li *et al.,* 2023], sería el juego de Othello), a partir de dichas cadenas. Y todo esto, a pesar de que en ningún momento hemos dejado de hablar de modelos entrenados, simple y exclusivamente, para predecir la continuación plausible de un texto (o, en el caso de Othello, de una lista de jugadas). El modelo de Li *et al.* construye un modelo del tablero de

Othello porque *la mejor forma de predecir la siguiente jugada de una partida es comprender el juego*; así que con este único objetivo, es capaz de adquirir comprensión (un modelo) de cómo funciona Othello. Del mismo modo, se podría hipotetizar que la mejor forma de predecir la siguiente palabra en un texto es comprender el mundo al que hace referencia el texto, y de ahí podría provenir la capacidad de modelado de mundo de los *LLMs*; aunque la manera en la que la logran siga siendo un misterio.

En cualquier caso, las obvias lagunas que aún existen en nuestro entendimiento de cómo los *LLMs* adquieren sus habilidades hacen que el debate continúe abierto sobre la cuestión de si comprenden o no el lenguaje: ni tenemos una refutación concluyente de la posibilidad de que un "loro estocástico" lo suficientemente grande y complejo no pueda adquirir una comprensión del lenguaje, ni tenemos pruebas concluyentes de que sea así, dado que el de Othello no deja de ser un ejemplo de juguete en comparación con la complejidad de comprender el lenguaje humano.

De hecho, el debate que suscita esta cuestión ha avivado, a su vez, el debate sobre qué significa exactamente "comprender" el lenguaje humano, algo en lo que estamos lejos del consenso (Søgaard, 2022; Havlík, 2023). Y es que la aparición de estos sistemas que son capaces de manejar el lenguaje de maneras que hasta ahora eran exclusivas de los humanos plantea profundas cuestiones relacionadas con el antropomorfismo. Es frecuente caer en el error de describir a los *LLMs* utilizando términos antropomórficos (Abercrombie, *et al.,* 2023) y atribuirles cualidades o intenciones humanas, ya sea para alabar su funcionamiento (por ejemplo, atribuyéndoles empatía o emociones) o para criticarlo (por ejemplo, acusándolos de mentir o engañar al usuario). No obstante, también es cuestionable descartar por completo que una entidad no humana y no consciente pueda ser el sujeto de verbos tradicionalmente exclusivos de los humanos, como "razonar", si en la práctica estos modelos logran resultados que a menudo resultan indistinguibles de los que obtendría un ser humano al realizar esas mismas acciones (Huang y Chang, 2023). Por otra parte, tampoco podemos descartar que algunas de las barreras que actualmente existen entre la inteligencia humana y las capacidades de los *LLMs* se vayan difuminando cada vez más a medida que los avances técnicos continúen mejorando estos últimos. Por ejemplo, Chalmers (2024) advierte de que aunque es improbable que los modelos de lenguaje actuales sean conscientes, debemos tomar en serio la posibilidad de que sus sucesores pudiesen llegar a serlo en un futuro.

5. CONCLUSIÓN

Tras explicar por qué los *LLMs* son radicalmente diferentes a anteriores tecnologías del lenguaje (sección 2), la descripción de cómo funcionan (sección 3) nos ha permitido comprender mejor sus principales capacidades y limitaciones (sección 4). Al ser esencialmente sistemas que predicen una continuación plausible de un texto a partir de distribuciones estadísticas obtenidas de los datos de entrenamiento, están sujetos a problemas como la presencia de sesgos procedentes de dichos datos, así como la generación de respuestas falsas

o sin sentido ("alucinación"). Estas limitaciones no son comportamientos anómalos, sino consecuencias inherentes al funcionamiento de estos sistemas, que se pueden mitigar hasta cierto punto con diversas técnicas (de las cuales hemos citado el ajuste de instrucciones) pero, al menos de momento, sin garantías de eliminarlas por completo.

Por otra parte, y siendo conscientes de estas limitaciones cuando los usamos, los *LLMs* resultan muy útiles dado su manejo del idioma y su enorme versatilidad, siendo capaces de realizar una gran cantidad de tareas que involucran textos. Esto incluye la generación de respuestas que, para un ser humano, requerirían habilidades como creatividad, razonamiento lógico o conocimiento de la realidad física. Aunque es debatible hasta qué punto se puede decir que los *LLMs* poseen realmente estas capacidades o si solo actúan como loros imitadores y cualquier muestra aparente de creatividad o razonamiento en sus respuestas es una ilusión; lo cierto es que desde un punto de vista práctico, el resultado final a menudo es indistinguible de si las tuviesen.

No obstante, no debemos perder de vista que ahora mismo no sabemos cómo surgen las capacidades emergentes de los *LLMs,* y esto hace que su funcionamiento nos resulte muy opaco. Por lo tanto, y teniendo en cuenta también las observaciones previas sobre sesgos y alucinaciones, nunca se debe confiar en un *LLM* para tomar una decisión relevante.

Por último, cabe puntualizar que este capítulo se ha centrado en tratar de arrojar luz sobre los aspectos del debate público que hacen referencia a sus capacidades y limitaciones, que se derivan directamente de su funcionamiento. Quedan fuera del alcance de este capítulo cuestiones que escapan a los aspectos técnicos de los *LLMs*, y se centran más bien en consideraciones sociales, éticas, económicas o legales de su utilización: por ejemplo, las implicaciones legales de usar material protegido por copyright para entrenar *LLMs* sin permiso de los autores; los riesgos creados por su uso para generar información falsa, manipular a personas o cometer fraude académico; la posibilidad de que los *LLMs* acaben sustituyendo masivamente puestos de trabajo; la huella de carbono que generan debido a sus considerables requisitos computacionales; o el oligopolio que las grandes empresas tecnológicas ostentan sobre los mejores modelos. Si bien la discusión detallada de estos debates queda fuera de este capítulo, confío en que el análisis aquí presentado pueda contribuir también a abordar estas cuestiones en otros contextos.

Referencias

Abercrombie, G. et al. (2023). Mirages. On Anthropomorphism in Dialogue Systems. En H. Bouamor, J. Pino y K. Bali, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (4776-4790). Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.290. https://aclanthology.org/2023.emnlp-main.290

Banerjee, S., Agarwal, A. y Singla, S. (2024). LLMs Will Always Hallucinate, and We Need to Live With This. *arXiv*: 2409.05746 [stat.ML]. https://arxiv.org/abs/2409.05746

Bender, E. M. (2020). *Thought Experiment in the National Library of Thailand*. Accessed: 2024-11-02. 2020. https://medium.com/@emilymenonbender/thought-experiment-inthe-national-library-of-thailand-f2bf761a8a83

Bender, E. M. et al. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? En *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. FAccT '21*. (610-623). Virtual Event, Canada: Association for Computing Machinery, 2021. ISBN: 9781450383097. doi: 10.1145/3442188.3445922. https: //doi.org/10.1145/3442188.3445922

Bommasani, R. et al. (2021). On the Opportunities and Risks of Foundation Models. En *ArXiv.* https://crfm.stanford.edu/assets/report.pdf

Brown, P. F. et al. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, 16.2, 79-85. https://aclanthology.org/J90-2002

Brown, T. et al. (2020). Language Models are Few-Shot Learners. En H. Larochelle et al. (eds.) *Advances in Neural Information Processing Systems* (1877-1901). Vol. 33. Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Chakrabarty, T. et al. (2024). Art or Artifice? Large Language Models and the False Promise of Creativity. En: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. CHI '24. Honolulu, HI, USA: Association for Computing Machinery. ISBN: 9798400703300. doi: 10.1145/3613904.3642731. https://doi.org/10.1145/ 3613904.3642731

Chalmers, D. J. (2024). Could a Large Language Model be Conscious? arXiv: 2303.07103 [cs.AI]. https://arxiv.org/abs/2303.07103

Chen, S. F. y Goodman, J. (1996). An Empirical Study of Smoothing Techniques for Language Modeling. En *34th Annual Meeting of the Association for Computational Linguistics* (310-318). Santa Cruz, California, USA: Association for Computational Linguistics. doi: 10.3115/981863.981904. url: https://aclanthology.org/P96-1041

Chiang, T. (2023). Why A.I. Isn't Going to Make Art. *The New Yorker.* Accessed: 2024-11-02: https://www.newyorker.com/culture/the-weekend-essay/why-aiisnt-going-to-make-art

Cho, K. et al. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. En A. Moschitti, B. Pang y W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (1724-1734). Doha, Qatar: Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. url: https://aclanthology.org/D14-1179

Chomsky, N. (1957). *Syntactic Structures*. The Hague: Mouton y Co.

Devlin, J. et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. En J. Burstein, C. Doran y T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers) (4171-4186). Minneapolis, Minnesota: Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. url: https://aclanthology.org/N19-1423

Du, Z. et al. (2024). Understanding Emergent Abilities of Language Models from the Loss Perspective. *arXiv*: 2403.15796 [cs.CL]. https://arxiv.org/abs/2403.15796

Dubey, A. et al. (2024). The Llama 3 Herd of Models. *arXiv*: 2407.21783 [cs.AI]. https://arxiv.org/abs/2407.21783

Fang, T. et al. (2023). Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation. *arXiv*: 2304.01746 [cs.CL]. https://arxiv.org/abs/2304.01746

Frankfurt, H. On Bullshit. Princeton University Press, 2009. isbn: 9781400826537. url: https://books.google.es/books?id=bFpzNItiO7oC

Gage, P. (1994). A new algorithm for data compression. *C Users J*. 12.2 (feb. de 1994), 23-38. ISSN: 0898-9788.

Gambardella, A., Iwasawa, Y. y Matsuo, Y. (2024). Language Models Do Hard Arithmetic Tasks Easily and Hardly Do Easy Arithmetic Tasks. En L.-W. Ku, A. Martins y V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (85-91). Bangkok, Thailand: Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.8. https://aclanthology.org/2024.acl-short.8

Goldberg, Y. e Hirst, G. (2017). *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers. ISBN: 1627052984.

Gómez-Rodríguez, C. y Williams, P. (2023). A Confederacy of Models: a Comprehensive Evaluation of LLMs on Creative Writing. En H. Bouamor, J. Pino y K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023* (14504-14528). Singapore: Association for Computational Linguistics. doi: 10.18653/v1/2023. findings-emnlp.966. url: https://aclanthology.org/2023.findings-emnlp.966

Gómez-Rodríguez, C. y Williams, P. (2024). The Unlikely Duel: Evaluating Creative Writing in LLMs through a Unique Scenario. En: XX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2024), 225-226. A Coruña, Spain: Asociación Española para la Inteligencia Artificial. ISBN: 978-84-09-62724-0.

Goodfellow, I. J., Bengio, Y. y Courville, A. (2016). *Deep Learning*. Cambridge, MA, USA: MIT Press. http://www.deeplearningbook.org/

Gu, A. y Dao, T. Mamba. (2024). Linear-Time Sequence Modeling with Selective State Spaces. En First Conference on Language Modeling. https://openreview.net/forum?id=tEYskw1VY2

Havlík, V. (2023). Meaning and understanding in large language models. *arXiv*: 2310.17407 [cs.CL]. https://arxiv.org/abs/2310.17407

Hicks, M. T., Humphries, J. y Slater, J. (2024). ChatGPT is bullshit. *Ethics and Inf. Technol.*, 26.2. ISSN: 1388-1957. doi: 10.1007/s10676-024-09775-5. url: https://doi.org/10.1007/s10676-024-09775-5

Hirschberg, J. (1998). Every Time I Fire a Linguist, My Performance Goes Up, and Other Myths of the Statistical Natural Language Processing Revolution. Invited talk, *Fifteenth National Conference on Artificial Intelligence (AAAI-98)*

Huang, J. y Chang, K. C.-C. (2023). Towards Reasoning in Large Language Models: A Survey. En A. Rogers, J. Boyd-Graber y N. Okazaki, Findings *of the Association for Computational Linguistics: ACL 2023*. (1049-1065). Toronto, Canada: Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.67. https://aclanthology.org/2023.findings-acl.67

Hutchins, W. J. (2024). The Georgetown-IBM experiment demonstrated in January 1954. En R. E. Frederking y K. B. Taylor (eds.) *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas: Technical Papers* (102-114). Washington, USA: Springer. https://link.springer.com/chapter/10.1007/978-3-540-30194-3_12

Jelinek, F. (1980). Interpolated estimation of Markov source parameters from sparse data. https://api.semanticscholar.org/CorpusID:61012010

Ji, Z. et al. (2023). Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55.12. ISSN: 0360-0300. doi: 10.1145/3571730. url: https://doi.org/10.1145/3571730

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. En: *IEEE Transactions on Acoustics, Speech, and Signal Processing 35.3* (400-401). doi: 10.1109/TASSP.1987.1165125

LeCun, Y., Bengio, Y. e Hinton, G. (2015). Deep learning. *Nature*, 521.7553, 436.

Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

Li, B. Z., Nye, M. y Andreas, J. (2021). Implicit Representations of Meaning in Neural Language Models. En C. Zong et al. (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. (1813-1827). Online. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. https://aclanthology.org/2021.acl-long.143

Li, K. et al. (2023). Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. En *The Eleventh International Conference on Learning Representations.* https://openreview.net/forum?id=DeG07_TcZvT

Liu, Y. et al. (2024). Datasets for Large Language Models: A Comprehensive Survey. *arXiv*: 2402.18041 [cs.CL]. https://arxiv.org/abs/2402.18041

Manning, C. D. y Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.*Cambridge, Massachusetts: The MIT Press. http://nlp.stanford.edu/fsnlp/

Manning, C. et al. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. En K. Bontcheva y J. Zhu (eds.). *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (55-60). Baltimore, Maryland: Association for Computational Linguistics. doi: 10.3115/v1/P14- 5010. https://aclanthology.org/P14-5010

Marco, G. et al. (2024). Pron vs Prompt: Can Large Language Models already Challenge a World-Class Fiction Author at Creative Text Writing? En Y. Al-Onaizan, M. Bansal e Y.-N. Chen (EDS.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (1954-1967). Miami, Florida, USA: Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1096. https://aclanthology.org/2024.emnlp-main.1096

Mikolov, T. et al. (2013). Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3781. http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781

Ouyang, L. y et al. (2022). Training language models to follow instructions with human feedback. En S. Koyejo et al. (eds.). *Advances in Neural Information Processing Systems. Ed. por. Vol. 35. Curran Associates* (27730-27744). https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

Patel, R. y Pavlick, E. (2022). Mapping Language Models to Grounded Conceptual Spaces. En: *International Conference on Learning Representations.* https://openreview.net/forum?id=gJcEM8sxHK

Peeperkorn, M. et al. (2024). Is Temperature the Creativity Parameter of Large Language Models? *arXiv:* 2405.00492 [cs.CL]. https://arxiv.org/abs/2405.00492

Peng, K. et al. (2023). Towards Making the Most of ChatGPT for Machine Translation. En H. Bouamor, Pino, J. y K. Bali (eds.), Findings of the Association for Computational Linguistics: EMNLP 2023 (5622-5633). Singapore: Association for Computational Linguistics. doi: 10. 18653 / v1 / 2023. findings - emnlp. 373. https://aclanthology.org/2023.findings-emnlp.373

Pennington, J., Socher, R. y Manning, C. (2014). GloVe: Global Vectors for Word Representation. En A. Moschitti, B. Pang y W. Daelemans (Eds.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (1532-1543). Doha, Qatar: Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. url: https://aclanthology.org/D14-1162

Pu, X., Gao, M. y Wan, X. (2023). Summarization is (Almost) Dead. *arXiv*: 2309.09558 [cs.CL]. https://arxiv.org/abs/2309.09558

Schaeffer, R., Miranda, B. y Koyejo, S. (2024). Are emergent abilities of large language models a mirage? En *Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23.* New Orleans, LA, USA: Curran Associates Inc.

Searle, J. R. (1985). Minds, brains, and programs. En *Mind Design.* (282-307). Cambridge, MA, USA: MIT Press. ISBN: 0262580527.

Søgaard, A. (2022). Understanding models understanding language. English. *Synthese*, 200.6., 1-16. ISSN: 0039-7857. doi: 10.1007/s11229-022-03931-4

Spanish Concordancer - Bruno Spanish Corpus. Accessed: 2024-10-22. 2010. https://www.lextutor.ca/conc/span/

Turing, A. M. (1950). Computing Machinery and Intelligence. English. *Mind. New Series*, 59.236, (433-460). ISNN: 00264423. http://www.jstor.org/stable/2251299

Vaswani, A. et al. (2017). Attention is All you Need. En I. Guyon et al., *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://arxiv.org/abs/1706.03762

Wei, J. et al. (2022). Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.*, 2022. http://dblp.uni-trier.de/db/journals/tmlr/tmlr2022.html#WeiTBRZBYBZMCHVLDF22

Weizenbaum, J. (1966). ELIZA a Computer Program for the Study of Natural Language Communication Between Man and Machine. En *Commun.* ACM 9.1 (ene. de 1966), (36-45). ISSN: 0001-0782. doi: 10.1145/365153.365168. url: http://doi.acm.org/10.1145/365153.365168

Workshop, B. et al. (2023). BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv:* 2211.05100 [cs.CL]. https://arxiv.org/abs/2211.05100

Xu, Z., Jain, S. y Kankanhalli, M. (2024). Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv*: 2401.11817 [cs.CL]. https://arxiv.org/abs/2401.11817

CAPÍTULO V

Transformación de la movilidad urbana: aplicaciones y perspectivas de la inteligencia artificial

Ibai Laña

Las técnicas de inteligencia artificial y *big data* se han convertido en un fuerte atractor de interés mundial dentro de la industria del transporte. La combinación de tecnologías disruptivas y nuevos conceptos como la *Smart City* actualiza el ciclo de vida de los datos de transporte. En este contexto, la inteligencia artificial se considera una nueva baza para que la industria del transporte gestione eficazmente todos los datos que este sector necesita para ofrecer medios de transporte más seguros, limpios y eficientes, así como para que los usuarios personalicen su experiencia de transporte.

Palabras clave: movilidad urbana, inteligencia artificial, optimización,vehículo autónomo.

1. INTRODUCCIÓN

El volumen y la velocidad a la que se generan hoy en día los datos sobre transporte y movilidad han superado con creces las escalas a las que solían captarse, procesarse y analizarse a principios de este siglo. La combinación de nuevos paradigmas de digitalización como el Internet de las cosas (IoT), la proliferación de *Smart Cities* (Zanella *et al.,* 2014), el fuerte descenso de los costes de almacenamiento de datos en el mercado del silicio, los últimos avances en tecnologías inalámbricas y el uso generalizado y de bajo coste de sensores o dispositivos personales ha mejorado drásticamente la capacidad del ser humano para hacerse una idea más detallada de la realidad del transporte y la movilidad (Meekan *et al.,* 2017) gracias a formas más diversas y actualizadas de recopilar, transmitir, almacenar, fusionar, recuperar y procesar datos sobre el transporte (Stathopoulos *et al.,* 2017). Debido al volumen, la variedad (fuente, tipo y formato) y la variabilidad (cambios frecuentes de datos que complican descifrar su significado exacto dentro de su contexto) de los datos de transporte y movilidad, las tareas intensivas en datos como la integración (Khattak *et al.,* 2017), la visualización (Andrienko *et al.,* 2017), la consulta y el análisis para sistemas a gran escala en tiempo real (Amini *et al.,* 2017) son cada vez más importantes en el diseño y la implantación de sistemas de transporte inteligentes (*ITS*, de su nombre en inglés *Intelligent Transportation Systems*). Es un hecho ampliamente reconocido que las aplicaciones *ITS* actuales presentan funcionalidades limitadas de monitorización y análisis de datos (Suh *et al.,* 2017). Como resultado, los retos mencionados exigen funciones realistas y eficaces de supervisión, toma de decisiones y gestión de datos, cuyos requisitos no suelen cumplir los *ITS* desplegados actualmente. Además, la llegada masiva de fuentes de datos en tiempo real conduce, a través de las técnicas de agregación, fusión y aprendizaje incremental, a conocimientos sin precedentes y mejoras relevantes para la política de servicios, actividades y operaciones de transporte.

En este contexto, las tecnologías de inteligencia artificial (IA) y *big data* se han postulado como herramientas clave para la industria del transporte (Rusitschka y Curry, 2016). Las soluciones que se encuentran dentro de este nuevo paradigma tecnológico permiten capturar, gestionar y analizar enormes volúmenes de datos estructurados y no estructurados para mejorar el dominio del transporte y resolver los retos planteados anteriormente. La respuesta pasa por construir nuevos *ITS* y servicios de movilidad basados en los principios y tecnologías del paradigma *big data* y extraer el valor y los conocimientos de la nueva ingente cantidad de datos disponibles. El objetivo final es garantizar que la industria del transporte obtenga valor de sus datos, lo que en el caso concreto del transporte se traducirá en futuros desarrollos centrados en formas de ofrecer métodos de transporte más seguros, limpios y eficientes y experiencias de transporte agradables a sus usuarios finales. Por otra parte, esta disponibilidad de datos habilita el análisis predictivo y en general las técnicas de modelado basado en datos para aplicarlo a dominios tan diversos como la conducción autónoma, el mantenimiento predictivo de infraestructuras [Jeong *et al.,* 2016; Wang y Guo, 2016], el desarrollo de servicios de movilidad con valor añadido (Fonzone *et al.,* 2016; Su *et al.,* 2016), una mejor comprensión de las necesidades de los usuarios (Chen *et al.,* 2016)] o la visualización de los flujos de personas y su evolución en las ciudades (Kitchin, 2015).

La inteligencia artificial y el procesamiento masivo de datos están revolucionando el sector del transporte y la logística, ofreciendo oportunidades sin precedentes de eficiencia, optimización e innovación. La transformación que estas tecnologías están aplicando sobre la movilidad urbana es el objeto de este documento, que plantea las principales áreas de actuación de la IA en la movilidad urbana y los caminos a los que se dirige la investigación en este ámbito.

2. DATOS Y *SMART CITIES*

Los datos son una parte fundamental de la conceptualización de una *Smart City* (Colado, *et al.,* 2014)], y su utilización para mejorar la comprensión de los procesos de una ciudad, así como para definir nuevos tipos de servicios, fomenta el despliegue de más sensores y sistemas de captura de datos. Esta situación favorece que las ciudades inteligentes se comporten como grandes generadores de conjuntos de datos que, a menudo, son publicados en portales de datos abiertos, a disposición de ciudadanos y organizaciones que los pueden utilizar y generar nuevas necesidades que, a su vez, promueven el despliegue de nuevos sistemas de captura de datos.

Dentro del amplio conjunto de ámbitos que abarca el concepto de *Smart City*, los relacionados con la movilidad se pueden compendiar en las siguientes categorías:

- *Gestión de la movilidad urbana.* La perspectiva de la administración pública es esencial en movilidad urbana, ya que es de donde parten las políticas que definen tanto la disponibilidad de infraestructuras y su configuración, como la red de transporte público. Las administraciones son un gran consumidor de datos de movilidad para la toma de decisiones. Desde la generalidad de la elaboración de los planes de movilidad hasta la especificidad de la simulación de la afección al tráfico de una intervención viaria, cualquier medida tomada se ve ampliamente beneficiada si se dispone de datos que permitan analizarla y predecir sus consecuencias. Precisamente por este motivo, las administraciones públicas son, al mismo tiempo, grandes generadores de datos de movilidad, ya que son los entornos donde más sensores y elementos de captura de datos de movilidad se despliegan (Liu y Dijk, 2022).

- *Logística urbana y transporte privado.* Las operaciones de logística urbana han experimentado un incremento notable en los últimos años, debido principalmente al auge del comercio online y al reparto de mercancías asociado al mismo (Lauenstein y Schank, 2022). Además del transporte de bienes, las tecnologías digitales han impulsado la aparición y crecimiento de sistemas privados de transporte de pasajeros alternativos a los tradicionales, e incluyen modalidades nuevas como la movilidad compartida o el transporte a la demanda. Tanto los primeros como los segundos suponen un incremento del tráfico y de la demanda de infraestructuras viarias, así como del impacto que estos tienen en la calidad de vida en las ciudades (Hensher y Puckett, 2024). La optimización de estos sistemas, así como su integración en las lógicas del transporte urbano constituyen un área de investigación prolífica, centrada fundamentalmente en las aproximaciones basadas en datos y en modelos de inteligencia artificial.

- *Vehículo autónomo.* Los vehículos autónomos conectados y cooperativos (CCAV) están emergiendo como una fuerza transformadora en la movilidad urbana. Todos los sistemas que facultan el funcionamiento autónomo de estos vehículos, desde los que permiten la percepción del entorno hasta los que controlan cada uno de sus actuadores para navegar por la red viaria, están basados en técnicas y modelos de inteligencia artificial. Desde el punto de vista de la movilidad urbana, la seguridad, la interacción con otros usuarios de las vías (mucho más variados en entornos urbanos) y las comunicaciones entre las infraestructuras de la ciudad y los vehículos (V2X) son los principales campos de investigación (Campisi *et al.*, 2021). Los CCAV pueden mejorar significativamente la eficiencia y la seguridad del tráfico, planificar mejor las rutas, reducir los atascos y mejorar la fluidez del tráfico. Esta conectividad también permite maniobras de conducción cooperativa, como el *platooning*, que pueden aumentar aún más la capacidad de las carreteras y la eficiencia del combustible. Todos estos aspectos, además de los relacionados con el control de los propios vehículos, configuran un gran ámbito de investigación relacionado con los vehículos inteligentes y las smart cities.

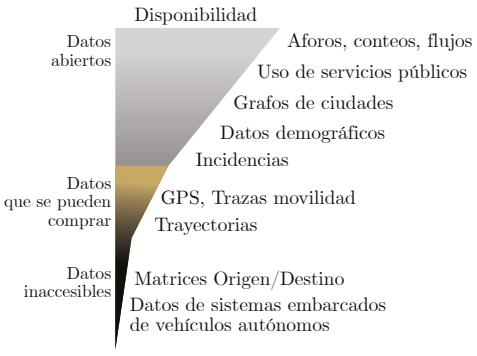
2.1. Fuentes de datos

Los diferentes contextos de aplicación citados anteriormente son, al mismo tiempo, consumidores y generadores de datos de diferente índole y disponibilidad. La figura 1 muestra una representación de la disponibilidad de los tipos de datos más habituales, indicando a su vez los que, por lo general, únicamente pueden obtenerse comprándolos.

Mientras los datos generados por los sensores desplegados por diferentes instituciones en las infraestructuras de movilidad suelen ser publicados en portales de datos abiertos,

Figura 1.

Disponibilidad de diferentes tipos de datos de movilidad urbana



Fuente: Elaboración propia.

y refieren generalmente al uso de dichas infraestructuras, los datos generados por sistemas embarcados en vehículos tienden a no ser públicos. Así, es relativamente sencillo encontrar fuentes de datos con información sobre conteo de vehículos, tiempos de recorrido, u ocupación de diferentes vías, pero la disponibilidad de datos de percepción vehicular generados por el sistema LIDAR de un vehículo son, como norma general, inaccesibles de manera pública. De igual manera, los datos que permiten caracterizar viajes (origen y destino), y que son esenciales para numerosas tareas de gestión y definición de infraestructuras y servicios de transporte público, no suelen estar disponibles de forma pública.

Los datos son un aspecto fundamental en la investigación y desarrollo de sistemas de inteligencia artificial, y su disponibilidad condiciona la forma en la que se produce el avance en los diferentes campos mencionados anteriormente. Así, mientras la investigación de los sistemas de percepción y control de los vehículos se restringe a los contextos de los fabricantes y centros de investigación que tienen acceso a los sistemas implicados, los trabajos relacionados con los datos de alta disponibilidad son abundantes y se extienden mucho más allá del dominio de aplicación.

Es más, como se analizaba en (Manibardo *et al.,* 2021) y se observa en la figura 2, una gran parte del corpus académico relacionado con la movilidad urbana está relacionado con la predicción de tráfico, y en particular con el uso de herramientas de aprendizaje profundo (*Deep Learning*), que implican una alta complejidad y baja interpretabilidad de modelos para producir resultados con un impacto escasamente superior al de modelos de regresión mucho más sencillos. La disponibilidad de conjuntos de datos (datasets) completos y abundantes propicia que investigadores sin interés particular en el dominio de la movilidad copen la investigación en este campo, con el objetivo de testear y demostrar las capacidades de las tecnologías de inteligencia artificial más avanzadas.

Figura 2.

Evolución de las publicaciones de predicción de tráfico basadas en *Deep Learning*, con las fechas de publicación de técnicas específicas de aprendizaje profundo

Fuente: Elaboración propia.

En el lado opuesto, los problemas de optimización inteligente que, para el contexto de la movilidad urbana, especialmente en logística, representan una buena parte de los retos a resolver y requieren menos disponibilidad de datos, acaparan una fracción considerablemente menor de la atención investigadora.

Esta falta de encuentro entre la investigación en inteligencia artificial y los intereses del campo de la movilidad urbana se está viendo mitigada con las tendencias más actuales, explicadas más adelante. Por un lado, el desarrollo del vehículo autónomo acapara una gran parte de la actividad investigadora actual en el campo, y requiere del uso y despliegue de todas las herramientas IA disponibles, así como el desarrollo de nuevas tecnologías para resolver los retos que plantea. Por otra parte, la aplicación de tecnologías IA modernas a problemas clásicos (por ejemplo aprendizaje por refuerzo para resolver problemas de optimización) está generando interés en ambas comunidades. A continuación se muestran algunas de las principales áreas de investigación de aplicación de la IA en la movilidad urbana.

3. INTELIGENCIA ARTIFICIAL APLICADA A LA MOVILIDAD URBANA

Como se ha visto, las herramientas basadas en inteligencia artificial son esenciales en el desarrollo de los llamados sistemas de transporte inteligentes, con un gran impacto en los elementos que configuran la movilidad urbana. Prácticamente cualquier técnica IA que se haya desarrollado tiene aplicación en alguno de los ámbitos citados anteriormente, si bien las aplicaciones principales podrían categorizarse principalmente en las que hacen uso del modelado basado en datos (aprendizaje máquina) y las que se basan en técnicas de optimización inteligente.

Los modelos de aprendizaje automático y aprendizaje profundo tienen especial relevancia en todos los problemas relacionados con los vehículos autónomos y, como se ha visto anteriormente, para tareas predictivas que frecuentemente se relacionan con la estimación del estado o uso de una infraestructura, o bien predicciones de la demanda de un servicio, bien sea de transporte público o logístico. Estas predicciones tienen gran relevancia para la toma de decisiones tanto en el ámbito administrativo como de gestión logística, sin embargo el reto tecnológico que plantean puede considerarse ampliamente resuelto con las tecnologías disponibles a día de hoy. Por esta razón es posible apreciar un cambio de orientación en las tendencias de investigación, en el que las cada vez más avanzadas técnicas de aprendizaje máquina se utilizan principalmente para el contexto del vehículo autónomo, frente a su aplicación clásica para predicción de series temporales.

Por otra parte, las tecnologías de optimización inteligente, que tuvieron un gran relieve académico en la primera década de los 2000, con el auge de los modelos bioinspirados, han experimentado un declive investigador en los últimos años.

No obstante, como se puede observar en la figura 3, los problemas de optimización representan una gran parte de los retos de la movilidad urbana. Este tipo de problemas afectan particularmente al ámbito de la logística, donde reducir la ruta a recorrer, los vehículos a usar o la forma de cargarlos son tareas que se pueden ver ampliamente beneficiadas por las técnicas de optimización inteligente. Fuera de este ámbito, estas técnicas tienen también impacto en aspectos que pueden ir desde la gestión de los ciclos en los semáforos hasta la selección de rutas óptimas para vehículos eléctricos.

Figura 3.

Ejemplos de problemas de optimización en diferentes áreas de la movilidad urbana

Fuente: Elaboración propia.

Así, la movilidad urbana en el contexto de la smart city es un dominio permeado por la inteligencia artificial y sus aplicaciones afectan a todas sus disciplinas de manera profunda y significativa. Desde la optimización del tráfico y la gestión de flotas hasta la mejora de la seguridad y la experiencia del usuario, la IA ya se utiliza de forma práctica y efectiva en diversas aplicaciones. Sin embargo, el potencial de la IA en este campo es aún más prometedor. A continuación, exploraremos algunas de las tendencias de aplicación y de tecnología más impactantes y transformadoras.

3.1. Tendencias por aplicación

En esta sección se muestran las direcciones emergentes que están guiando la investigación y el desarrollo actuales en este campo desde la perspectiva de la aplicación o el problema a resolver. Estas tendencias pueden observarse en la investigación más reciente en el dominio y revelan las potenciales transformaciones en la manera de moverse en las ciudades que podrían tener lugar en los próximos años.

3.1.1. Transporte público a la demanda

El transporte público a la demanda es un modelo de movilidad que se adapta a las necesidades específicas de los usuarios en tiempo real. A diferencia de los servicios de transporte público tradicionales, que operan con horarios y rutas fijas, el transporte a la demanda ofrece cuando y donde lo necesiten. La implementación de este tipo de transporte público ya es una realidad en algunas ciudades (Cabildo de Tenerife, 2024), y suele utilizarse para dar cobertura pública a zonas con poca demanda o inaccesibles para los servicios de transporte regular, combinándose con las líneas de transporte regulares.

El transporte a la demanda representa un reto tecnológico en lo que refiere al cálculo de rutas, ya que, si bien existen algunas implementaciones que se basan en la pre-reserva del servicio, el enfoque general consiste en la solicitud del servicio en tiempo real, lo que implica adaptar las rutas de algunos vehículos dinámicamente, considerando las peticiones actuales, los vehículos disponibles y la calidad del servicio.

El uso de sistemas de optimización inteligente dinámica puede permitir la creación de rutas de transporte público flexible que pueden cambiar sustancialmente el modelo de transporte de las ciudades y sus entornos.

3.1.2. Esquemas de incentivos al transporte público/micromovilidad

El incremento de modos de transporte de pasajeros que ha tenido lugar en los últimos años, así como la gradual incorporación de zonas de bajas emisiones en las grandes ciudades, han favorecido la disponibilidad de alternativas sostenibles al vehículo privado para los desplazamientos intraurbanos. No obstante, la preeminencia de este último está impulsando la promoción desde la administración pública de esquemas de incentivos para el uso de transporte público (EU Urban Mobility Observatory, 2024). Los esquemas de incentivos para el uso de micromovilidad y transporte público son estrategias diseñadas para fomentar el uso de modos de transporte más sostenibles y eficientes en las ciudades. Estos incentivos pueden incluir subsidios, descuentos en tarifas, programas de recompensas y beneficios fiscales para los usuarios que optan por bicicletas, patinetes eléctricos, autobuses, trenes y otros medios de transporte colectivo. Además, las políticas de infraestructura, como la creación de carriles bici y estaciones de carga para vehículos eléctricos, también juegan un papel crucial. Estos esquemas no solo reducen la dependencia de los vehículos privados, sino que también contribuyen a disminuir la congestión, mejorar la calidad del aire y promover un estilo de vida más saludable y sostenible.

Estos esquemas de incentivación no son particularmente novedosos, y tienen un arraigo en aspectos como cuestiones demográficas o culturales, pero la incorporación de los modelos basados en agentes y los sistemas de optimización inteligente están estimulando un área de investigación que permite dar solución al problema de una forma mucho más detallada, y crear incentivos a medida para cada usuario.

3.1.3. Platooning

El *platooning* o circulación en ”pelotón“ es una técnica de conducción en la que varios vehículos se desplazan en convoy, siguiendo de cerca al vehículo delantero de manera automatizada. En el contexto interurbano, el platooning se utiliza principalmente para camiones y vehículos comerciales en autopistas, mejorando la eficiencia del combustible y reduciendo las emisiones gracias a la disminución de la resistencia aerodinámica. En el entorno urbano, el *platooning* puede aplicarse a autobuses y otros vehículos de transporte público, optimizando el flujo del tráfico y mejorando la seguridad vial. Además, el

platooning está empezando a plantearse en entornos urbanos como esquema de funcionamiento para determinados vehículos autónomos, como vehículos de reparto o taxis autónomos. El funcionamiento en convoy de estos vehículos, frecuentemente eléctricos, hace su consumo más eficiente, pero presenta diversos retos en lo que respecta al diseño del sistema de control de unión y separación del pelotón (Wang *et al.,* 2020), el control de la trayectoria o el diseño de la ruta óptima.

Nuevamente, las herramientas IA son fundamentales para la implementación de esta tecnología, permitiendo la coordinación precisa y en tiempo real entre los vehículos. Además, la IA mejora la capacidad de los vehículos para reaccionar a situaciones imprevistas, como cambios repentinos en el tráfico o condiciones climáticas adversas, garantizando así una operación segura y eficiente.

3.1.4. Movilidad urbana aérea

La movilidad urbana aérea está emergiendo como una solución innovadora para descongestionar las ciudades y ofrecer nuevas formas de transporte eficiente y sostenible. Actualmente, diversas empresas están desarrollando prototipos de vehículos aéreos autónomos, conocidos como taxis voladores o eVTOLs (vehículos eléctricos de despegue y aterrizaje vertical), que prometen revolucionar la manera en que nos desplazamos en entornos urbanos (Tecnalia, 2024). Los algoritmos de IA son fundamentales para la navegación autónoma, permitiendo a los vehículos aéreos planificar rutas, evitar obstáculos y aterrizar con precisión. Además, la IA será esencial en la gestión del tráfico aéreo en tiempo real, un contexto altamente restringido y regulado, cuya complejidad ya supone un reto para el tráfico aéreo actual, que se verá multiplicado cuando se trate de espacio aéreo urbano. Sin la IA, la coordinación y la precisión necesarias para operar estos sistemas complejos serían prácticamente imposibles de alcanzar, haciendo de la IA un componente crucial para el futuro de la movilidad urbana aérea.

3.2. Tendencias por tecnología

Tras explorar las tendencias de aplicación que están moldeando la movilidad urbana, esta sección se enfoca en las tendencias de la investigación en tecnologías de inteligencia artificial que pueden tener capacidad de provocar transformaciones cruciales en la forma en la que se resuelven algunos de los problemas descritos anteriormente.

3.2.1. Redes neuronales de grafos

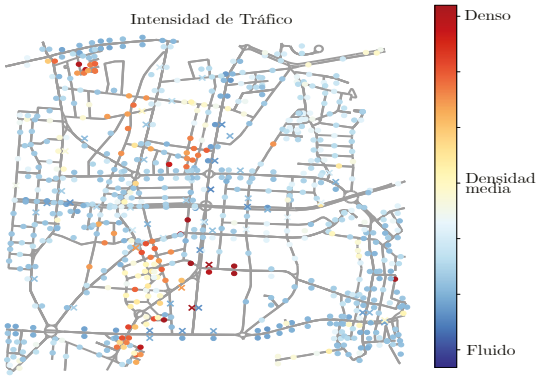
Las redes neuronales de grafos (*GNNs*, por sus siglas en inglés) son una clase de modelos de aprendizaje profundo diseñados para operar sobre datos estructurados en forma de

grafos. Estos modelos extienden las capacidades de las redes neuronales tradicionales para manejar relaciones complejas y estructuras no euclidianas, lo que los hace particularmente adecuados para aplicaciones donde los datos tienen una estructura de grafo, como lo es la red de carreteras de una ciudad. Como los mapas de las ciudades son habitualmente públicos, es relativamente sencillo obtener un grafo que represente la conectividad de intersecciones y carreteras de una ciudad, y por tanto utilizarlo como fuente de partida para entrenar una *GNN*. De hecho, esta tecnología ha adquirido gran relevancia en lo que respecta a las predicciones de tráfico urbanas (un dominio que como se ha visto anteriormente acapara una gran parte de la actividad investigadora). Actualmente es relativamente sencillo encontrar sistemas de predicción de tráfico cuyas fuentes de datos son los sensores en diferentes puntos de la ciudad y el grafo de la misma, obteniendo por un lado predicciones que consideran la topología de la ciudad, así como un sistema para estudiar las relaciones causales entre el tráfico observado en dichas localizaciones (Zhang *et al.,* 2022).

No obstante, su potencial y futuros usos van más allá de obtener predicciones en localizaciones en las que hay sensores. A través de técnicas basadas en características de los grafos, como el grado, la centralidad o la centralidad de intermediación se pueden obtener caracterizaciones de todos los nodos de un grafo y realizar estimaciones de flujo, ocupación o tiempo de recorrido en cada nodo (Manibardo *et al.,* 2023), tal como se observa en la figura 4. Esto tiene implicaciones de mayor alcance, ya que todos los problemas de rutado por los nodos de una ciudad pueden utilizar tiempos de recorrido estimados en cada uno de los nodos, para obtener rutas más óptimas. Igualmente, en otros entornos en los que se disponga de grafo, como almacenes logísticos, las *GNN* pueden permitir desarrollar modelos conscientes de las topologías reales. Así, el uso de *GNNs* tiene un impacto directo en los nuevos desarrollos de aplicaciones de movilidad urbana.

Figura 4.

Ejemplo de aplicación de grafos para estimar el tráfico a nivel de red



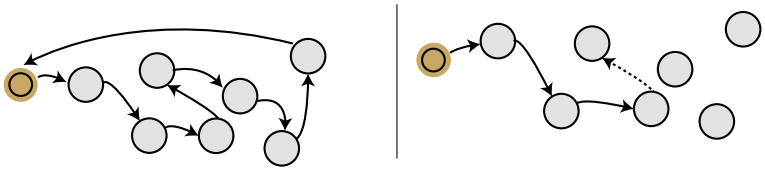
Nota: En la figura se aprecia el tráfico obtenido en los nodos de sensor (marcados con una x), asó como en los nodos generados en cada arista.

3.2.2. Aprendizaje por refuerzo

El aprendizaje por refuerzo es una técnica de inteligencia artificial en la que un agente aprende a tomar decisiones óptimas a través de la interacción con un entorno, recibiendo recompensas o penalizaciones en función de sus acciones, donde el objetivo es maximizar una recompensa acumulada a lo largo del tiempo. Este enfoque es bien conocido y establecido para problemas en los que se implementan soluciones basadas en agentes, y tiene una aplicación directa y clara en el aprendizaje de los sistemas de trayectorias de los vehículos autónomos, por ejemplo. En estos problemas, el vehículo es tratado como un agente que busca trazar una trayectoria entre dos puntos, y para ello dispone de acciones, que son recompensadas o penalizadas en función de lo útiles que sean para realizar la trayectoria de la manera más eficaz. Sin embargo, más allá del uso ”tradicional“ de los sistemas de aprendizaje por refuerzo, existe actualmente una corriente de investigación en la que se está aplicando esta tecnología para resolver problemas de optimización combinatoria. Bajo este paradigma, frente al enfoque de diseñar una solución y evaluarla de acuerdo a una función objetivo, haciendo cambios para evolucionar hacia soluciones mejores, el enfoque basado en aprendizaje por refuerzo construye soluciones dinámicamente y las va penalizando o recompensando de acuerdo con un esquema de recompensas similar a la función objetivo. La figura 5 muestra las diferencias de entrenamiento entre estos dos esquemas. Así, para un problema de rutado en el que hay que encontrar el mejor orden posible para recorrer una lista de nodos, una solución basada en optimización combinatoria (a la izquierda en la figura 5) es construida de una vez, estableciendo un orden particular para navegar por los nodos y evaluando de forma global cómo de buena es esa solución. La solución basada en aprendizaje por refuerzo establece que el agente empieza en uno de los nodos y sus posibles acciones consisten en desplazarse a uno de los nodos disponibles. Cada una de estas acciones es penalizada o recompensada en función de lo buena que es esta decisión de forma global, es decir, considerando la ruta completa. Esta aproximación, mucho más costosa de calcular la primera vez (en su entrenamiento), es capaz de generalizar para mejor para conjuntos grandes de nodos, y obtener soluciones de forma muy rápida, una vez se ha entrenado.

Figura 5.

Diferencias en los esquemas de creación de soluciones para un problema de rutado de vehículos que parte del nodo inicial (en ocre)



Definición completa de solución para resolver con algoritmo de optimización metaheurístico

Definición incremental para resolver con algoritmo de aprendizaje por refuerzo

Nota: Los modelos metaheurísticos crean una solución completa y la evalúan, mientras que las aproximaciones basadas en aprendizaje por refuerzo van calculando la mejor opción a visitar considerando el estado actual del agente y los nodos visitados previamente.

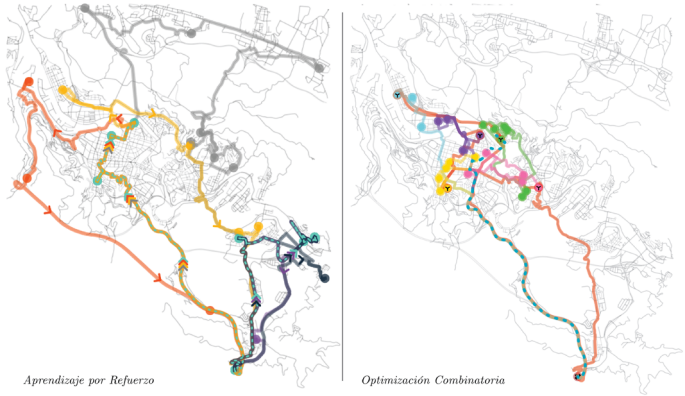
Entrenar un modelo de este tipo es mucho más costoso y elaborado, pero su principal ventaja reside en la capacidad de generalización y la rapidez en tiempo de inferencia. Un sistema de aprendizaje por refuerzo moderno aprenderá, de forma latente en las redes neuronales que lo componen, las características del espacio de solución, y una vez entrenado producirá soluciones nuevas en tiempos de respuesta muy bajos. Este cambio es diferencial respecto a los sistemas de optimización, que requieren un gran esfuerzo computacional cada vez que alguna variable del contexto ha cambiado.

Por este motivo, la utilización de aprendizaje por refuerzo para problemas de optimización puede tener un impacto claro en el ámbito de la movilidad urbana, pudiendo ser empleado para optimizar rutas de transporte, gestionar el tráfico en tiempo real, y mejorar la eficiencia de los sistemas de transporte público, entre otros. Como ejemplo, en (Andres, *et al.,* 2024) se utiliza aprendizaje por refuerzo para resolver un problema de rutado de vehículos frente a una solución equivalente con algoritmos de optimización metaheurística multiobjetivo. El problema está planteado para vehículos autónomos de reparto que comparten parte del recorrido en formación de pelotón (*platoon*), de forma que se produce un aumento de la eficiencia energética del conjunto de vehículos. El objetivo es maximizar el recorrido en formación de pelotón, al mismo tiempo que se minimiza la distancia total recorrida cuando los agentes se separan del mismo para hacer repartos de forma individualizada.

En los resultados mostrados en la figura 6 se puede apreciar que la aproximación de aprendizaje por refuerzo no solamente obtiene mejores soluciones en términos de distancia recorrida,

Figura 6.

Soluciones a un problema de rutado con diferentes agentes que hacen repartos simultáneos basadas en aprendizaje por refuerzo (izquierda) y optimización combinatoria (derecha)



Nota: La solución basada en aprendizaje por refuerzo propone el uso de varios platoons (pelotones de vehículos) que operan de forma simultánea y aumentan considerablemente la ruta compartida. Las rutas compartidas son mostradas con líneas punteadas.

sino también provee de soluciones que no estaban contempladas en la formulación original del problema (como considerar dos *platoons* simultáneos), además de ser capaz de generar soluciones múltiples en tiempos muy bajos. Esta capacidad de adaptación a entornos dinámicos y complejos hace que el aprendizaje por refuerzo sea una herramienta poderosa para abordar los desafíos de la movilidad urbana, permitiendo soluciones más eficientes y sostenibles.

3.2.3. Explicabilidad

La explicabilidad de los modelos de inteligencia artificial (xAI) ha emergido como un aspecto crítico en un contexto en el que los sistemas IA están cada vez más integrados en decisiones que afectan diversas áreas de la vida cotidiana, desde la atención médica hasta las finanzas. La explicabilidad ha cobrado especial relevancia en el contexto europeo, con la vigencia de la AI-Act, que establece diversos estándares de uso de las herramientas IA, muchos de ellos relacionados con la explicabilidad de los mismos. La capacidad de entender cómo y por qué los modelos de IA toman ciertas decisiones no solo mejora la transparencia y la confianza en estos sistemas, sino que también es esencial para identificar y corregir sesgos y errores, acercando las salidas de estos modelos a la comprensión de los expertos que los usan.

En el ámbito de los vehículos autónomos, la explicabilidad de los modelos de IA adquiere una relevancia especial. Estos vehículos dependen de algoritmos complejos para tomar decisiones en tiempo real, como la navegación, la detección de obstáculos y la respuesta a situaciones de emergencia. La falta de explicabilidad puede hacer que estos sistemas sean vulnerables a ataques adversarios, donde pequeñas perturbaciones en los datos de entrada pueden llevar a decisiones incorrectas o peligrosas. La explicabilidad de la IA permite a los desarrolladores y reguladores entender mejor el comportamiento de los algoritmos, identificar posibles puntos débiles y fortalecer la seguridad contra ataques adversarios. Además, la xAI facilita la auditoría y la regulación de estos sistemas, asegurando que cumplan con los estándares de seguridad y ética necesarios para su implementación en entornos urbanos.

3.2.4. IA generativa en la movilidad urbana

La inteligencia artificial generativa y los modelos de lenguaje están en plena ebullición como una de las áreas más innovadoras y de actualidad dentro de la investigación en IA. Estos modelos tienen la capacidad de generar contenido nuevo y coherente, como texto, imágenes, música y más, a partir de patrones aprendidos de grandes volúmenes de datos. La popularidad de estos modelos se debe en gran parte a su versatilidad y a la multitud de aplicaciones que abarcan, desde la creación de contenido creativo y la personalización de experiencias de usuario hasta la automatización de tareas rutinarias y la mejora de la interacción humano-máquina. En principio el uso de estos sistemas parece alejado de cualquiera de los dominios de la movilidad urbana, más allá de su aplicación en los sistemas de interacción de los vehículos autónomos; no obstante, y quizá por el gran atractivo actual que estos sistemas tienen

en la comunidad investigadora, están empezando a ser utilizados en diversos ámbitos. Sus aplicaciones van desde áreas tangenciales al campo como el análisis de redes sociales para predecir el tráfico, hasta la resolución de problemas mucho más relevantes, como su uso para la explicabilidad de modelos produciendo explicaciones textuales o la agilización del entrenamiento de modelos de aprendizaje por refuerzo a través de instrucciones en lenguaje natural. El potencial de estas herramientas para determinadas aplicaciones, especialmente relacionadas con los vehículos autónomos, está todavía por explorar, pero al igual que en otros campos, es posible que la IA generativa tenga un impacto transformador en la movilidad urbana.

4. CONCLUSIONES

El impacto de la inteligencia artificial en la movilidad urbana es palpable actualmente a través de multitud de servicios y tecnologías de movilidad de las que ya disponemos. La inteligencia artificial ya ha transformado la forma en la que desplazamos personas y bienes en las ciudades. Muchos de los servicios de movilidad, que se suelen dar por sentados, como la entrega eficiente de paquetería, se solucionan con herramientas IA, mientras para otros aspectos como los vehículos autónomos, la intervención de herramientas IA es más evidente. Pero la IA no solo ha revolucionado la forma en que gestionamos y optimizamos sistemas de todos los ámbitos de la movilidad, sino que también está abriendo nuevas oportunidades para mejorar la eficiencia, la seguridad y la sostenibilidad en el entorno urbano. Así, el camino recorrido es importante, pero el avance de este conjunto de tecnologías plantea nuevos retos cada día. Como se ha podido observar a través de los diferentes ejemplos mostrados, la IA está transformando la manera en que las personas y los bienes se desplazan dentro de las ciudades, con soluciones cada vez más eficientes, rápidas y humanizadas. A medida que estas tecnologías continúen evolucionando, es esencial que se implementen de manera ética y sostenible, asegurando que los beneficios de la IA sean accesibles y equitativos para todos los ciudanos.

Referencias

Amini, S., Gerostathopoulos, I. y Prehofer, C. Big data analytics architecture for real-time traffic control. In *2017 5th IEEE international conference on models and technologies for intelligent transportation systems (MT-ITS)* (710–715). IEEE.

Andres, A., Laña, I., Bravo, N., Echeverria, I. (2024). Single agent formulation for reinforcement learning based routing of urban last mile logistics with platooning vehicles. En *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, page TBA. IEEE.

Andrienko, G., Andrienko, N., Chen, W., Maciejewski, R. y Zhao, Y. (2017). Visual analytics of mobility and transportation: State of the art and further research directions. *IEEE Transactions on Intelligent Transportation Systems, 18*(8),2232–2249.

Cabildo de Tenerife. (2024). *Transporte a la demanda en tenerife, 2024.* https://www.tenerife.es/portalcabtfe/es/noticias-cabtfenews/177-transportes/16667-el-cabildo-mejora-la-movilidad-de-mas-de-13-000-personas-con-la-puesta-en-marcha-del-transporte-a-la-demanda-en-fasnia-arico-y-guimar#: :text=paradaste

Campisi, T., Severino, A., Al-Rashid, M. y Pau, G. (2021). The development of the smart cities in the connected and autonomous vehicles (cavs) era: From mobility patterns to scaling in cities. *Infrastructures, 6*(7),100.

Chen, C., Ma, J., Susilo, Y., Liu, Y. y Wang, M. (2016). The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation research part C: emerging technologies,* 68, 285–299.

Colado, S., Gutiérrez, A., Vives, C. J. y Valencia, E. (2014). *Smart city: Hacia la gestión inteligente.* Marcombo.

EU Urban Mobility Observatory. (2024). *Mastering mobility with public transport 'microincentives', 2024.* https://urban-mobility-observatory.transport.ec.europa.eu/news-events/news/mastering-mobility-public-transport-microincentives-2024-08-02_en

Fonzone, A., Schmöcker, J-D. y Viti, F. (2016). *New services, new travelers, old models? directions to pioneer public transport models in the era of big data.*

Hensher, D. A. y Puckett, S. (2004). *Freight distribution in urban areas: The role of supply chain alliances in addressing the challenge of traffic congestion for city logistics.*

Jeong, S., Zhang, Y., O'Connor, S., Lynch, J. P., Sohn, H., Law, K. H. (2016). A nosql data management infrastructure for bridge monitoring. *Smart structures and systems, 17*(4), 669–690, 2016.

Khattak, A. J. (2015). Integrating big data in metropolitan regions to understand driving volatility and implications for intelligent transportation systems. In *Information Technology and Intelligent Transportation Systems: Volume 1, Proceedings of the 2015 International Conference on Information Technology and Intelligent Transportation Systems ITITS 2015, held December 12-13, 2015*, Xi'an *China*, (3–4). Springer, 2017.

Kitchin, R. (2015). *Data-driven, networked urbanism.*

Lauenstein, S. y Schank, CH. (2022). Design of a sustainable last mile in urban logistics a systematic literature review. *Sustainability, 14*(9), 5501.

Liu, X. y Dijk, M. (2022). The role of data in sustainability assessment of urban mobility policies. *Data & Policy,* 4, e2,.

Manibardo, E. L., Laña, I. y Del Ser, J. (2021). Deep learning for road traffic forecasting: Does it make a difference? *IEEE Transactions on Intelligent Transportation Systems, 23*(7), 6164–6188.

Manibardo, E. L., Laña, I., Villar-Rodriguez, E. y Del Ser, J. (2023). A graph-based methodology for the sensorless estimation of road traffic profiles. *IEEE Transactions on Intelligent Transportation Systems, 24*(8), 8701–8715.

Meekan, M. G., Duarte, C. M., Fernández-Gracia, J.,Thums, M., Sequeira, A. M. M., Harcourt, R. y Eguíluz, V. M. (2017). The ecology of human mobility. *Trends in ecology & evolution, 32* (3), 198–210.

Rusitschka, S. y Curry, E. (2016). Big data in the energy and transport sectors. *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of big data in Europe*, 225–244.

Stathopoulos, A., Cirillo, C., Cherchi, E., Ben-Elia, E., Li, Y-T. y Schmöcker, J-D. (2017). Innovation adoption modeling in transportation: New models and data. *Journal of choice modelling.*

Su, J-M., Erdenebat, N., Ho, L_H. y Ting Zhan, Y. (2016). Integration of transit demand and big data for bus route design in taiwan. En *Bridging the East and Wes*t (19–26). American Society of Civil Engineers.

Suh, W., Henclewood, D., Guin, A., Guensler, R., Hunter, M. y Fujimoto, R. Dynamic data driven transportation systems. *Multimedia Tools and Applications*, 76, 25253–25269.

Tecnalia. (2024). *Aerotaxi. vehículo volador para el transporte urbano, una alternativa al transporte terrestre.* https://www.tecnalia.com/proyectos/aerotaxi-vehiculo-volador-para-el-transporte-urbano-una-alternativa-al-transporte-terrestre

Wang, S., Homem de Almeida Correia, G. y Lin, H. X. (2020). Effects of coordinated formation of vehicle platooning in a fleet of shared automated vehicles: An agent-based model. *Transportation Research Procedia*, 47, 377–384.

Wang, W. y Guo, F. (2016). Roadlab: Revamping road condition and road safety monitoring by crowdsourcing with smartphone app. *Technical report, Transportation Research Board.*

Zanella, A., Bui, N., Castellani, A., Vangelista, L. y Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal, 1*(1), 22–32.

Zhang, L., Fu, K., Ji, T. y Lu, CH. T. (2022). Granger causal inference for interpretable traffic prediction. En *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)* (1645–1651). IEEE.

CAPÍTULO VI

Interpretabilidad con redes bayesianas*

Pedro Larrañaga

La creciente difusión y popularidad de los sistemas inteligentes basados en aprendizaje automático plantea la disyuntiva de su uso indiscriminado frente a la necesidad de comprender su funcionamiento interno. La explicación *post hoc*, a la que habitualmente recurren los modelos de caja negra, no resulta suficiente en el ámbito científico ni tampoco en situaciones de alto riesgo. En tales casos se requiere que el humano sea capaz de interpretar tanto la salida del sistema como el proceso interno seguido por el mismo hasta alcanzar dicho resultado. En este capítulo analizaremos las potencialidades en interpretabilidad del paradigma conocido como red bayesiana enfatizando su transparencia y versatilidad para llevar a cabo distintos tipos de razonamiento, como por ejemplo el predictivo, diagnóstico, intercausal, contrafáctico, o abductivo. Se mostrará un caso de uso real en neurociencia computacional.

Palabras clave: interpretabilidad, redes bayesianas, neurociencia computacional.

* Agradecimientos: este trabajo ha sido financiado en parte por el Ministerio de Ciencia e Innovación de España a través de los Proyectos AEI/10.13039/501100011033-PID2022-139977NB-I00, PLEC2023-010252 y TED2021-131310B-I00, y por la Comunidad Autónoma de Madrid en el marco de la *ELLIS Unit Madrid* y del proyecto IDEA-CM (TEC-2024/COM-89), junto con el proyecto del CDTI, PLEC2023-010252.

1. INTRODUCCIÓN

La inteligencia artificial se ha convertido en una tecnología disruptiva capaz de transformar aspectos relevantes de nuestras vidas y con una gran repercusión en el desarrollo económico de las naciones. Los medios de comunicación se hacen eco a diario de avances en inteligencia artificial al tiempo que las grandes empresas tecnológicas, la mayoría de las cuales tienen sus sedes en Estados Unidos o en China, llevan años compitiendo por un mercado cada vez más numeroso y que les proporciona ganancias importantes. Los avances conseguidos en estos últimos años en el desarrollo de aplicaciones relativas al tratamiento de imágenes y al procesamiento de lenguaje natural han sido sorprendentes.

A pesar de dichos avances tecnológicos, la realidad en cuanto a su implementación y utilización deja mucho que desear. Así por ejemplo, la Agencia de Administración de Alimentos y Medicamentos del Gobierno de los Estados Unidos norteamericanos (FDA, en su acrónico en inglés) ha ido incrementando anualmente de manera exponencial el número de sistemas inteligentes médicos aprobados, hasta situarse en cifras cercanas a los 120 en estos últimos años, la mayoría de los cuales han sido en el ámbito de la radiología (Muehlematter *et al.,* 2020). Sin embargo la utilización de dichos sistemas inteligentes en el día a día hospitalario tiene una tasa de implantación muy baja. Los motivos aducidos son muy variados, si bien destaca el hecho de que los médicos se resisten a utilizar sistemas que no les explican el porqué de las decisiones aconsejadas. Esta necesidad de entender todo el proceso de razonamiento (tanto la algoritmia en la que se fundamenta la inducción del modelo como la propia cadena de pasos que conduce a una determinada salida) está ocasionando que los humanos requieran de explicaciones ante las decisiones que les proponen los sistemas inteligentes cuando estos se aplican a situaciones que pueden considerarse críticas o de especial relevancia. Dichas situaciones pueden producirse en la generación de nuevo conocimiento científico, en decisiones judiciales acerca de la concesión de libertad provisional a presos, en el diagnóstico y el pronóstico de enfermedades en medicina, en el mantenimiento predictivo de plantas industriales o en la monitorización de puentes en carreteras por las que diariamente pasan miles de coches.

Esta necesidad de explicar tanto los modelos de aprendizaje automático, como los procesos de inducción de los mismos y de los diferentes tipos de inferencia que pueden llevarse a cabo con ellos, ha dado origen a la denominada explicabilidad en inteligencia artificial (XAI) (Gunning *et al.,* 2019). Esta XAI es también recogida por la normativa de regulación sobre la inteligencia artificial (EU AI Act) aprobada recientemente por el Parlamento Europeo. EU AI Act se fundamenta en una jerarquía de riesgos, que van desde mínimo a limitado, alto o inaceptable, y que desde el segundo nivel ya requieren de una transparencia en las decisiones propuestas. Si bien el término XAI se extiende a toda la inteligencia artificial lo cierto es que la inmensa mayoría de su aplicabilidad es a modelos inducidos a partir de métodos de aprendizaje automático. Cualquiera de las tareas que se llevan a cabo dentro del mismo, es decir clasificación supervisada, regresión, *clustering,* selección de variables, aprendizaje por refuerzo u optimización heurística, por citar las más habituales, han ido desarrollando sus propios métodos de explicabilidad, siendo la clasificación supervisada la que ha recibido más atención en la literatura especializada (Molnar, 2022).

La necesidad de explicar modelos de aprendizaje automático calificados como cajas negras (por ejemplo los modelos entrenados con redes neuronales profundas) ha llevado a los investigadores a desarrollar explicaciones basadas en modelos subrrogados que sean transparentes, que si bien proporcionan aproximaciones al desempeño de los modelos originales, muchas veces se encuentran alejados en sus prestaciones (Nakka *et al.,* 2023). Además la dificultad de explicar adecuadamente dichas cajas negras ha ocasionado la existencia de una línea de investigación que desaconseja este tipo de explicaciones en situaciones que sean de vital importancia para el ser humano y que defiende el uso de modelos interpretables en su lugar (Rudin, 2019). Esta es también la aproximación propuesta en este capítulo.

La organización del capítulo es como sigue. La sección 2 describe los conceptos de interpretabilidad y explicabilidad en sistemas inteligentes recalcando sus diferencias fundamentales, enfatizando las tres propiedades que un sistema inteligente debe de cumplir para ser considerado interpretable y describiendo las ventajas de contar con dichos sistemas interpretables. La sección 3 introduce el paradigma de redes bayesianas a partir del concepto de independencia condicional entre tripletas de variables, proporcionando algoritmos de aprendizaje a partir de datos, así como distintos métodos de inferencia tanto exacta como aproximada. En la sección 4 se mostrará cómo las redes bayesianas verifican las tres propiedades necesarias para su consideración como paradigma interpretable, focalizándose a continuación en las potencialidades de interpretabilidad intrínsecas a las redes bayesianas. La sección 5 muestra un ejemplo de utilización de las redes bayesianas en la interpretación de modelos de clasificación morfológica de interneuronas GABA-érgicas. La sección 6 recoje las conclusiones del capítulo junto con propuestas para seguir avanzando en la utilización de las redes bayesianas como modelos interpretables.

2. INTERPRETABILIDAD VERSUS EXPLICABILIDAD

En 2016 la agencia norteamericana de investigación DARPA lanzó un programa donde el foco principal era el desarrollo de modelos de inteligencia artificial que fueran explicables. Los responsables de dicho programa (Gunning *et al.,* 2029) lo motivan por la necesidad de que los usuarios sean capaces de comprender la solución proporcionada por el sistema, así como todas y cada una de las fases que el mismo ha efectuado hasta alcanzar la solución. Se pretendía que el ser humano se involucrase de manera activa en las decisiones sugeridas por el sistema, llegando a aceptarlas incluso en casos en los que no coincida su juicio con el sugerido por el sistema, siempre y cuando la explicación propuesta fuese comprensible y admisible para el experto. Esto generaría la posibilidad de crear un nuevo conocimiento basado en aquellas sugerencias que no coincidiesen con el estado del arte, pero que el experto una vez analizadas y comprendido todo el proceso de decisión estuviese de acuerdo en admitirlas.

Si bien los sistemas inteligentes de caja negra, han sido (están siendo) explicados con la ayuda de otros paradigmas (subrogados) de caja blanca que son inducidos a partir del comportamiento de dichas cajas negras, y por tanto no llegan a explicarlas en realidad, esta

práctica está siendo admitida y dada por buena por una parte importante de la comunidad científica. Sin embargo, en problemas que puedan ser considerados como críticos, tanto porque afectan a aspectos relevantes de la naturaleza humana, como por su relevancia económica, judicial o científica, la necesidad de contar con sistemas inteligentes interpretables es innegable (Rudin, 2019).

Según Lipton, 2019, la interpretabilidad se refiere a la capacidad de un ser humano para comprender el funcionamiento interno del modelo. Es decir, permite que las personas comprendan cómo el modelo llega a sus conclusiones o predicciones. Dicha interpretabilidad descansa en tres propiedades que todo sistema inteligente debe de cumplir para ser considerado como interpretable.

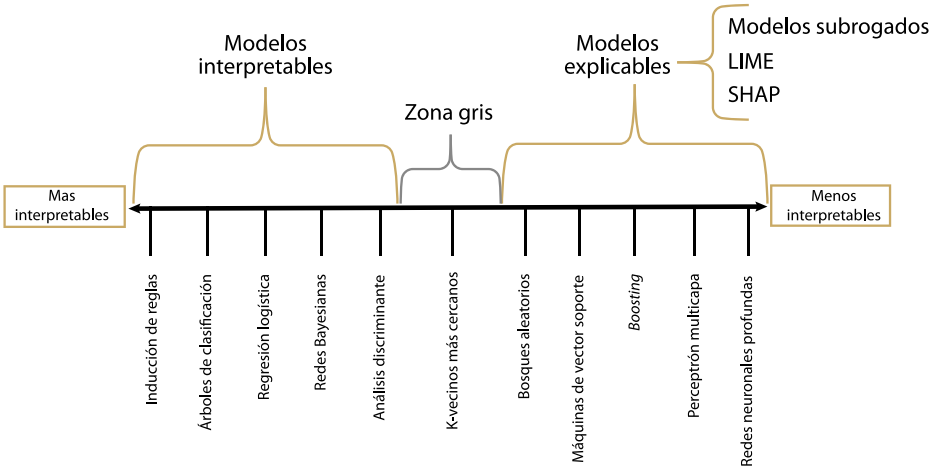
- *Simulabilidad*, la cual hace referencia a la capacidad de un modelo para ser simulado por un ser humano. No se trata solo de que sea sencillo, sino de que una persona pueda reproducir el proceso llevado a cabo por el modelo para sugerir una respuesta. Este concepto se aplica al nivel del modelo completo.
- *Descomponibilidad*, denota la capacidad de descomponer un modelo en partes, para de esta manera facilitar la comprensión del modelo global. Esto se aplica a nivel de los componentes individuales del modelo.
- *Transparencia algorítmica*, expresa la capacidad de comprender el procedimiento que el modelo sigue para generar su salida. Esto se aplica a nivel del algoritmo de entrenamiento del modelo, permitiendo entender cómo el modelo aprende y procesa los datos, así como a nivel del algoritmo de inferencia para entender cómo se ha llegado a la decisión.

La interpretabilidad del modelo nos va a permitir que el humano sea el centro de las decisiones y participe de manera activa en las posibles mejoras del sistema. Por otra parte, un modelo interpretable va a posibilitar la justificación de las decisiones sugeridas por el mismo, la mejora de las prestaciones a nivel cuantitativo, la compresión de sus debilidades, el descubrimiento de nuevo conocimiento, el testeo de su robustez ante situaciones extremas, el análisis de la existencia de sesgos que puedan mediatizar su comportamiento, y la transferencia de los modelos inducidos a problemas en otros dominios de características similares.

La figura 1 muestra una ordenación de algunos tipos de modelos de clasificación supervisada. Los interpretables (inducción de reglas, árboles de clasificación, regresión logística, redes bayesianas, análisis discriminante) se posicionan a la izquierda, mientras que los k-vecinos más cercanos se encuentran en una posición intermedia. En la parte de la derecha nos encontramos con paradigmas que sin ser interpretables pueden ser explicados. Los bosques aleatorios, las máquinas de vector soporte, el *boosting*, el perceptrón multicapa y las redes neuronales profundas forman parte de este grupo. La explicabilidad en estos paradigmas se lleva a cabo usando distintos modelos subrogados, así como por medio de técnicas como LIME, generando aproximaciones locales a las predicciones del modelo, o SHAP que busca calcular la contribución de cada variable predictora al modelo.

Figura 1.

Paradigmas de clasificación supervisada en función de su interpretabilidad



Fuente: Elaboración propia.

Una crítica habitual sobre los modelos interpretables es que sus prestaciones cuantitativas en cuanto a la clasificación no llegan a ser competitivas al compararlas con las que proporcionan los modelos explicativos. Este asunto ha quedado rebatido por varios trabajos empíricos (Lukas-Valentin *et al.,* 2023; Shueli, 2010).

Las redes bayesianas al compararlas con los otros paradigmas interpretables de la figura 1 destacan por su gran versatilidad en cuanto a los diferentes tipos de interpretabilidad que pueden efectuarse con las mismas, algunos exclusivos de este paradigma (véase sección 4).

3. REDES BAYESIANAS

Las redes bayesianas representan explícitamente el conocimiento incierto inherente a un dominio por medio de una representación gráfica que resulta ser muy intuitiva, facilitando la interpretabilidad del modelo. Son capaces de acomodar tanto variables discretas como continuas, e incluso variables que varían en el tiempo. Son también aplicables a situaciones con datos *missing*. El modelo de red bayesiana puede ser obtenido con la ayuda de un experto, automáticamente a partir de datos, o combinando ambas aproximaciones. La aproximación automática resulta de especial interés en la actualidad donde la accesibilidad a grandes bases de datos es cada vez más sencilla, y donde la modelización basada en el experto puede acarrear un gran coste en tiempo a la vez que está sometida a errores derivados del gran número de variables a tratar al unísono.

Una gran ventaja de las redes bayesianas en comparación con otros paradigmas del aprendizaje automático es que con ellas se pueden llevar a cabo la gran mayoría de las tareas habituales en aprendizaje automático: clasificación supervisada tanto en la predicción de una única variable clase (Varando *et al.,* 2015), como cuando nuestro interés radica en predecir varias variables clase a la vez (Bielza y Larrañaga, 2011); regresión de una única variable a predecir como de varias variables (Borchani *et al.,* 2015); clustering probabilista con sus diferentes variantes de multivista o multipartición (Rodriguez-Sanchez *et al.,* 2021); descubrimiento de asociaciones entre variables (Larrañaga *et al.,* 1996); aprendizaje por refuerzo (Valverde *et al.,* 2023); detección de anomalías (Puerto-Santana *et al.,* 2022); selección de variables (Inza *et al.,* 2000) u optimización heurística (Larrañaga y Bielza, 2024).

3.1. Elementos básicos

Supongamos que partimos de un conjunto de datos con N observaciones y n variables, denotadas como $X_1,...,X_n$. Sea $\mathcal{D}=\{\boldsymbol{x}^1,...,\boldsymbol{x}^N\}$ el conjunto de datos, donde $\boldsymbol{x}^h=(x_1^h,...,x_n^h)$, $h=1,...,N$, $\boldsymbol{X}=(X_1,...,X_n)$ y $x_i\in\Omega_{X_i}=\{1,2,...,R_i\}, i=1,...,n$.

Una *red bayesiana* (Pearl, 1988; Koller y Friedman, 2009) es una representación de una distribución de probabilidad conjunta (JPD) $p(X_1,...,X_n)$ sobre un conjunto de variables aleatorias discretas, $X_1,...,X_n$. La *regla de la cadena* permite escribir la JPD como

$$p(X_1,...,X_n)=p(X_1)p(X_2\mid X_1)p(X_3\mid X_1,X_2)...p(X_n\mid X_1,...,X_{n-1}) \qquad [1]$$

La JPD contiene toda la información probabilística sobre el dominio y puede ser usada para responder a cualquier tipo de pregunta en términos probabilistas. El problema de usar la JPD es que su tamaño crece exponencialmente con el número de variables *n*. Las redes bayesianas evitan la necesidad de contar con un número exponencial de parámetros gracias a la consideración del concepto de independencia condicional que involucra a tripletas de variables.

Dos variables aleatorias X e Y son *condicionalmente independentes* (c.i.) dada una tercera variable aleatoria Z si $p(x,y\mid z)=p(x\mid z)\ \ \forall x,y,z$. Es decir, para cualquier $Z=z$, la información $Y=y$ no influencia la probabilidad de $X=x$. La definición puede escribirse de manera alternativa como $p(x,y\mid z)=p(x\mid z)p(y\mid z)\ \ \forall x,y,z$, y también puede extenderse al caso en que X, Y, Z sean vectores disjuntos de variables aleatorias.

El concepto de independencia condicional va a permitir reducir el número de parámetros necesarios para especificar la JPD. Supongamos que para cada X_i somos capaces de encontrar un subconjunto de variables $\boldsymbol{Pa}(X_i)\subseteq\{X_1,...,X_{i-1}\}$ tal que dado $Pa(Xi)$, Xi es c.i. de todas las variables en $\{X_1,...,X_{i-1}\}\setminus\boldsymbol{Pa}(X_i)$, *i. e.,* $p(X_i\mid X_1,...,X_{i-1})=p(X_i\mid\boldsymbol{Pa}(X_i))$. La JPD de la ecuación [1] es ahora:

$$p(X_1,...,X_n)=p(X_i\mid\boldsymbol{Pa}(X_i))...p(X_n\mid\boldsymbol{Pa}(X_n)) \qquad [2]$$

en la que es de esperar que se reduzca sustancialmente el número de parámetros. Esta modularidad va a facilitar el mantenimiento del modelo, así como el desarrollo de procedimientos de razonamiento eficientes que posibiliten su interpretabilidad.

Una red bayesiana representa esta factorización de la JPD con un grafo acíclico dirigido (DAG), el cual constituye la estructura de la red bayesiana. Un grafo G se representa por el par (V,E), donde V es el conjunto de nodos y E es el conjunto de *arcos* entre los nodos en V. Los nodos del DAG representan las variables aleatorias del dominio, $X_1,...,X_n$. Los arcos representan las dependencias probabilísticas entre las variables. Dichas dependencias se cuantifican por medio de distribuciones de probabilidad condicionadas. Los *padres* de un nodo X_i, $\boldsymbol{Pa}(X_i)$, X_i, están constituidos por todos los nodos que apuntan a X_i. Diremos que X_i es un nodo *hijo* de dichos nodos padres. El término *acíclico* significa que el grafo no contiene ciclos, es decir, que no existe una secuencia de nodos que siguiendo la dirección de los arcos comience y termine en el mismo nodo.

Por tanto, la red bayesiana tiene dos componentes: un DAG y un conjunto de distribuciones de probabilidad condicionales asociadas a cada nodo X_i dados sus padres en el DAG, $p(X_i \mid \boldsymbol{Pa}(X_i))$, que determinan una única JPD tal y como se muestra en la ecuación [2]. El primer componente cualitativo se refiere a la *estructura de la red bayesiana*, mientras que el segundo componente cuantitativo hace referencia a los *parámetros de la red bayesiana*. Cuando todas las variables son discretas estos parámetros están tabulados y constituyen la *tabla de probabilidad condicional* (CPT).

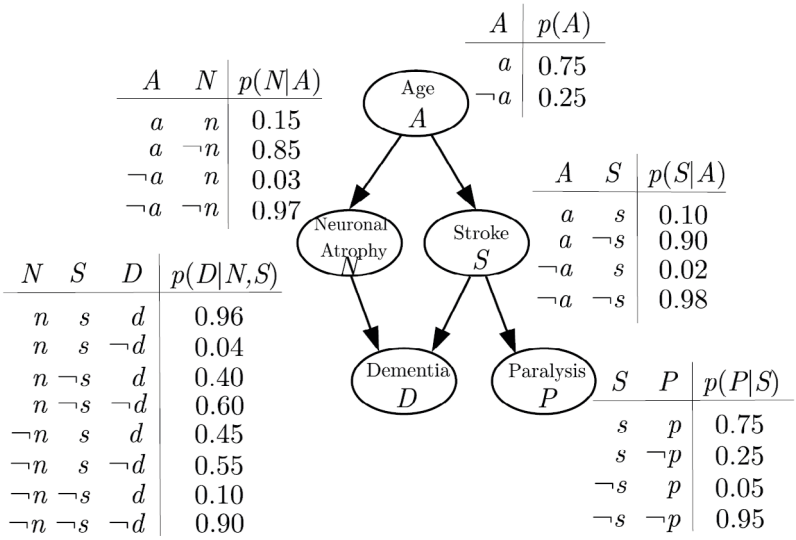
Ejemplo. Riesgo de demencia

La figura 2 muestra un ejemplo hipotético de una red bayesiana modelando el riesgo de demencia. Todas las variables son binarias, con *x* denotando la "presencia" y ¬*x* la "ausencia", para `Dementia` *D*, `Neuronal Atrophy` *N*, `Stroke` *S* y `Paralysis` *P*. Para `Age` *A*, *a* significa "edad +65", en caso contrario el estado es ¬*a*. De la estructura de la red bayesiana se desprende que tanto `Stroke` como `Neuronal Atrophy` están influenciados por `Age` (padre de ambos en el DAG). Ambas variables tienen influencia en `Dementia` (su hijo). `Paralysis` está directamente asociada con `Stroke`. Las CPT indican las probabilidades condicionales asociadas a cada nodo. Por ejemplo, un paciente que tiene "atrofia neuronal" y ha sufrido un "accidente cerebro vascular" tiene una probabilidad de 0,96 de tener "demencia": $p(d \mid n,s) = 0,96$. Sin embargo, en ausencia de "atrofia neuronal" y "accidente cerebro vascular", esta probabilidad es sólo de 0,10, *i. e.*, $p(d \mid \neg n,\neg s) = 0.10$.

La JPD factoriza como $p(A,N,S,D,p) = p(A)p(N \mid A)p(S \mid A)p(D \mid N,S)p(P \mid S)$, requiriendo 11 probabilidades en lugar de los 2^5−1=31 parámetros necesarios en el caso de no usar la red bayesiana.

Figura 2.

Ejemplo hipotético de una red bayesiana modelando el riesgo de demencia



Fuente: Elaboración propia.

La propiedad de independencia condicional entre tripletas de variables puede ser chequeada gráficamete a partir del criterio de *u-separation* (Lauritzen *et al.,* 1990). El criterio para chequear si ***X*** e ***Y*** están u-separados por ***Z*** requiere de tres pasos. En el primer paso se obtiene el menor subgrafo que contiene a ***X, Y*** y ***Z*** y a sus ancestros. En el segundo paso se moraliza el subgrafo resultante, *i. e.,* se añade una arista entre padres con hijos en común y, a continuación, se transforman los arcos en aristas. En el tercer paso, diremos que ***Z*** u-separa ***X*** e ***Y*** si cualquier camino que se pueda establecer desde una variable en ***X*** a una variable en ***Y*** contiene una variable en ***Z***.

3.2. Inferencia

Las redes bayesianas además de proporcionarnos visualizaciones de las relaciones entre las variables del dominio considerado y de darnos la posibilidad de chequear gráficamente independencias condicionales entre tripletas de variables, son modelos muy útiles para llevar a cabo predicciones, diagnósticos y explicaciones. Esto se efectúa por medio del cálculo de la distribución de probabilidad de una variable (o de un conjunto de variables) de interés dados los valores de otras variables. Las variables observadas se denominan *evidencia observada* ***E=e***. A tales efectos existen tres tipos de variables en ***X***: una variable sobre la que se efectúa la pregunta, X_i, (habitualmente una única variable, aunque un vector de variables es también posible), las variables evidencia ***E*** y el resto de variables, que son variables no observadas ***U***.

El término *inferencia* se refiere a encontrar la probabilidad de una variable X_i (o de un vector de variables) condicionado a ***e***, *i. e.,* $p(X_i \mid \boldsymbol{e})$. La teoría de la probabilidad nos proporciona herramientas para computar $p(X_i \mid \boldsymbol{e})$. Esta computación se denomina *razonamiento probabilista* bajo incertidumbre, donde la evidencia se propaga a través de la estructura probabilística actualizando el resto de probabilidades. Si no hay evidencia, las probabilidades de interés son distribuciones *a priori*, $p(X_i)$. La inferencia en redes bayesianas puede combinar evidencia existente en cualquier parte de la red y responder a diferentes tipos de preguntas. Bajo causalidad, podemos predecir los efectos dadas las causas (*razonamiento predictivo*), diagnosticar las causas dados los efectos (*razonamiento diagnóstico*) o explicar una causa como responsable de un efecto (*razonamiento intercausal*). Dicho razonamiento intercausal es único en redes bayesianas. En la estructura $C_1 \rightarrow X \leftarrow C_2, C_1 \; y \; C_2$ son independientes, pero una vez que su hijo en común es observado se convierten en dependientes. Además, cuando el efecto *X* es conocido, la presencia de una causa explicativa hace que la causa alternativa sea menos probable.

Inferencia tambien se refiere a encontrar los valores de un conjunto de variables que mejor explican la evidencia observada. Esto se denomina *inferencia abductiva*. En *abducción total*, se busca el $\arg \max_{\mathbf{U}} p(\mathbf{U}|\mathbf{e})$, *i. e.*, el objetivo es encontrar la *explicación más probable* (MPE), mientras que en la *abducción parcial* el objetivo es similar, en este caso referido a un subconjunto de variables de ***U*** (el conjunto de explicación), y se conoce también bajo el nombre de *máximo a posteriori parcial* (MAP).

Ejemplo. Riesgo de demencia (diferentes tipos de razonamiento probabilístico)

El primer tipo de probabilidades que podemos examinar se refieren a las distribuciones *a priori* de cada variable univariante $p(X_i)$, *i. e.*, sin haber observado ningún tipo de evidencia. La figura 3(a) muestra estas probabilidades usando GeNIe[1] . Por ejemplo, la probabilidad de padecer "demencia" es 0,17.

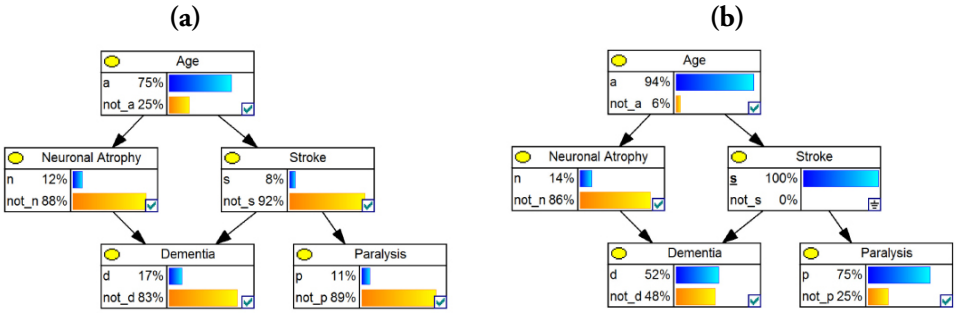
Supongamos que tenemos un paciente que ha sufrido un "accidente cerebro vascular" (*E*=*S*=*s*), es decir el estado *s* de la variable `Stroke` está fijado con la barra al 100 %. Las probabilidades actualizadas dada esta evidencia, *i. e.,* $p(x_i \mid s)$ para los nodos *A,N,D* o *P*, se muestran en la figura 3(b). La probabilidad de sufrir "demencia" se ha incrementado a $p(d \mid s) = 0,52$. Para el estado opuesto –no tener "accidente cerebro vascular"– se obtiene $p(d \mid \neg s) = 0,14$ (no mostrado). Este es un ejemplo de razonamiento predictivo.

Como un ejemplo de razonamiento diagnóstico, dado el efecto de "parálisis" (*P*=*p*), la probabilidad de que la causa sea un "accidente cerebro vascular" es alta, $p(s \mid p) = 0,57$. Para llevar a cabo el razonamiento intercausal, podemos considerar *N, S* y *D*. `Neuronal atrophy` (*N*) y `Stroke` (*S*) son independientes y tenemos $p(n) = 0,12, p(s) = 0,08$ (figura 3(a)), pero una vez que `Dementia` (*D*) es observada, *e. g., D*=*d*, las probabilidades de

[1] https://www.bayesfusion.com/

Figura 3.

(a) Las distribuciones *a priori* p(X_i) se muestran como gráficos de barras, para cada nodo X_i. (b) Después de observar que el paciente ha sufrido un "accidente cerebro vascular" (S=s), las distribuciones se actualizan como p($X_i|s$).



las posibles causas, *n* y *s*, aumentan: $p(n \mid d) = 0,33$ y $p(s \mid d) = 0,25$. La presencia de "atrofia neuronal" (N=n), explicaría la "demencia" observada *d*, lo cual hace decrecer la probabilidad de que `Stroke` sea la causa, *i. e.,* $p(s \mid d, n) = 0,20 < p(s \mid d) = 0,25$. Por otra parte, la presencia de un "accidente cerebro vascular" hace que la "atrofia neuronal" sea menos probable, $p(n \mid d, s) = 0,26 < p(n \mid d) = 0,33$.

Además, podemos encontrar la explicación mas probable para un paciente con parálisis, si resolvemos $\arg\max_{\{A,N,S,D\}} p(A,N,S,D \mid p)$. Obtenemos ($a$,¬$n$,$s$,¬$d$), con probabilidad 0,25. Es decir, la parálisis se explica por: tener una "edad de +65 años", haber tenido un "accidente cerebro vascular", y no tener ni "atrofia neuronal" ni "demencia". Las otras posibles configuraciones son menos probables. Finalmente, en la abducción parcial buscaríamos la explicación (reducida) del conjunto de variables (A, S) dada la misma evidencia de parálisis. En otras palabras tratamos de resolver el problema $\arg\max_{\{A,S\}} p(A,S \mid p)$ y encontramos la configuración (a, s), con probabilidad 0,53. Esto nos dice que "parálisis" está (parcialmente) explicada por tener una "edad de +65 años" y haber sufrido un "accidente cerebro vascular".

3.2.1. Inferencia exacta

La *inferencia exacta* en redes bayesianas es un problema NP-difícil[2] (Cooper, 1990). En la práctica muchas de las inferencias exactas que se llevan a cabo no se relacionan con el peor

[2] La clase de problemas NP-difíciles (*NP-hard*) es un concepto fundamental en teoría de la complejidad computacional y describe problemas que son al menos tan difíciles como los problemas de la clase NP (*Nondeterministic Polynomial time*), problemas cuya solución puede ser verificada en tiempo polinómico por una máquina determinista.

caso y se puede dar una respuesta no muy costosa computacionalmente usando los algoritmos que se muestran en lo que sigue.

Existen varios algoritmos exactos de propagación de la evidencia en redes bayesianas. La *aproximación por fuerza bruta* usa la factorización proporcionada por la red bayesiana para obtener la JPD y posteriormente por marginalización da respuesta a las preguntas de inferencia, siendo sin embargo, incapaz de organizar eficientemente las operaciones necesarias para llevar a cabo dicha inferencia, aspecto en el que sí que se focaliza el *algoritmo de eliminación de variables*. Este algoritmo se ilustra en el ejemplo que aparece, a continuación. El *algoritmo de paso de mensajes* (Lauritzen y Spiegelhalter, 1988) opera enviando mensajes entre los nodos de la red bayesiana. Un nodo actúa como un procesador autónomo que colecta los mensajes que le llegan (factores) de sus nodos vecinos, efectúa ciertas operaciones (sumas y multiplicaciones) y envía un mensaje de salida a sus nodos vecinos. El proceso termina cuando el nodo raíz ha recibido mensajes de todos sus nodos adjuntos.

Ejemplo. Riesgo de demencia (algoritmo de eliminación de variables)

Supongamos que estamos interesados en la probabilidad de `Stroke` para un paciente que no sufre de demencia, $p(S \mid \neg d)$. Comenzamos con una lista $\mathcal{L}$ que contiene todas las distribuciones locales dadas en la red bayesiana instanciadas por $D = \neg d$: $\mathcal{L} = \{f_A(A), f_N(N,A), f_S(S,A), f_P(P,S), f_D(\neg d, S, N)\}$.

Consideremos el orden de eliminación *P-A-N*. En primer lugar eliminamos P computando $\sum_P f_P(P,S) \equiv 1$. La nueva lista no contiene $f_P(P,S)$. En segundo lugar eliminamos A calculando $\sum_A f_A(A) f_N(N,A) f_S(S,A) = g_1(N,S)$, que nos proporciona $g_1(n,s) = 0,0114, g_1(\neg n, s) = 0,0686, g_1(n, \neg s) = 0,1086$ y $g_1(n, \neg s) = 0,8114$. Ahora la lista es $\mathcal{L} = \{g_1(N,S), f_D(\neg d, S, N)\}$. Finalmente, eliminamos N calculando $\sum_N f_D(\neg d, S, N) g_1(N,S) = g_2(S)$, con $g_2(s) = 0,0382$ y $g_2(\neg s) = 0,7954$. Ahora $\mathcal{L} = \{g_2(S)\}$ y $g_2(S)$ es $p(S, \neg d)$. Para obtener $p(S \mid \neg d)$, una normalización es necesaria:

$$p(s \mid \neg d) = \frac{0,0382}{0,0382 + 0,7954} = 0.0458 \text{ y } p(\neg s \mid \neg d) = \frac{0,7954}{0,0382 + 0,7954} = 0,9542.$$

La expresión computada ha sido $p(S \mid \neg d) \propto \sum_N p(\neg d \mid N, S) \sum_A p(N \mid A) p(S \mid A) p(A) \sum_P p(P \mid S)$.

3.2.2. Inferencia aproximada

La inferencia aproximada se plantea como una alternativa a la exacta para redes bayesianas extremadamente complejas y de gran tamaño. Sigue siendo un problema NP-difícil (Dagum y Luby, 1993). La inferencia aproximada se basa en la simulación de la red bayesiana para generar un gran número de casos (instancias completas) de la JPD, y posteriormente estimar la probabilidad solicitada a partir de las frecuencias observadas en las muestras.

Consideremos que nuestro objetivo es estimar la probabilidad $p(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{e})$ para cualquier $\boldsymbol{Y} \subseteq \{X_1,...,X_n\}$. El *muestreo lógico probabilístico* (Henrion, 1988) parte de un orden ancestral (padres anteceden a los hijos) entre los nodos. Se muestrea un nodo X después de muestrear de todos sus padres, $\boldsymbol{Pa}(X)$, lo cual proporciona un valor fijo, $\boldsymbol{pa}(X)$. Después de haber muestreado $p(X|\boldsymbol{pa}(X))$ de todos los nodos, obtenemos una muestra de $p(\boldsymbol{X})$, la distribución de probabilidad conjunta. Este proceso se repite M veces. Podemos estimar $p(\boldsymbol{y} \mid \boldsymbol{e}) = \frac{p(\boldsymbol{y},\boldsymbol{e})}{p(\boldsymbol{e})}$ a partir de la estimación del numerador, como la fracción de muestras (del total de M) en las que hemos visto ($\boldsymbol{y}$, $\boldsymbol{e}$) y del denominador como la fracción de muestras (del total de M) donde hemos visto $\boldsymbol{e}$.

Otros métodos de inferencia aproximada incluyen el denominado *pesado por verosimilitud* (Shachter y Peot, 1989) y el *muestreo de Gibbs* (Pearl, 1987).

3.3. Aprendizaje a partir de datos

Los modelos de redes bayesianas pueden obtenerse a partir de la interacción con un experto en el dominio a modelar, de manera automática induciendo el modelo a partir de un algoritmo o como una combinación de ambos.

3.3.1. Descubrimiento de relaciones entre variables

R_i denota la cardinalidad de Ω_{Xi}, es decir el número de valores posibles de la variable X_i. Sea $q_i = |\Omega_{\mathbf{pa}(X_i)}|$ el número de combinaciones posibles, cada una denotada $\mathbf{pa}_i^j$, de los valores de los padres de X_i, es decir, $\Omega_{\mathbf{pa}(X_i)} = \left\{\mathbf{pa}_i^1,...,\mathbf{pa}_i^{q_i}\right\}$. Entonces, la CPT de X_i contiene los parámetros $\theta_{ijk} = p(X_i = k \mid \mathbf{Pa}(X_i) = \mathbf{pa}_i^j)$, la probabilidad condicional de que X_i tome su k-ésimo valor dado que sus padres toman su j-ésimo valor. Por lo tanto, la CPT de X_i requiere la estimación de los parámetros $\boldsymbol{\theta}_i$, un vector de $R_i q_i$ componentes. $\boldsymbol{\theta}=(\boldsymbol{\theta}_1,..., \boldsymbol{\theta}_n)$ que incluye todos los parámetros en la red bayesiana, es decir, $\theta_{ijk}, \forall i = 1,...,n, j = 1,...,q_i, k = 1,...,R_i$, siendo un vector con $\Sigma_{i=1}^{n} R_i q_i$ componentes.

Aprendizaje de parámetros. Una vez que se ha encontrado la estructura de la red bayesiana $\mathcal{G}$, los parámetros θ_{ijk} se estiman a partir del conjunto de datos $\mathcal{D}$. Sean N_{ij} el número de casos en $\mathcal{D}$ en los que se ha observado la configuración $\mathbf{Pa}(X_i) = \mathbf{pa}_i^j$, y N_{ijk} el número de casos en $\mathcal{D}$ donde se ha observado simultáneamente que $X_i = k$ y $\mathbf{Pa}(X_i) = \mathbf{pa}_i^j$ $(N_{ij} = \Sigma_{k=1}^{R_i} N_{ijk})$.

La *estimación máximo verosimil* busca los valores $\hat{\theta}^{ML}$ de los parámetros que maximizan la verosimilitud del conjunto de datos dado el modelo:

$$\hat{\boldsymbol{\theta}}^{ML} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{G}) = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D} \mid \mathcal{G}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{h=1}^{N} p(x^h \mid \mathcal{G}, \boldsymbol{\theta}) \qquad [3]$$

Usando las premisas de *independencia global de los parámetros* y de *independencia local de los parámetros* (Spiegelhalter y Lauritzen, 1990), la anterior expresión puede calcularse como $\mathcal{L}(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{G}) = \Pi_{i=1}^{n} \Pi_{j=1}^{q_i} \Pi_{k=1}^{R_i} \theta_{ijk}^{N_{ijk}}$. Por lo tanto, los estimadores máximo verosímiles de θ_{ijk} pueden estimarse mediante conteos de frecuencias en $\mathcal{D}: \hat{\theta}_{ijk}^{ML} = \frac{N_{ijk}}{N_{ij}}$, donde $N_{ij} = \Sigma_{k=1}^{R_i} N_{ijk}$.

En el *enfoque de estimación bayesiana* los parámetros $\boldsymbol{\theta}$ se modelan con una variable aleatoria con distribución de probabilidad $f(\boldsymbol{\theta} \mid \mathcal{G})$, codificando el conocimiento previo sobre los posibles valores de $\boldsymbol{\theta}$. La distribución *a posteriori* dados los datos $\mathcal{D}$ y el grafo $\mathcal{G}$ se calcula aplicando la regla de Bayes: $f(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{G}) \propto p(\mathcal{D} \mid \mathcal{G}, \boldsymbol{\theta}) f(\boldsymbol{\theta} \mid \mathcal{G})$. Esta distribución puede resumirse usando alguna medida de tendencia central, típicamente la media *a posteriori* $\hat{\boldsymbol{\theta}}^{Ba} = \int \boldsymbol{\theta} f(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{G}) d\boldsymbol{\theta}$.

La distribución de Dirichlet es una familia conjugada para distribuciones categóricas (como las que aparecen en las redes bayesianas discretas). Por lo tanto, asumiendo una distribución de Dirichlet para la distribución *a priori* (Spiegelhalter y Lauritzen, 1990), la *a posteriori* también seguirá una distribución de Dirichlet. Así, tenemos que para la distribución *a posteriori* de los parámetros $\boldsymbol{\theta}_{ij} = (\theta_{ij}, ..., \theta_{ijRi})$ en la variable condicionada $Xi \mid \mathbf{pa}_i^j$, si $(\boldsymbol{\theta}_{ij} \mid \mathcal{G}) \sim Dir(\alpha_{ij1}, ..., \alpha_{ijR_i})$, entonces $(\boldsymbol{\theta}_{ij} \mid \mathcal{D}, \mathcal{G}) \sim Dir(\alpha_{ij1} + N_{ij1}, ..., \alpha_{ijRi} + N_{ijR_i})$, y por lo tanto la media *a posteriori* de cada $\boldsymbol{\theta}_{ijk} (k = 1, ..., R_i)$ da lugar a la estimación bayesiana: $\hat{\theta}_{ijk}^{Ba} = \frac{N_{ijk} + \alpha_{ijk}}{N_{ij} + \alpha_{ij}}$, donde $\alpha_{ij} = \Sigma_{k'=1}^{R_i} \alpha_{ijk'}$. Los hiperparámetros α_{ijk} pueden interpretarse como conteos imaginarios basados en nuestra experiencia previa. La interpretación de la distribución de Dirichlet sería que antes de obtener $\mathcal{D}$ hemos observamos virtualmente una muestra de tamaño α_{ij}, donde $X_i = k \mid \mathbf{pa}_i^j$, fue visto α_{ijk} veces.

Aprendizaje de estructuras. Existen dos maneras diferentes de abordar el aprendizaje de la estructura: (a) Testando las independencias condicionales de las variables por medio de tests de hipótesis; (b) Puntuando la bondad de cada estructura candidata la cual es buscada heurísticamente.

(a). Los métodos basados en restricciones testan estadísticamente las independencias condicionales entre tripletas de variables a partir de los datos, para a continuación encontrar un DAG que representa un gran porcentaje de (y siempre que sea posible todas) las restricciones de independencia condicional identificadas por los tests de hipótesis.

El método más representativo es el *algoritmo PC* (Spirtes y Glymour, 1991). PC comienza con un grafo completo no dirigido y consta de tres etapas. La etapa 1 produce las adyacencias en el grafo (el esqueleto de la estructura aprendida) mediante la eliminación de aristas a través de pruebas de hipótesis. La etapa 2 identifica colisionadores que sirven para orientar algunas aristas. La etapa 3 tiene como objetivo orientar el resto (dentro de lo posible) de las aristas.

La etapa 1 actúa de manera iterativa. La primera iteración de PC verifica la independencia marginal para todas las parejas de nodos (X_i, X_j), es decir, sin condicionar en otras variables $(\boldsymbol{S} = \emptyset)$. Si son independientes, se elimina la conexión y el conjunto vacío se guarda como conjunto de separación en $\boldsymbol{S}_{ij} = \boldsymbol{S}_{ji}$. En la siguiente iteración, el tamaño t de $\boldsymbol{S}$ aumenta en una unidad. Así, el algoritmo verifica si para cada pareja ordenada (X_i, X_j) que aún están adyacentes en $\mathcal{G}$, X_i y X_j son c.i. dado $\boldsymbol{S}$, para cualquier $\boldsymbol{S}$ de tamaño t=1 de $\text{Adj}_i \backslash \{X_j\}$. Se consideran todos los posibles conjuntos $\boldsymbol{S}$ según el orden. La arista X_i–X_j se elimina si y solo si se encuentra un conjunto $\boldsymbol{S}$ que haga a $\boldsymbol{X}_i$ y $\boldsymbol{X}_j$ c.i. Este $\boldsymbol{S}$ se guarda como conjunto de separación en $\boldsymbol{S}_{ij} = \boldsymbol{S}_{ji}$. Si se han considerado todas las parejas ordenadas de nodos adyacentes para testar la independencia condicional dados todos los subconjuntos $\boldsymbol{S}$ de tamaño t de sus conjuntos de adyacencia, el algoritmo aumenta t en una unidad. Por lo tanto, el proceso se repite para conjuntos de condicionamiento $\boldsymbol{S}$ con dos nodos, luego tres nodos, etc., afinando el grafo hasta que se agoten todas las posibilidades (hasta n–2 variables), y no queden más conjuntos de adyacencia por verificar. El número de pruebas de independencia condicional para una pareja dada (X_i, X_j) se va reduciendo en cada iteración. A medida que PC avanza, la fiabilidad de las pruebas estadísticas de independencia condicional disminuye porque los conjuntos $\boldsymbol{S}$ incluyen un número creciente de variables (reduciendo así el tamaño de las muestras sobre las que se aplican las pruebas).

Basado en el esqueleto y los conjuntos de separación, la etapa 2 identifica colisionadores para orientar algunas aristas. En las conexiones divergentes ($X_i \leftarrow X_j \rightarrow X_k$) y en serie ($X_i \rightarrow X_j \rightarrow X_k$) entre tres nodos, X_i y X_k son independientes dado X_j. Para una conexión convergente ($X_i \rightarrow X_j \leftarrow X_k$), X_i y X_k son dependientes dado X_j. Por lo tanto, si consideramos X_i–X_j–X_k (donde X_i y X_k no son adyacentes) en $\mathcal{G}$, primero identificamos un colisionador en X_j probando si X_i y X_k son dependientes dado X_j. Si lo son, dibujamos $X_i \rightarrow X_j \leftarrow X_k$. Esto se puede inferir a partir de las pruebas ya realizadas para eliminar la arista entre X_i y X_k y los conjuntos de separación guardados. X_j es un colisionador si y solo si $X_j \notin S_{ik}$, es decir, si X_j no se incluyó en el conjunto de condicionamiento que hizo a X_i y X_k c.i.

Finalmente, la etapa 3 orienta tantas aristas no dirigidas restantes como sea posible. Este proceso se realiza por consistencia con las aristas ya orientadas y de modo que no se formen nuevos colisionadores ni ciclos. Se lleva a cabo de manera recursiva hasta que no se puedan orientar más aristas. Las siguientes tres reglas se aplican repetidamente:

- Orientar X_j–X_k como $X_j \rightarrow X_k$ siempre que haya una arista dirigida $X_i \rightarrow X_j$ tal que X_i y X_k no sean adyacentes (de lo contrario se crearía un nuevo colisionador);
- Orientar X_i–X_j como $X_i \rightarrow X_j$ siempre que se forme una cadena $X_i \rightarrow X_k \rightarrow X_j$ (de lo contrario se crearía un ciclo dirigido);
- Orientar X_i–X_j como $X_i \rightarrow X_j$ siempre que haya dos cadenas X_i-$X_k \rightarrow X_j$ y X_i–$X_h \rightarrow X_j$ tales que X_k y X_h no sean adyacentes (de lo contrario se crearía un nuevo colisionador o un ciclo dirigido).

(b). Los *métodos basados en puntuación y búsqueda* utilizan una función de puntuación relativa a los datos para medir la calidad de cada estructura de red bayesiana candidata. El objetivo es encontrar una estructura de red que maximice la función de puntuación. Los métodos generalmente comienzan a partir de una estructura inicial (generada aleatoriamente o a partir del conocimiento ya existente sobre el dominio). Un método de búsqueda, habitualmente basado en un heurístico, es el encargado de realizar movimientos inteligentes en el espacio de posibles estructuras de red. En esta sección nos centraremos en el espacio de los DAG.

La cardinalidad del *espacio de los DAG* viene dada por la fórmula recursiva de Robinson (Robinson, 1977) y es superexponencial en el número de nodos. En concreto, el número $f(n)$ de posibles DAG que contienen n nodos está dado por la recurrencia $f(n)=\Sigma_{i=1}^{n}(-1)^{i+1}\binom{n}{i}2^{i(n-i)}f(n-i)$, para $n>2$, la cual se inicializa con $f(0)=f(1)=1$. La tarea de encontrar una estructura de red que optimice la puntuación es un problema de optimización combinatoria y se sabe que es NP-difícil (Chickering, 1996). Esto motiva el uso de heurísticos para llevar a cabo dicha búsqueda.

La calidad del ajuste de un DAG $\mathcal{G}$ al conjunto de datos $\mathcal{D}$ se mide por una función $Q(\mathcal{D},\mathcal{G})$, de tal manera que cuanto mejor sea el ajuste, mayor será la puntuación. El problema del aprendizaje de la estructura es encontrar $\arg\max_{\mathcal{G}} Q(\mathcal{D},\mathcal{G})$. Un criterio simple es la *log-verosimilitud estimada de los datos dada la red bayesiana*:

$$\log \mathcal{L}(\hat{\theta} \mid \mathcal{D},\mathcal{G}) = \log p(\mathcal{D} \mid \mathcal{G},\hat{\theta}) = \log \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{R_i}\hat{\theta}_{ijk}^{N_{ijk}} = \Sigma_{i=1}^{n}\Sigma_{j=1}^{q_i}\Sigma_{k=1}^{R_i}N_{ijk}\log\hat{\theta}_{ijk}, \qquad [4]$$

donde $\hat{\theta}_{ijk}=\hat{\theta}_{ijk}^{ML}=\dfrac{N_{ijk}}{N_{ij}}$ es el estimador máximo verosimil de θ_{ijk}.

El inconveniente de usar la verosimilitud como puntuación es que aumenta de manera monótona con la complejidad del modelo (sobreajuste estructural). Por lo tanto, la estructura que maximiza la verosimilitud coincide con el grafo completo. Para evitar este problema se ha propuesto una familia de puntuaciones de *log-verosimilitud penalizada* que penalizan la complejidad de la red. Su expresión general es $Q^{pen}(\mathcal{D},\mathcal{G})=\Sigma_{i=1}^{n}\Sigma_{j=1}^{q_i}\Sigma_{k=1}^{R_i}N_{ijk}\log\dfrac{N_{ijk}}{N_{ij}}-dim(\mathcal{G})pen(N)$, donde $dim(\mathcal{G})=\Sigma_{i=1}^{n}(R_i-1)q_i$ denota la dimensión del modelo (número de parámetros necesarios en la red bayesiana), y $pen(N)$ es una función de penalización no negativa. Si $pen(N)=1$, tenemos el *criterio de información de Akaike* (AIC) (Akaike, 1974) y si $pen(N)=\dfrac{1}{2}\log N$, el *criterio de información bayesiano* (BIC) (Schwarz, 1978).

El objetivo de un enfoque bayesiano es encontrar la estructura con la máxima probabilidad *a posteriori* dado el conjunto de datos, es decir, $\arg\max_{\mathcal{G}} p(\mathcal{G}|\mathcal{D})$. Usando la fórmula de Bayes, $p(\mathcal{G} \mid \mathcal{D}) \propto p(\mathcal{D},\mathcal{G}) = p(\mathcal{D} \mid \mathcal{G})p(\mathcal{G})$. El segundo factor, $p(\mathcal{G})$, denota la distribución *a priori* sobre las estructuras. El primer factor, $p(\mathcal{D}|\mathcal{G})$, es la *verosimilitud marginal* de los datos, definida como $p(\mathcal{D} \mid \mathcal{G}) = \int p(\mathcal{D} \mid \mathcal{G},\boldsymbol{\theta}) = f(\boldsymbol{\theta} \mid \mathcal{G})d\boldsymbol{\theta}$, donde $p(\mathcal{D}|\mathcal{G},\boldsymbol{\theta})$ es la verosimilitud de los datos dada la red bayesiana (estructura $\mathcal{G}$ y parámetros $\boldsymbol{\theta}$), y $f(\boldsymbol{\theta}|\mathcal{G})$ es la distribución *a priori* sobre los parámetros.

Supongamos que el vector de parámetros de la red bayesiana tiene una distribución de Dirichlet $(\boldsymbol{\theta}_{ij} \mid \mathcal{G}) \sim Dir(\alpha_{ij1},...,\alpha_{ijR_i})$. En tal caso (Cooper y Herskovits, 1992) obtuvieron una expresión cerrada para la verosimilitud marginal que al tomar logaritmos se transforma en la puntuación *bayesiana de Dirichlet*

$$Q^{BD}(\mathcal{D},\mathcal{G}) = \log p(\mathcal{G}) + \Sigma_{i=1}^{n}\Sigma_{j=1}^{q_i}\left(\log\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} + \Sigma_{k=1}^{R_i}\log\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\right).$$

Esta puntuación tiene poco interés práctico ya que requiere la especificación de todos los hiperparámetros α_{ijk} para todos los *i, j, k*. Por esta razón se han propuesto algunos casos particulares más simples. La *puntuación K2* (Cooper y Herskovits, 1992) utiliza la asignación no informativa $\alpha_{ijk} = 1$, para todos los *i, j, k*, obteniéndose

$$Q^{K2}(\mathcal{D},\mathcal{G}) = \log p(\mathcal{G}) + \Sigma_{i=1}^{n}\Sigma_{j=1}^{q_i}\left(\log\frac{(R_i-1)!}{(N_{ij}+R_i-1)!} + \Sigma_{k=1}^{R_i}\log N_{ijk}!\right).$$

En cuanto a procedimientos de búsqueda, el más conocido es el *algoritmo K2* (Cooper y Herskovits, 1992). K2 utiliza un método de búsqueda voraz y la *puntuación K2*. Comienza con una estructura sin padres, un orden sobre los nodos y un límite superior sobre el número máximo de padres que se permite tener a cualquier nodo. El algoritmo agrega de manera incremental, desde el conjunto de nodos que preceden a cada nodo X_i, y siguiendo el orden establecido, el padre cuya adición incremente más la función $g(X_i,\mathbf{Pa}(X_i)) = \prod_{j=1}^{q_i}\frac{(R_i-1)!}{(N_{ij}+R_i-1)!}\prod_{k=1}^{R_i}N_{ijk}!$. Cuando la adición de ningún padre individual no mejora la puntuación, no se añaden más padres al nodo X_i, y pasamos al siguiente nodo en el orden.

Otros algoritmos que se han usado para la búsqueda de la mejor estructura de red bayesiana han sido: búsqueda voraz (Buntine, 1991), enfriamiento estocástico (Heckerman *et al.,* 1995) métodos MCMC (Giudici y Green, 1999), algoritmos genéticos (Larrañaga *et al.,* 1996), y algoritmos de estimación de distribuciones (Larrañaga *et al.,* 2000).

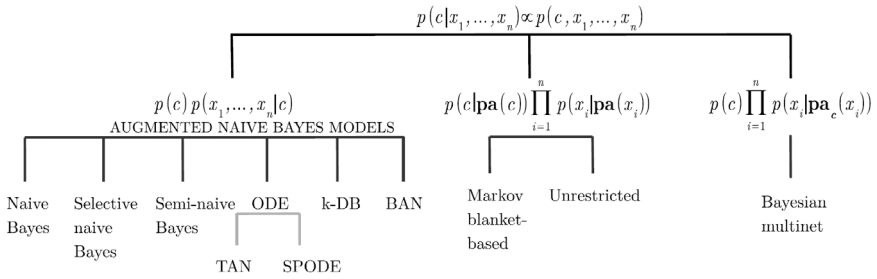
3.3.2. Clasificadores bayesianos

Los clasificadores bayesianos con variables predictoras discretas aproximan $p(\boldsymbol{x}, c)$ con la factorización proporcionada por la red bayesiana para las variables aleatorias $\boldsymbol{X_1},...,\boldsymbol{X_n}$, C. Es decir $p(\boldsymbol{x},c) = p(c \mid \mathbf{pa}(c))\prod_{i=1}^{n}p(x_i \mid \mathbf{pa}(x_i))$. Con una función de pérdida 0-1, la regla de decisión de Bayes consiste en encontrar c* tal que $c^* = \arg\max_c p(c \mid \boldsymbol{x}) = \arg\max_c p(\boldsymbol{x} \mid c)$.

La figura 4 muestra un esquema de distintos clasificadores bayesianos en función de si la variable C tenga o no padres y de si se permiten independencias condicionales dependientes del contexto.

Figura 4.

Clasificaciones bayesianos discretos en base a la factorización de $p(x,c)$



Fuente: Elaboración propia.

Para el caso en que $\mathbf{Pa}(C) = \emptyset$, el problema consiste en máx$_c$ $p(c)p(\boldsymbol{x}|c)$. En tal caso, los diferentes clasificadores bayesianos se agrupan bajo el nombre de *modelos de naive Bayes aumentado* donde se recogen modelos clasificatorios que van aumentando en complejidad. Si $\mathbf{Pa}(C) \neq \emptyset$, $p(\boldsymbol{x},c)$ se puede factorizar de diversas maneras, bien por medio del *manto de Markov* de C o usando *clasificadores basados en redes bayesianas no restringidas*. Finalmente, se pueden considerar relaciones de independencia condicional específicas para cada valor de c, lo cual da origen a *clasificadores bayesianos basados en multirredes*.

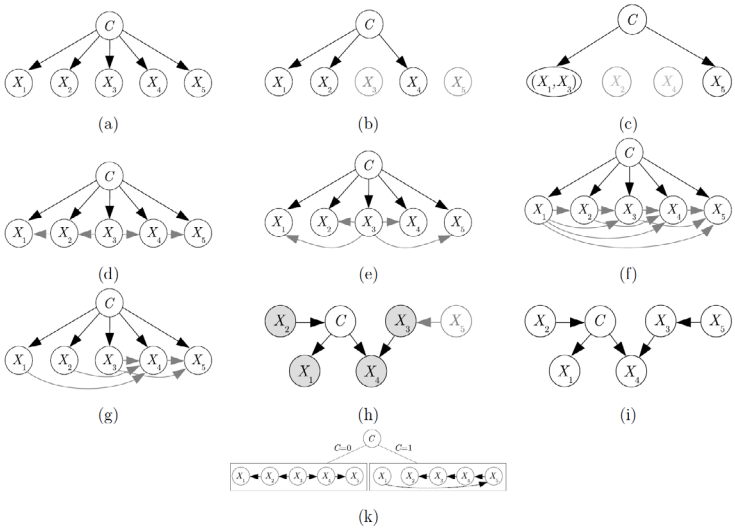
La figura 5 muestra las estructuras de los diferentes clasificadores bayesianos que se presentarán a continuación. *Naive Bayes* (Maron y Kuhns, 1960) es el clasificador bayesiano más simple. Las variables predictoras se asumen que son condicionalmente independientes dada la clase. Esta premisa, si bien está muy alejada de lo que realmente ocurre en un problema real, resulta ser muy útil cuando n es grande y/o N es pequeño. La premisa conduce a que $p(c \mid \boldsymbol{x}) \propto p(c)\prod_{i=1}^{n} p(x_i \mid c)$ (Minsky, 1961) demostró que la frontera de decisión asociada a un clasificador naive Bayes es un hiperplano.

Las prestaciones en cuanto a resultados clasificatorios del naive Bayes mejoran si tan sólo se consideran aquellas variables predictoras que son relevantes y no redundantes. Este proceso de selección de variables conduce al modelo *naive Bayes selectivo*, donde $p(c \mid \boldsymbol{x}) \propto p(c \mid \boldsymbol{x}_F) = p(c)\prod_{i\in F} p(x_i \mid c)$, y $\boldsymbol{X}_F$ denota la proyección de $\boldsymbol{X}$ en el subconjunto de variables seleccionadas $F \subseteq \{1, 2, ..., n\}$. La selección de variables puede llevarse a cabo por medio de una aproximación por filtrado (Pazzani y Billsus, 1997), o a partir de una aproximación basada en envoltura (Inza *et al.,* 2000).

El modelo *semi naive Bayes* relaja la asunción de independencia condicional introduciendo nuevas variables obtenidas como el producto cartesiano de dos o más variables predictoras originales. Estas nuevas variables predictoras siguen siendo condicionalmente independientes dado el valor de la clase. Se tiene $p(c \mid \boldsymbol{x}) \propto p(c)\prod_{j=1}^{K} p(x_{S_j} \mid c)$, donde $S_j \subseteq \{1,2,\ldots,n\}$ denota los índices en la j-ésima variable (original o producto cartesiano), $j = 1, ..., K$, $S_j \cap S_l = \emptyset$, para $j \neq l$. El algoritmo *de selección secuencial hacia adelante y unión*

Figura 5.

Estructuras de clasificación bayesianos



Notas: (a) Naive Bayes. (b) Selective naive Bayes. (c) Seminaive Bayes. (d) Naive Bayes aumentado a árbol. (e) Clasificador basado en superpadres con dependencias de orden 1. (f) Clasificador bayesiano *k*-dependiente. (g) Naive Bayes aumentado a red bayesiana. (h) Clasificador bayesiano basado en el manto de Markov. (i) Clasificador bayesiano no restringido. (k) Clasificador bayesiano basado en multirredes. Fuente: Elaboración propia.

(Pazzani, 1996) se guía por la precisión en la clasificación, comenzando con una estructura vacía. A continuación, el algoritmo considera la mejor opción entre (a) agregar una variable no utilizada por el clasificador actual como condicionalmente independiente de las variables (originales o productos cartesianos) utilizadas en el clasificador, y (b) unir una variable no utilizada por el clasificador actual con cada variable (original o producto cartesiano) presente en el clasificador. La mejor de estas dos opciones se aplica de manera iterativa hasta que el modelo deje de mejorar en sus prestaciones clasificatorias.

El subgrafo formado por las variables predictoras de un modelo *naive Bayes aumentado con árbol* (TAN) (Friedman *et al.,* 1997) es un árbol. Por lo tanto, todas las variables predictoras tienen exactamente un padre, excepto la variable raíz del árbol, que no tiene padres. El algoritmo para aprender una estructura TAN calcula la información mutua de cualquier par de variables predictoras condicionada a C, $I(X_i, X_j|C)$. A continuación, las aristas de un grafo no dirigido completo con nodos X_1, ..., X_n se anotan con los números de información mutua condicional calculados previamente. El algoritmo de Kruskal (Kruskal, 1956) se usa para encontrar las $n-1$ aristas del grafo, de tal manera que formen un árbol y cuya suma de pesos sea máxima. El árbol no dirigido se convierte entonces en un árbol dirigido seleccionando una variable al azar como nodo raíz y reemplazando las aristas por arcos. Final-

mente, se superpone una estructura naive Bayes para formar la estructura TAN. Se tiene: $p(c \mid \boldsymbol{x}) \propto p(c)p(x_r \mid c)\prod_{i=1,i\neq r}^{n} p(x_i \mid c, x_{j(i)})$, donde X_r denota el nodo raíz y $\{X_{j(i)}\} = \mathbf{Pa}(X_i) \setminus C$, para cualquier $i \neq r$.

Los *clasificadores bayesianos basados en superpadres con dependencias de orden* 1 (SPODEs) (Keogh y Pazzani, 2002) son modelos en los que todas las variables predictoras dependen de una única variable predictora (la misma para todas) llamada superpadre, además de la clase. En tal caso, $p(c \mid \boldsymbol{x}) \propto p(c)p(x_{sp} \mid c)\prod_{i=1,i\neq sp}^{n} p(x_i \mid c, x_{sp})$, donde X_{sp} denota el nodo superpadre. La gran ventaja de esta aproximación es que no necesita de un algoritmo de aprendizaje para fijar la estructura.

El *clasificador bayesiano con k dependencias* (*k*-DB) (Sahami, 1996) permite que cada variable predictora tenga un máximo de *k* variables padre, además de la variable clase. Se tiene $p(c \mid \boldsymbol{x}) \propto p(c)\prod_{i=1}^{n} p(x_i \mid c, x_{i_1}, ..., x_{i_k})$ donde X_{i1}, ...,X_{ik} son los padres de X_i en la estructura. El orden de inclusión de las variables predictoras X_i en el modelo se basa en I(X_i, C), empezando con la más alta. Una vez que X_i entra en el modelo, sus padres se seleccionan eligiendo las *k* variables X_j en el modelo con los valores más altos de I(X_i, $X_j|C$).

El clasificador *naive Bayes aumentado con una red bayesiana* (BAN), tiene cualquier estructura de red bayesiana como el subgrafo de variables predictoras (sección 3.3.1). Se tiene $p(c \mid \boldsymbol{x}) \propto p(c)\prod_{i=1}^{n} p(x_i \mid \mathbf{pa}(x_i))$. Ezawa y Norton (1996) primero, calculan I(X_i, C) para las *n* variables predictoras y luego seleccionan el número mínimo de variables predictoras *k* que satisfacen $\Sigma_{j=1}^{k}\mathrm{I}(X_j, C) \geq t_{CX}\Sigma_{j=1}^{n}\mathrm{I}(X_j, C)$, donde $0 < t_{CX} < 1$. En segundo lugar, se calcula I(X_i, $X_j|C$) para todos los pares de las variables seleccionadas. Se seleccionan las aristas correspondientes a los valores más altos hasta que se supera un porcentaje t_{XX} de la información mutua condicional total $\Sigma_{i<j}^{k}\mathrm{I}(X_i, X_j \mid C)$. La direccionalidad de las aristas se basa en la ordenación de las variables predictoras obtenida en el primer paso.

El manto de Markov de C está formado por aquellas variables (padres, hijos y padres de los hijos de C) cuyo conocimiento es lo único que afecta a la distribución de probabilidad de la variable clase. Podemos inducir un *clasificador bayesiano basado en el manto de Markov* (Koller y Sahami, 1996), partiendo de todas las variables predictoras, y en cada paso eliminando del manto de Markov considerado en esa iteración, aquella variable que menos información aporta con respecto de C. En concreto, proponen eliminar la variable que permite mantener $p(C|\boldsymbol{MB}^{(t)}(C))$ lo más cercana posible a $p(C|\boldsymbol{X})$.

Los clasificadores bayesianos generales no restringidos no consideran a C como una variable especial en el proceso de inducción. Podemos usar cualquier algoritmo de la sección 3.3.1 para inducir el DAG con las n+1 variables. El correspondiente manto de Markov de C se usa con fines de clasificación.

Las *multirredes bayesianas* (Geiger y Heckerman, 1996) permiten codificar independencias condicionales asimétricas, que solo se cumplen para algunos, pero no para todos los valores de las variables involucradas. Para clasificación supervisada, consisten en varias redes

bayesianas (locales), cada una de las cuales está asociada con un subconjunto del dominio de una variable *C*. Así, condicionado a cada c, las variables predictoras pueden formar diferentes redes locales con diferentes estructuras. Por lo tanto, las relaciones entre las variables predictoras no tienen que ser las mismas para todos los valores de *c*. Se tiene $p(c \mid \boldsymbol{x}) \propto p(c)\prod_{i=1}^{n} p(x_i \mid \mathbf{pa}_c(x_i))$, donde $\mathbf{Pa}_c(X_i)$ es el conjunto de padres de X_i en la red bayesiana local asociado con $C = c$.

4. INTERPRETABILIDAD CON REDES BAYESIANAS

Veamos cómo las redes bayesianas verifican las tres condiciones impuestas en [6] para que un paradigma pueda ser considerado como interpretable (sección 2).

Por lo que respecta a la *simulabilidad*, un ser humano puede reproducir el proceso que lleva a cabo una red bayesiana para sugerir una respuesta, ya que todos los procedimientos tanto exactos como aproximados para llevar a cabo cualquier tipo de inferencia consisten en efectuar sumas y multiplicaciones. Obviamente si el número de nodos del DAG es elevado y/o la densidad de las conexiones es alta, el seguimiento por parte del humano de todas las operaciones necesarias para que el modelo proporcione una respuesta puede ser tedioso.

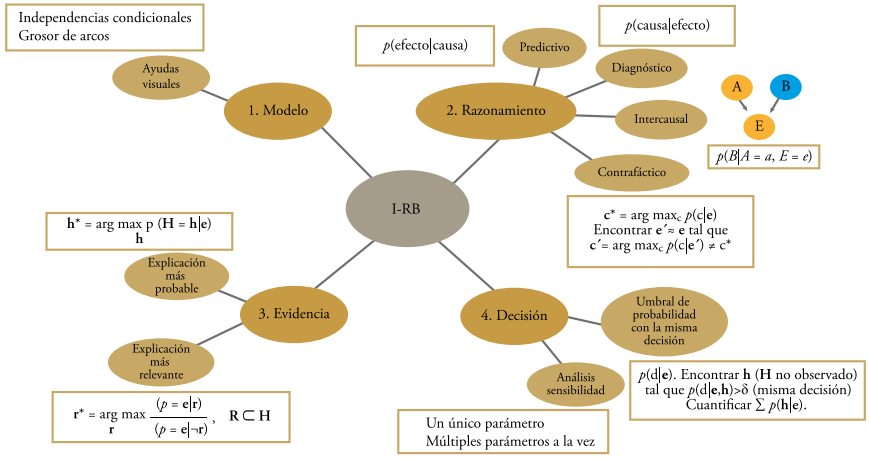
La *descomponibilidad* del modelo es una propiedad que es intrínseca a las redes bayesianas. La factorización de la distribución de probabilidad conjunta como producto de las distribuciones de probabilidad de cada variable dados sus padres en el DAG es la esencia de la modularidad que va a facilitar la interpretación del modelo global a partir de la interpretación de cada uno de los submodelos asociados a cada una de las variables en el DAG. Por otra parte, la descomponibilidad de las métricas habitualmente usadas para inducir el modelo a partir de datos (sección 3.3) hace que el aprendizaje de las estructuras de redes bayesianas pueda llevarse a cabo de manera modular.

Finalmente, la *transparencia algorítmica* tanto en el proceso de inducción del modelo, como de su posterior uso, son evidentes en las redes bayesianas. Tan sólo se requiere comprender conceptos básicos de Probabilidad, Estadística, y de Optimización Heurística, disciplinas en las que descansan ambos procesos.

La figura 6 muestra las posibilidades que proporciona una red bayesiana para poder interpretar un modelo desde distintos puntos de vista. Por lo que se refiere al *modelo* es factible testar en el grafo cualquier independencia condicional entre tripletas de variables a partir del criterio de u-separación. Adicionalmente podemos representar los arcos del DAG con grosores que sean proporcionales a la importancia de estos, cuantificando la importancia de cada arco a partir de su aportación a la métrica que el algoritmo de inducción está tratando de maximizar. Las redes bayesianas posibilitan distintos tipos de *razonamiento* que van desde un razonamiento predictivo, en el cual dadas unas causas nos preguntamos por las probabilidades de determinados efectos, o alternativamente por un razonamiento diagnóstico, en el cual se conocen las causas y se trata de calcular las probabilidades de los distintos efectos. El razonamiento intercausal (exclusivo de las redes bayesianas) asume la existencia de dos (o más causas) comunes para un mismo efecto. Si se conoce que una de las causas se ha pro-

Figura 6.

Interpretando una red bayesiana a partir del estudio del modelo, de diferentes tipos de razonamiento, de distintas opciones de propagación de la evidencia y de posibilidades de entender varias maneras de decidir



Fuente: Elaboración propia.

ducido, el modelo de red bayesiana va a reducir la probabilidad de que la otra causa haya ocurrido. En el denominado razonamiento contrafáctico se busca variar lo menos posible una determinada evidencia con el objetivo de que el valor de la variable clase predicha por el modelo cambie hacia una situación que resulte más satisfactoria (por ejemplo "vivo" en lugar de "muerto").

Dada una *evidencia*, dentro del denominado razonamiento abductivo, podemos preguntarnos por la instanciación del resto de las variables (inferencia abductiva global) que tenga asignada la máxima probabilidad tratando de encontrar la denominada explicación más probable. Si nuestro interés radicase exclusivamente en un subconjunto de las variables no instanciadas estaríamos tratando de resolver un problema de inferencia abductiva local. En la explicación más relevante, dada una evidencia se trata de encontrar la instanciación de un subconjunto de variables (a determinar por el algoritmo) no evidenciadas que tengan asociadas el máximo valor del factor de Bayes generalizado.

A nivel de *decisión*, dada una evidencia y una decisión adoptada al constatar que su probabilidad *a posteriori* dada la evidencia supera un determinado umbral preestablecido, la red bayesiana nos va a posibilitar la búsqueda de instanciaciones de las variables no observadas que actuando de condicionantes (junto a la evidencia) hagan que la probabilidad *a posteriori* de la decisión previa siga superando el umbral. La cuantificación de las probabilidades de todas las instanciaciones en las que se verifica esta condición nos permite determinar la robustez de la decisión adoptada. Las redes bayesianas también nos permiten llevar a cabo un análisis de sensibilidad bien de un único parámetro cada vez, o de varios parámetros al unísono.

5. INTERPRETANDO LA CLASIFICACIÓN MORFOLÓGICA DE INTERNEURONAS

El sistema nervioso humano es el sistema biológico más complejo. Para detectar y responder eficazmente a los cambios en el entorno, es capaz de aprender, tener autoconciencia y dar lugar al intelecto. Aunque muchos aspectos fundamentales de la estructura y función neuronal se entienden bien, quedan muchas preguntas sin respuesta. Responder a estas preguntas se está volviendo cada vez más urgente, principalmente debido al enorme costo social y económico de los trastornos del sistema nervioso. Los trastornos cerebrales, como la demencia, la depresión y la adicción, representan el 36 % de la carga total de enfermedades en los países de altos ingresos (Silberberg, 2015), con ocho millones de muertes atribuibles por año (Walker *et al.,* 2015).

Progresar hacia la comprensión del cerebro es un esfuerzo monumental. Con este fin, se han lanzado proyectos ambiciosos de neurociencia a nivel mundial en la última década. Estos incluyen el Proyecto Cerebro Humano (HBP) (Markram, 2012) de la Unión Europea, la iniciativa de Investigación del Cerebro a través del Avance de Neurotecnologías Innovadoras (BRAIN) (Insel *et al.,* 2013), el Instituto Allen para la Ciencia del Cerebro en los Estados Unidos, existiendo también iniciativas en Canadá, China, Japón, Corea e Israel. La mayoría son proyectos extremadamente grandes que requieren de esfuerzos interdisciplinarios, lo que refleja la complejidad de la tarea.

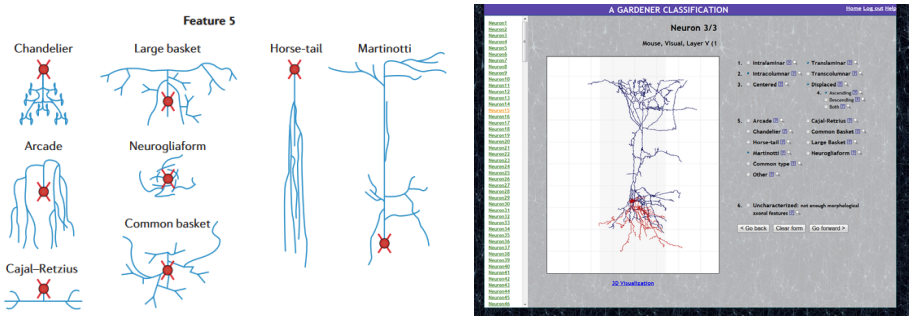
Las redes bayesianas han sido ampliamente aplicadas en la investigación en neurociencia. En Bielza y Larrañaga, 2014 se revisan muchas de estas aplicaciones. Un desafío clave en la neurociencia es la clasificación de interneuronas GABA-érgicas (Ascoli *et al.,* 2008). Estas neuronas constituyen alrededor del 20 %-30 % de las neuronas en la corteza cerebral y son el componente principal de los circuitos corticales inhibitorios, los cuales, a su vez, están asociados con trastornos como la epilepsia, el autismo y la esquizofrenia. Aunque la generación de datos a gran escala puede permitir el aprendizaje de una taxonomía sistemática a partir de datos en un futuro cercano mediante la agrupación de características moleculares, morfológicas y electrofisiológicas (Yuste *et al.,* 2020; Mihaljević *et al.,* 2018), los investigadores actualmente utilizan y se refieren a tipos morfológicos establecidos, como *chandelier, Martinotti, neurogliaforme* y *basket* (Markram *et al.,* 2004). Tener un modelo simple y preciso para clasificar automáticamente las interneuronas en estos tipos morfológicos podrían aportar conocimiento y ser útil para los profesionales.

Varios estudios han abordado el problema de la clasificación de interneuronas con métodos basados en redes bayesianas. Todos ellos se basan en un estudio fundamental sobre el consenso entre la comunidad científica en cuanto a la clasificación de interneuronas (DeFelipe *et al.,* 2013), en el cual 42 neurocientíficos expertos clasificaron 320 interneuronas de acuerdo con una taxonomía predefinida (figura 7). El estudio requería clasificar cinco variables morfológicas, además del tipo de inter- neurona, de ahí que por cada uno de los 42 neurocientíficos se obtuvo una tabla con 320 filas y 6 variables clase. Además, las morfologías de 240 de las 320 interneuronas fueron reconstruidas digitalmente, lo que permitió estudiar la clasificación supervisada de morfologías de interneuronas (Mihaljević *et al.,* 2014; Mihaljević *et al.,* 2015).

Figura 7.

Tipos de interneuronas y características morfológicas en el esquema de clasificación

(a) (b)



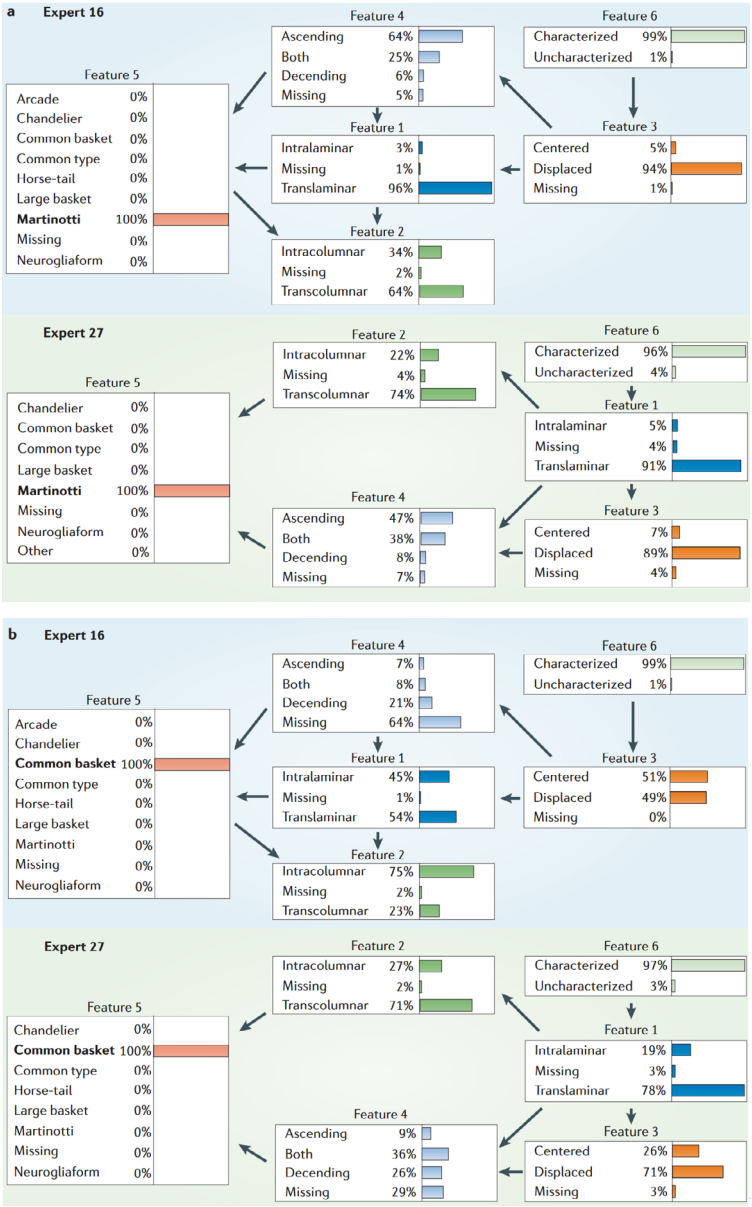
Nota: El esquema contempla diversos tipos de interneuronas: (a) Arcade, chandelier, horse-tail, Martinotti, common basket, Cajal-Retzius, large basket y neurogliaform (los tipos "común" y "otro" no se muestran en la gráfica). El tipo "otro" se utiliza cuando el neurocientífico considera que ninguno de los tipos restantes es adecuado. Además del tipo de interneurona, el esquema de clasificación contempla cinco variables morfológicas de alto nivel. Estas características, denominadas F1, F2, F3, F4 y F6 (F5 siendo el tipo de interneurona previamente discutido), tienen las siguientes categorías: (F1) intralaminar y translaminar; (F2) intracolumnar y transcolumnar; (F3) centrado y desplazado; (F4) ascendente, descendente y ambos; (F6) caracterizado y no caracterizado. La categoría no caracterizado de F6 significa que la reconstrucción de una célula no es lo suficientemente buena como para clasificarla de manera confiable. Cuando una célula se etiqueta como no caracterizada en la característica F6, el neurocientífico no puede anotarla según ninguna de las cinco características restantes, F1-F5. F4 solo se aplica a las células que se etiquetan como translaminares y desplazadas en F1 y F3, respectivamente. (b) La aplicación web utilizada para recopilar las elecciones de clasificación de los neurocientíficos para el conjunto de 320 interneuronas.
Fuente: DeFelipe *et al.* (2013).

En el estudio original (DeFelipe *et al.*, 2013), los autores utilizaron redes bayesianas para analizar las elecciones de clasificación de los neurocientíficos (figura 8). En particular, aprendieron una red bayesiana para cada neurocientífico con el fin de modelar su razonamiento en términos de las seis variables. Gracias a esta modelización se pudo estudiar, entre otras cosas, cómo cada experto relacionaba las características morfológicas, por ejemplo, si el axón era intra- o translaminar, con el tipo de interneurona. Estas redes permitieron identificar similitudes y diferencias en el razonamiento entre los expertos. Así, por ejemplo, vemos que los expertos 16 y 27 coinciden en muchas apreciaciones en cuanto al concepto de *Martinotti* (ver figura 8). Para ambos expertos se trata de un tipo de célula con una alta probabilidad de ser Translaminar, Caracterizada y Desplazada. Sin embargo, para el concepto de *common basket*, ambos expertos no se ponen de acuerdo. Mientras que para el experto 16 se trata de un tipo de interneurona que es mayormente Intracolumnar, para el experto 27 sería fundamentalmente Transcolumnar. También presentan mucha discrepancia en sus valoraciones de Desplazado y de Translaminar.

Un segundo objetivo fue predecir el tipo de interneurona y cuatro características morfológicas a partir de las morfologías reconstruidas digitalmente. En (Mihaljević *et al.*, 2015)

Figura 8.

Redes bayesianas para dos de los 42 neurocientíficos

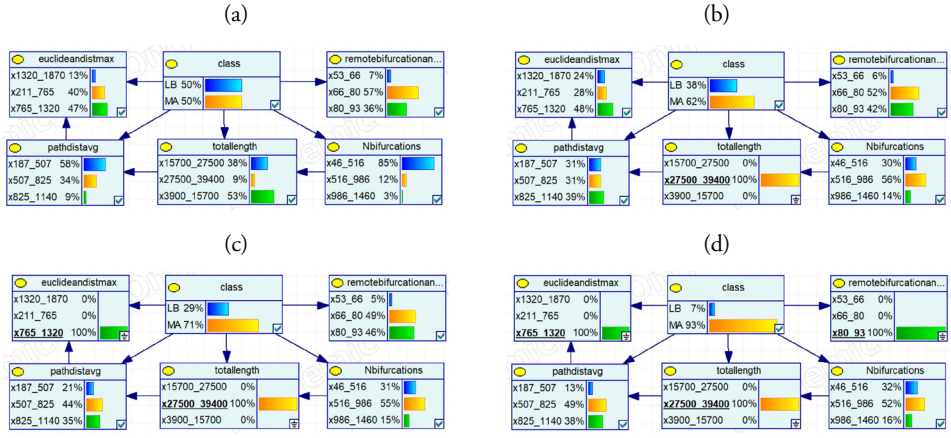


Nota: Los gráficos de barras muestran las probabilidades propagadas de las características restantes, condicionadas al tipo *Martinotti* (arriba) y al tipo *common basket* (abajo).
Fuente: DeFelipe *et al.* (2013).

se predijeron cada una de estas cinco variables por separado utilizando clasificadores bayesianos discretos. A diferencia de un escenario típico de clasificación supervisada, en este caso había hasta 42 etiquetas para cada instancia, proporcionadas por los diferentes neurocientíficos. El nivel de acuerdo entre los expertos variaba según las células: mientras que en 29 neuronas al menos 35 neurocientíficos coincidieron en su tipo de interneurona, hubo 67 células en las que no más de 15 expertos coincidieron en un tipo único. Los autores etiquetaron cada célula con la etiqueta más común entre las proporcionadas por los 42 neurocientíficos, pero repitieron la clasificación en diferentes subconjuntos de neuronas, formados al filtrar las células por debajo de un cierto umbral de fiabilidad de la etiqueta, definido como el número mínimo de neurocientíficos que coincidían en el tipo mayoritario. Los modelos alcanzaron una precisión de hasta el 89,52 % para el tipo de interneurona, siendo dicha precisión incluso mayor para las características morfológicas. La figura 9 ilustra cómo un clasificador naive Bayes aumentado a árbol puede utilizarse para interpretar el razonamiento detrás de la clasificación de una célula, proporcionando información sobre las características cuantitativas de dos tipos de interneuronas.

Figura 9.

Ilustración de la clasificación de una neurona con un clasificador bayesiano discreto naive Bayes aumentado a árbol, que discrimina entre los tipos de interneurona *Martinotti* (MA) y *large basket* (LB), aprendido a partir de 101 de las 240 reconstrucciones digitales utilizadas en DeFelipe *et al.* (2013)



Notas: El nodo de clase es el tipo de interneurona, mientras que los demás nodos corresponden a variables predictoras. Inicialmente, sin evidencia en las predictoras, una célula dada tiene la misma probabilidad de pertenecer a cualquiera de las dos clases (a). Si conocemos que la neurona tiene una longitud total media (en el rango de 27500–39400 µm) y establecemos eso como evidencia en la red (b), la probabilidad de que la neurona sea una célula Martinotti aumenta a 0,62. Observaciones posteriores sobre la máxima distancia euclidiana al soma (c) y el ángulo de bifurcación remota (d) de la neurona aumentan aún más esta probabilidad, hasta llegar a 0,93 en (d). Así, el neurocientífico puede entender la predicción del modelo y obtener información sobre las características cuantitativas de los dos tipos de interneuronas (Mihaljević *et al.*, 2021).
Fuente: DeFelipe *et al.* (2013).

6. CONCLUSIONES

Ante la necesidad cada vez más imperiosa de contar con sistemas inteligentes que sean interpretables por el ser humano (más allá de la explicabilidad a veces insuficiente y habitual en los paradigmas de caja negra) este capítulo ha mostrado la manera en la que las redes bayesianas dan respuesta a esta problemática. Por una parte, se ha mostrado cómo este paradigma verifica las tres condiciones necesarias para que el sistema sea considerado como interpretable: capacidad humana para simular el modelo, descomponibilidad del modelo en submodelos comprensibles, y transparencia algorítmica que posibilite al humano el entendimiento de los procesos de inducción y posterior utilización del modelo.

Adicionalmente hemos visto cómo las redes bayesianas posibilitan la interpretabilidad a distintos niveles. En primer lugar, a nivel de modelo con el testeo de independencias condicionales o con la asignación de pesos de importancia a cada uno de los arcos. En segundo lugar, posibilitando distintos tipos de razonamiento, desde el predictivo al diagnóstico, intercausal o contracfáctico. En tercer lugar, la evidencia introducida en la red bayesiana permite llevar a cabo distintos tipos de inferencia abductiva (global, parcial) como de explicación más probable. Finalmente, las decisiones tomadas con el modelo posibilitan dar soluciones al problema de determinar el umbral de decisión con la misma probabilidad, así como al análisis de sensibilidad del propio modelo.

En el capítulo nos hemos restringido a redes bayesianas discretas en dominios estáticos. Sin embargo, la extensión de los conceptos aquí mostrados a dominios continuos, con premisas de Gaussianidad o con estimaciones de densidades vía *kernels*, o incluso a dominios en los que los datos llegan temporalmente, a partir de las redes bayesianas dinámicas, las redes bayesianas en tiempo continuo, o los modelos ocultos de Markov, es factible existiendo un gran número de trabajos en la literatura. Finalmente, no sólo la clasificación supervisada puede beneficiarse de la interpretabilidad de las redes bayesianas, sino que problemas de regresión, *clustering*, aprendizaje por refuerzo u optimización heurística (vía algoritmos de estimación de distribuciones) pueden también ser interpretados bajo dicho prisma.

Referencias

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716–723.

Ascoli, G. A. *et al.* (2008). Petilla terminology: Nomenclature of features of GABAergic interneurons of the cerebral cortex. *Nature Reviews* Neuroscience, 9, 557–568.

Bielza, C., y Larrañaga, P. (2014). Bayesian networks in neuroscience: A survey. *Frontiers in Computational Neuroscience*, 8, 131.

Bielza, C., Li, G., y Larrañaga, P. (2011). Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning,* 52, 705-727.

Borchani, H., Varando, G., Bielza, C., y Larrañaga, P. (2015). A survey on multi-output regression. *WIREs Data Mining and Knowledge Discovery*, 5, 216–233.

Buntine, W. L. (1991). Theory refinement on Bayesian networks. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence* (52–609). Morgan Kaufmann.

Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. *Learning from Data: Artificial Intelligence and Statistics V,* 121–130. Springer.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence, 42*(23), 393–405.

Cooper, G. F., y Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347.

Dagum, P., y Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence, 6*(1), 141–153.

Ezawa, K. J., y Norton, S. W. (1996). Constructing Bayesian networks to predict uncollectible telecommunications accounts. *IEEE Expert, 11*(5), 45–51.

Friedman, N., Geiger, D., y Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, 131-163.

Geiger, D., y Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82, 45–74.

Giudici, P., y Green, P. J. (1999). Decomposable graphical Gaussian model determination. *Biometrika, 86*(4), 785–801.

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., y Yang, G. Z. (2019). XAI-Explainable artificial intelligence. *Scientific Robotics*, 4, 37.

Heckerman, D., Geiger, D., y Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20, 197–243.

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence,* 2, 149–163. Elsevier.

Insel, T. R., Landis, S. C., y Collins, F. S. (2013). The NIH BRAIN initiative. *Science,* 340, 687–688.

Inza, I., Larrañaga, P., Etxeberria, R., y Sierra, B. (2000). Feature subset selection by Bayesian network–based optimization. *Artificial Intelligence,* 123, 157-184.

Keogh, E. J., y Pazzani, M. J. (2002). Learning the structure of augmented Bayesian classifiers. *International Journal on Artificial Intelligence Tools, 11*(4), 587–601.

Koller, D., y Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques.* The MIT Press.

Koller, D., y Sahami, M. (1996). Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning,* 284–292.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society, 7*(1), 48-50.

Larrañaga, y P., Bielza, C. (2024). Estimation of distribution algorithms in machine learning: A survey. *IEEE Transactions on Evolutionary Computation, 28*(5).

Larrañaga, P., Etxeberria, R., Lozano, J. A., y Peña, J. M. (2000). Optimization in continuous domains by learning and simulation of Gaussian networks. In *Proceedings of the Genetic and Evolutionary Computation Conference* (201–204). Morgan Kaufmann.

Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R. H., y Kuijpers, C. M. H. (1996). Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(9), 912-926.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N., y Leimer, H. G. (1990). Independence properties of directed Markov fields. *Networks, 20*(5), 491–505.

Lauritzen, S. L., y Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological), 50*(2), 157–224.

Lipton, Z. C. (2019). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue, 16*(3), 31-57.

Lukas-Valentin, H., Heinrich, K., Wanner, J., y Janiesc, C. (2023). Stop ordering machine learning algorithms by their explainability! A user-centered investigation of performance and explainability. *International Journal of Information Management*, 69, 102538.

Markram, H. (2012). The Human Brain Project. *Scientific American*, 306, 50–55.

Markram, H., Toledo-Rodriguez, M., Wang, Y., Gupta, A., Silberberg, y G., Wu, C. (2004). Interneurons of the neocortical inhibitory system. *Nature Reviews Neuroscience* 5, 793–807.

Maron, M., y Kuhns, J. (1960). On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery,* 7, 216–244.

Mihaljević, B., Benavides-Piccione, R., Bielza, C., DeFelipe, J., y Larrañaga, P. (2015). Bayesian network classifiers for categorizing cortical GABAergic interneurons. *Neuroinformatics*, 13, 192–208.

Mihaljević, B., Bielza, C., Benavides-Piccione, R., DeFelipe, J., y Larrañaga, P. (2014). Multidimensional classification of GABAergic interneurons with Bayesian network-modeled label uncertainty. *Frontiers in Computational Neuroscience*, 8, 150.

Mihaljević, B., Bielza, C., y Larrañaga, P. (2021). Bayesian networks for interpretable machine learning and optimization. *Neurocomputing*, 456, 648-665.

Mihaljević, B., Larrañaga, P., Benavides-Piccione, R., Hill, S., DeFelipe, J., y Bielza, C. (2018). Towards a supervised classification of neocortical interneuron morphologies. *BMC Bioinformatics,* 19, 511.

Minsky, M. (1961). Steps toward artificial intelligence. *Transactions on Institute of Radio Engineers,* 49, 8–30.

Molnar, C. (2022). *Interpretable Machine Learning.* Lilu.com.

Muehlematter, U. J., Daniore, P., y Vokinger, K. N. (2020). Approval of artificial intelligence and machine learning-based medical devices in the USA and Europe (2015-20): A comparative analysis. *Lancet Digit Health,* 3, e195-203.

Nakka, R., Harursampath, D., y Ponnusami, S. A. (2023). A generalised deep learning-based surrogate model for homogenisation utilising material property encoding and physics-based bounds. *Scientific Reports*, 13, 9079.

Pazzani, M. (1996). Constructive induction of Cartesian product attributes. In *Proceedings of the Information, Statistics and Induction in Science Conference*, 66–77.

Pazzani, M., Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27, 313–331.

Pearl, J. (1987). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence, 32*(2), 245–257.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann.

Puerto-Santana, C., Larrañaga, P., y Bielza, C. (2022). Autoregressive asymmetric linear Gaussian hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(9), 4642-4658.

Robinson, R. (1977). Counting unlabeled acyclic digraphs. *Lecture Notes in Mathematics,* 622, 28-43. Springer.

Rodriguez-Sanchez, F., Rodriguez-Blazquez, C., Bielza, C., y Larrañaga, P. (2021). Identifying Parkinson´s disease subtypes with motor and non-motor symptoms via model-based multipartition clustering. *Scientific Reports, 11*(1), 1-10.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206-215.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 335–338.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6*(2), 461–464.

Shachter, R. C., y Peot, M. A. (1989). Simulation approaches to general probabilistic inference on belief networks. In *Proceedings of the 5th Annual Conference on Uncertainty in Artificial Intelligence* (221–234). Elsevier.

Shueli, G. (2010). To explain or to predict? *Statistical Science, 25*(3), 289-310.

Silberberg, D., Anand, N. P., Michels, y K., Kalaria, R. N. (2015). Brain and other nervous system disorders across the lifespan – global challenges and opportunities. *Nature*, 527, S151–S154.

Spiegelhalter, D. J., y Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20, 579–605.

Spirtes, P., y Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review, 90*(1), 62–72.

Valverde, G., Quesada, D., Larrañaga, P., y Bielza, C. (2023). Causal reinforcement learning based on Bayesian networks applied to industrial settings. *Engineering Applications of Artificial Intelligence*, 125, 106657.

Varando, G., Bielza, C., y Larrañaga, P. (2015). Decision boundary for discrete Bayesian network classifiers. *Journal of Machine Learning Research*, 16, 2725-2749.

Walker, E. R., McGee, R. E., y Druss, B. G. (2015). Mortality in mental disorders and global disease burden implications: A systematic review and meta-analysis. *JAMA Psychiatry*, 72, 334–341.

Yuste, R. *et al.* (2020). A community-based transcriptomics classification and nomenclature of neocortical cell types. *Nature Neuroscience*, 1–13.

CAPÍTULO VII

Redes bayesianas como modelos generativos: de los juegos a las finanzas*

Antonio Salmerón

Las redes bayesianas constituyen una herramienta versátil para la representación de conocimiento y cuantificación de la incertidumbre en inteligencia artificial. A través de su uso en varias aplicaciones, analizaremos su validez en el contexto de la IA generativa, y como modelos predictivos. Pondremos de manifiesto los problemas que surgen a la hora de aplicarlas en contextos de *big data* y propondremos algunas soluciones. Abordaremos aplicaciones de distinta naturaleza, que abarcan desde el ajedrez por computadora a la mejora genética vegetal y el seguimiento de la morosidad en créditos a clientes particulares.

Palabras clave: redes bayesianas, modelos generativos, aplicaciones.

* Ayuda PID2022-139293NB-C31 financiada por MCIN/AEI/10.13039/501100011033 y por "FEDER Una manera de hacer Europa".

1. INTRODUCCIÓN

La inteligencia artificial ha experimentado un desarrollo vertiginoso durante la última década, en gran medida debido al desarrollo del *deep learning* asociado a la disponibilidad de grandes volúmenes de datos. Un hito importante fue la irrupción del algoritmo AlphaZero (Silver *et al.,* 2017), capaz de obtener resultados a la altura de los mejores jugadores profesionales de go y ajedrez aprendiendo, sin intervención humana, en base a una enorme secuencia de partidas. Aunque el potencial de AlphaZero se mostró inicialmente en el contexto de esos juegos, con posterioridad se ha aplicado a problemas cruciales para la humanidad como la predicción de estructuras de proteínas (Jumper *et al.,* 2021).

Sin embargo, una cuestión importante es hasta qué punto los modelos que hay por debajo de tan llamativos resultados, son interpretables o entendibles por nosotros. Este hecho queda bien ilustrado por las palabras del jugador profesional de ajedrez Peter Nielsen, antiguo miembro del equipo de entrenadores de Magnus Carlsen, excampeón del mundo. Nielsen declaró, tras ver jugar al ajedrez a AlphaZero, lo siguiente:

> "Siempre me pregunté cómo sería si una especie superior aterrizara en la tierra y nos mostrara cómo juegan al ajedrez.
> Ahora lo sé".

Con esas palabras, se pone de manifiesto que, hasta un jugador profesional de ajedrez, es incapaz de entender por qué AlphaZero juega de una determinada forma o, lo que es lo mismo, por qué toma determinadas decisiones. Esta falta de entendimiento supone un problema a la hora de adoptar soluciones basadas en este tipo de modelos en aplicaciones críticas en ámbitos como pueden ser la salud o la economía, pues es necesario saber cómo se ha tomado una decisión para poder establecer los riesgos asociados a sus consecuencias. Es en este aspecto en el que los modelos basados en probabilidades ofrecen ventajas respecto a las soluciones basadas en redes neuronales profundas.

En la actualidad, ya nos encontramos con numerosas aplicaciones que de una forma o de otra, a veces en combinación con las redes neuronales, incorporan modelos basados en probabilidades. Entre ellas se encuentran las soluciones de vehículo autónomo, aplicaciones para la generación de imágenes, vídeo y audio, y muchas otras aplicaciones que tienen algunos aspectos en común:

- Operan en entornos con grandes volúmenes de datos disponibles.
- Sin embargo, los datos no cubren todos los posibles escenarios, por lo que siempre hay cierta incertidumbre asociada a las posibles predicciones de dichas aplicaciones.
- Usan un modelo probabilístico, en general aprendido a partir de datos, para manejar esa incertidumbre.

- Requieren procesos de inferencia para llevar a cabo *predicciones* y *análisis estructural* de los problemas que resuelven.

De forma natural, los modelos basados en probabilidades ofrecen una cuantificación de la incertidumbre bien fundamentada, y además permiten el tratamiento de datos faltantes pero, sobre todo, destacan por su interpretabilidad, dado que los humanos estamos habituados a trabajar con probabilidades. No obstante, para que estos modelos sean útiles en contextos de *big data*, es necesario dotarlos de la habilidad para operar en espacios de alta dimensionalidad (muchas variables) y darles soporte para realizar las tareas de inferencia y aprendizaje de manera eficiente.

2. REDES BAYESIANAS

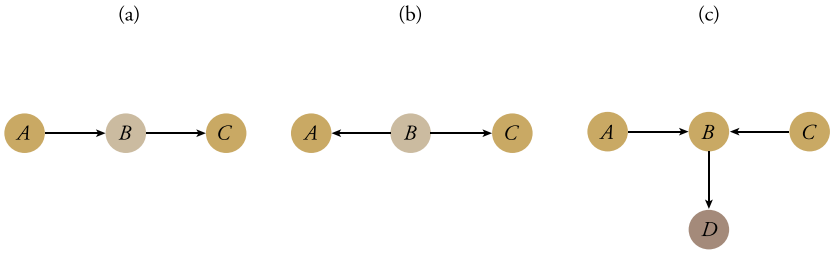
Las redes bayesianas (Jensen y Nielsen, 2007; Pearl, 1988) son modelos probabilísticos que tratan de cubrir los requisitos necesarios para ser operativas en entornos de *big data*. Formalmente, una red bayesiana es un grafo (un conjunto de vértices y aristas que los unen) dirigido (las aristas tienen dirección) acíclico (siguiendo la dirección de las aristas, no es posible volver al punto de partida desde ningún vértice) donde cada vértice es una variable aleatoria y las aristas determinan cuáles son las posibles dependencias entre las variables que forman el modelo.

Las redes bayesianas permiten una especificación estructurada de distribuciones de probabilidad de alta dimensionalidad en términos de factores de menor dimensión (conteniendo menos variables), dado que la distribución conjunta sobre todas las variables se puede representar como un producto de distribuciones condicionadas (una para cada variable dados sus padres en el grafo). Además, es posible definir procedimientos eficientes de inferencia y aprendizaje sacando partido de la estructura del grafo. Finalmente, la representación gráfica que ofrecen del problema que modelan es interpretable por los humanos. De hecho, en realidad cualquier grafo dirigido acíclico se puede construir en base a solo tres tipos básicos de conexiones entre sus vértices, que son las mostradas en la figura 1, de manera que conociendo la forma de interpretar esos tres tipos de conexiones, seremos capaces de entender la estructura completa de la red. Las conexiones en serie y divergentes indican que, si se conoce el valor de la variable intermedia (*B* en este caso), las variables de los extremos se vuelven independientes, mientras que si no se conoce el valor de *B, A* y *C* son independientes. Por contra, la presencia de una conexión convergente, indica que las variables de los extremos son independientes solo cuando no se conoce el valor de la variable intermedia ni de ninguno de sus descendientes.

Para ver si las redes bayesianas realmente constituyen una herramienta válida en el contexto de la IA y el *big data*, en primer lugar debemos aclarar lo que entendemos por aprendizaje automático o *machine learning* (ML), y para eso recurriremos a la definición original de Tom Mitchell (Mitchell, 1997):

Figura 1.

Tipos de conexiones en una red bayesiana: (a) en serie, (b) divergente y (c) convergente



Fuente: Elaboración propia.

> "Se dice que un programa de ordenador *P* aprende de la experiencia *E* con respecto a algún tipo de tarea *T* y alguna medida de rendimiento *R*, si su rendimiento en las tareas de *T*, medido en términos de *R*, mejora con la experiencia *E*".

Antes de comprobar si las redes bayesianas pueden ser consideradas como un modelo de ML, consideraremos un modelo estadístico muy sencillo, en concreto, un modelo lineal en el que tratamos de predecir el valor de una variable *Y* en función de otra variable, *X*. Esto conlleva la necesidad de estimar los parámetros *a* y *b* del modelo $\hat{y} = a + bx$ usando como medida de rendimiento (precisión) la raíz del error cuadrático medio:

$$rmse(y, \hat{y}) = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

con $y = \{y_1, ..., y_n\}$ los datos e $\hat{y} = \{\hat{y}_1, ..., \hat{y}_n\}$ los valores estimados por el modelo (predicciones).

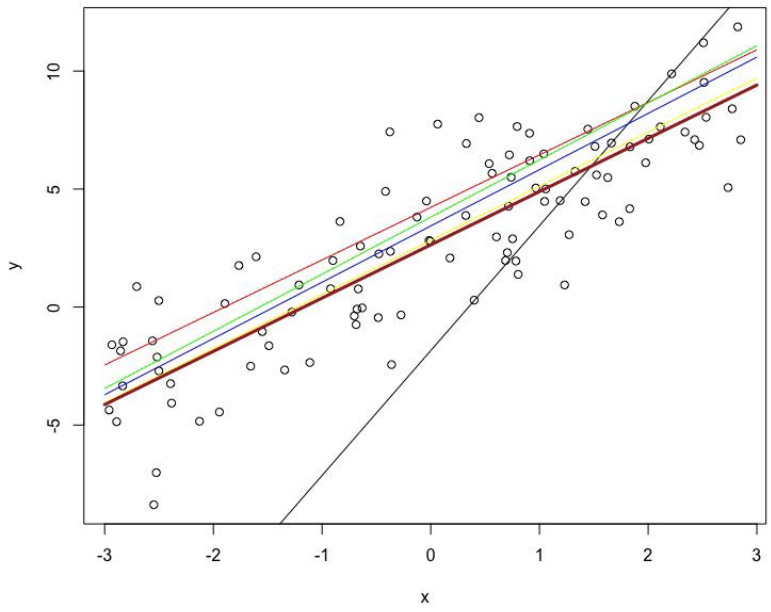
Consideremos un sencillo experimento consistente en generar un conjunto de 100 pares de valores (*X*,*Y*), correspondientes a la nube de puntos de la figura 2, y ajustar diferentes modelos lineales usando subconjuntos de datos, desde dos elementos (línea negra) hasta los 100 generados (línea gruesa, marrón). La figura 2 muestra cómo el modelo mejora su precisión (rendimiento) conforme aumenta la cantidad de datos, por lo que, según la definición de Mitchell, efectivamente podría considerarse un modelo de ML.

El modelo lineal que acabamos de ver es el más sencillo posible, pues solo involucra a dos variables y no tiene ningún componente aleatorio. La complejidad (y potencia) de dicho modelo puede ampliarse incluyendo algún elemento aleatorio que tenga en cuenta el posible error cometido por el modelo. Desde un punto de vista matemático, se trataría de ampliar el modelo de forma que:

- $Y_i \mid \{\boldsymbol{w}, \boldsymbol{x}_i\} = \boldsymbol{w}^T \boldsymbol{x}_i + \varepsilon_i \; con \; \boldsymbol{x}_i = [1, x_i]^T$ donde $\boldsymbol{w}$ son los coeficientes del modelo de regresión,
- $\varepsilon_i \sim N(0, 1/\tau)$ con τ conocida,
- $\boldsymbol{w} \sim N(\boldsymbol{\mu}_0 = \mathbf{0}, \Sigma_0 = \mathbf{I}_{2x2})$.

Figura 2.

Modelo de regresión lineal



Fuente: Elaboración propia.

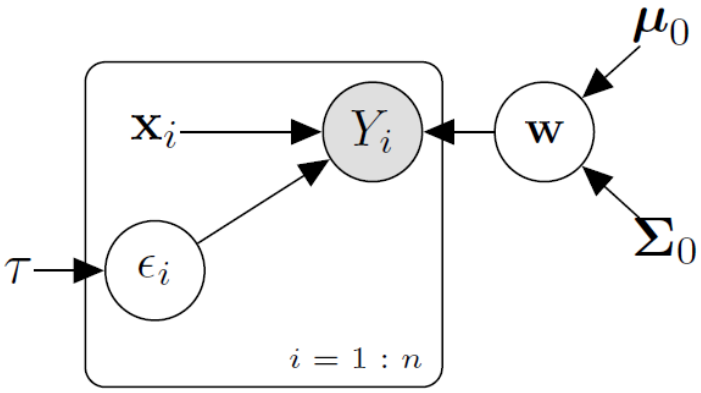
Descrito de esta forma, el modelo resulta aún más difícil de interpretar. Sin embargo, tiene la ventaja de que puede representarse como la red bayesiana de la figura 3, de manera que la propia estructura de la red indica cuáles son las interacciones entre los diferentes componentes del modelo. Es decir, aunque carezcamos de la formación matemática necesaria para entender los detalles técnicos expresados en los ítems anteriores, podemos extraer información muy válida sobre la estructura del problema.

Un caso particular muy popular dentro de las redes bayesianas es el llamado *Naive Bayes*, cuya estructura se muestra en la figura 4. En este modelo, el objetivo es resolver el problema de *clasificación*, donde la variable a predecir, Y, es de tipo categórico, mientras que las variables predictoras, $X_1,...,X_k$, pueden ser tanto categóricas como numéricas. La predicción se hace usando la distribución de probabilidad de la variable objetivo dadas las variables predictoras, que teniendo en cuenta las independencias codificadas por la estructura de la red, se puede expresar en términos de las distribuciones condicionadas de cada variable predictora dada la variable objetivo:

$$p(y \mid x_1,...,x_k) \propto p(x_1,...,x_k \mid y)p(y) = p(y)\prod_{i=1}^{k} p(x_1 \mid y).$$

Figura 3.

Modelo de regresión lineal representado como una red bayesiana

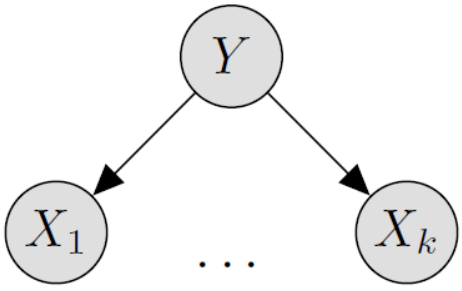


Fuente: Elaboración propia.

Precisamente, esta habilidad de descomponer el problema en unidades más pequeñas, hace que las redes bayesianas sean apropiadas en contextos de *big data* con muchas variables involucradas. Esta simplificación de la representación, además facilita el desarrollo de algoritmos de inferencia/predicción eficientes.

Figura 4.

Clasificador Naive Bayes



Fuente: Elaboración propia.

2.1. Las redes bayesianas como modelos generativos

Las redes bayesianas se enmarcan dentro de los llamados *modelos generativos*, frente al otro tipo de modelos predictivos, que son los *discriminativos*. Para entender la diferencia entre ambos, supongamos que queremos predecir el valor de una variable Y en función de otra variable X. Un modelo generativo aprende la distribución global (conjunta) de ambas variables, $p(x, y) = p(x \mid y)p(y)$, a partir de datos y calcula la distribución de la variable a

predecir condicionada a la variable predictora, $p(y \mid x)$, usando la regla de Bayes. Por contra, un modelo discriminativo aproxima $p(y \mid x)$ directamente a partir de los datos. Ejemplos de modelos discriminativos son las redes neuronales o la regresión logística. La principal ventaja de los modelos generativos es que, dado que calculan la distribución conjunta de todas las variables, pueden usarse para generar datos sintéticos acerca del problema que estamos resolviendo. Con los discriminativos no es posible hacer esto, dado que no podemos, en nuestro ejemplo, generar valores para *X* pues su distribución no ha sido calculada.

3. APLICACIONES DE LAS REDES BAYESIANAS

En esta sección trataremos de ilustrar la capacidad de las redes bayesianas para resolver problemas complejos mediante la descripción de tres aplicaciones, concretamente el juego del ajedrez, la mejora genética vegetal y las finanzas.

3.1. Redes bayesianas que aprenden a jugar al ajedrez

El ajedrez supone un reto considerable a la hora de tratar de construir un sistema de IA por lo siguiente:

- El sistema interactúa constantemente con el usuario y ha de responder a las acciones del mismo.
- La imposibilidad de calcular todas las jugadas posibles hacen necesario el uso de una heurística, cuya validez puede contrastarse en base a los resultados obtenidos.
- Existen diferentes estilos de juego o estrategias que tanto el usuario como el programa pueden adoptar.

Mostraremos a continuación cómo es posible construir un programa de ajedrez basado en redes bayesianas capaz de refinar la heurística de búsqueda, en base a la experiencia de juego del programa. En concreto, describiremos el programa de ajedrez BayesChess (Fernández y Salmerón, 2008).

El juego del ajedrez ha sido estudiado en profundidad por la IA, dentro de los llamados *juegos de información completa*. Veremos aquí que es posible usar una red bayesiana para actualizar la heurística de búsqueda conforme se dispone de más datos (más partidas de las que aprender). La heurística que hemos considerado se basa en dos aspectos: material (piezas de cada jugador en el tablero) y situación de cada pieza (dependiendo de la casilla que ocupen en un momento dado, las piezas pueden ser más o menos valiosas). De forma adicional, hemos dado importancia también al hecho de dar jaque al rey adversario, dado que, en igualdad de condiciones, una jugada que dé jaque puede ser preferible, pues restringe el conjunto de posibles respuestas del adversario ya que debe protegerse del jaque. La evaluación

del material se realiza asignando una puntuación a cada pieza. Hemos elegido la puntuación habitual en programas de ajedrez (tabla 1), donde el rey no tiene puntuación, pues no es necesario al estar prohibida su captura. En cuanto a la valoración de la posición de cada pieza sobre el tablero, hemos empleado una matriz de 8×8 para cada ficha, de forma que cada celda contiene la puntuación añadida al valor de la heurística en caso de que la pieza esté situada en ella. De esa forma se pueden favorecer, por ejemplo, movimientos que lleven a situar los caballos en posiciones centrales del tablero, donde tienen más margen de acción que en los bordes. En total, la función heurística está definida por 838 parámetros ajustables, que son el valor de cada pieza, el valor de dar jaque y el número almacenado en cada celda de cada una de las matrices 8×8 definidas anteriormente.

Tabla 1.

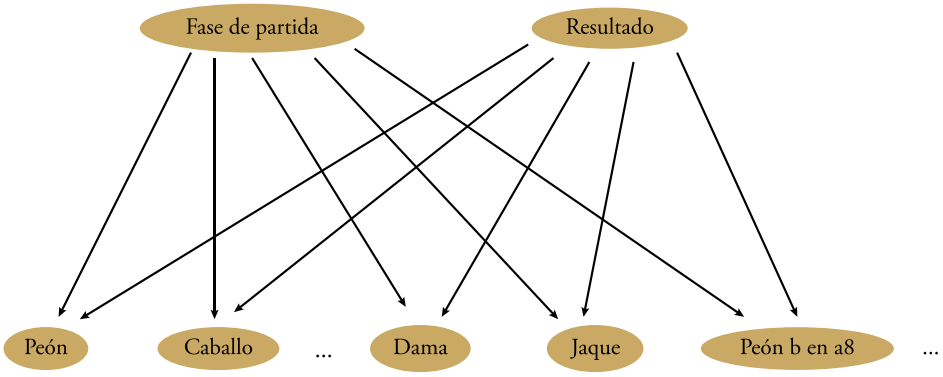
Puntuación de las piezas en la heurística empleada

Pieza	*Peón*	*Alfil*	*Caballo*	*Torre*	*Dama*
Puntuación	100	300	300	500	900

Para el ajuste de los parámetros de la heurística, hemos considerado una red bayesiana con estructura tipo Naive Bayes con la salvedad de que en lugar de una variable objetivo hay dos: la *fase actual de la partida* (apertura, medio juego o final) y el *resultado de la partida* (ganar, perder, tablas). Como variables predictoras, se han empleado todos los parámetros ajustables de la heurística, lo que significa que la red cuenta con un total de 776 variables con la estructura mostrada en la figura 5. El elevado número de variables viene dado principalmente porque hay una variable por cada una de las 64 posibles ubicaciones de cada una de las piezas en el tablero. La razón por la que se ha usado una estructura de red tipo Naive Bayes

Figura 5.

Estructura de la red bayesiana para el aprendizaje automático de la heurística



Fuente: Elaboración propia.

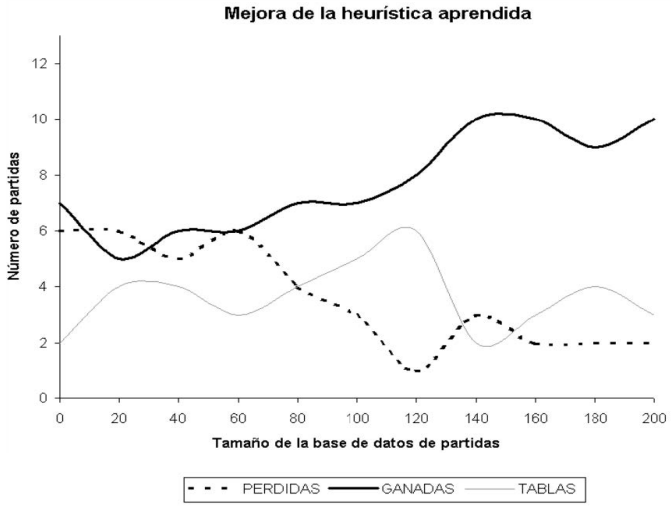
es precisamente el alto número de variables, ya que el uso de una estructura más compleja aumentaría drásticamente el tiempo necesario para evaluarla, lo que ralentizaría la evaluación de las posiciones durante la exploración del árbol de búsqueda.

Los parámetros de la red bayesiana se estiman inicialmente a partir de una base de datos generada enfrentando al programa contra él mismo, usando uno de los dos bandos la heurística tal y como se ha descrito anteriormente, y el otro una versión perturbada aleatoriamente, donde el valor de cada variable se incrementaba o decrementaba en un 20 %, 40 % o se mantenía a su valor inicial de forma aleatoria. Una vez construida la red bayesiana, BayesChess la utiliza para elegir los parámetros de la heurística. El proceso de selección consiste en instanciar las dos variables clase (fase de partida y resultado) y a partir de ahí se obtiene la configuración de parámetros que maximiza la probabilidad de los valores instanciados de las variables *Fase de partida y Resultado*. Por ejemplo, si instanciamos la variable resultado a ganar, elegirá la configuración de parámetros que maximizan la probabilidad de ganar, aunque ésta sea menor que la suma de las probabilidades de perder y hacer tablas. Esto puede ser equivalente a considerar que BayesChess adopta una estrategia agresiva. Por contra, puede optarse por minimizar la probabilidad de perder, o lo que es lo mismo, maximizar la de ganar o hacer tablas. Esto puede derivar en una estrategia de juego más conservadora.

La figura 6 muestra la capacidad de la red bayesiana para ajustar los parámetros de la heurística. La gráfica de la figura se refiere a un experimento en el que se usó una base de datos con 200 partidas jugadas entre la heurística inicial y una aleatoria. Se realizaron entonces

Figura 6.

Evolución de los resultados de la heurística aprendida conforme aumenta el número de partidas



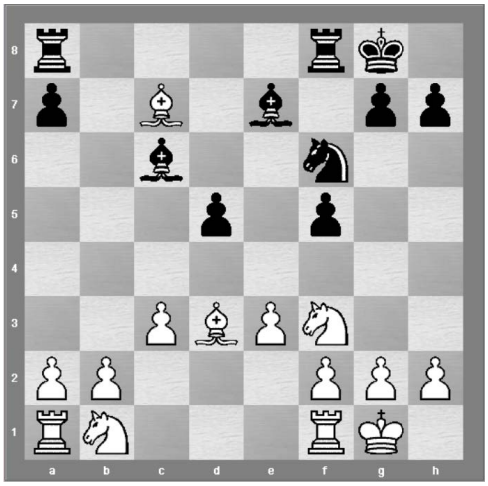
Fuente: Elaboración propia.

7 torneos de 15 partidas entre BayesChess con la heurística fija y él mismo con la heurística aprendida con subconjuntos de más o menos partidas de la base de datos. Se observa cómo la heurística aprendida mediante la red bayesiana llega a superar claramente a la fija, conforme aumenta el número de partidas, llegando a obtener alrededor de 10 victorias en 15 partidas.

En cuanto a la manera en que evoluciona la puntuación concreta asignada por la heurística aprendida a una posición, vamos a usar como ejemplo la posición mostrada en la figura 7,

Figura 7.

Posición de ejemplo en la que el jugador de las piezas blancas cuenta con dos peones y un caballo de ventaja



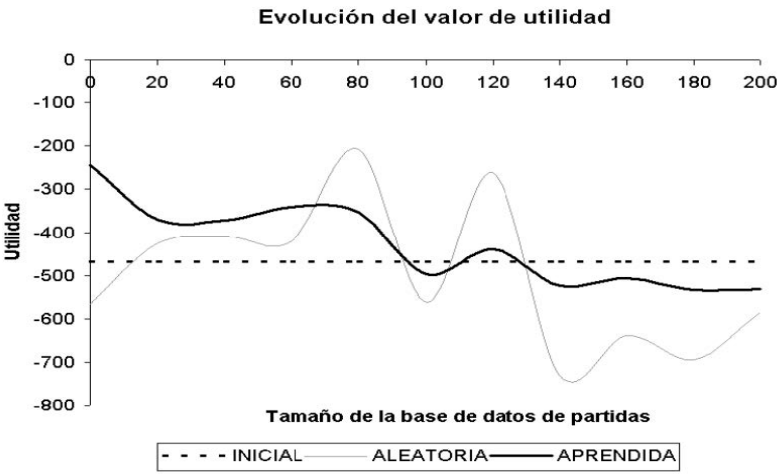
Fuente: Elaboración propia.

donde el bando blanco cuenta con dos peones y un caballo de ventaja, lo que, de acuerdo con la tabla 1 resultaría en una ventaja de alrededor de 500 puntos. La figura 8 representa la evolución de la puntuación asignada por la heurística. El punto de partida es una heurística con los parámetros inicializados al azar. Se observa que, conforme aumenta el número de partidas de las que aprende, la puntuación asignada por la heurística converge hacia una puntuación cercana a los 500 puntos antes mencionados.

Además de la capacidad de mejorar la heurística conforme aumenta la cantidad de datos (partidas) a partir de las que se aprende, una característica interesante es que el modelo aprendido es un *modelo generativo*, y por lo tanto podemos usarlo para obtener datos sintéticos. En este caso, podríamos usarlo para obtener ejemplos de buenas heurísticas (heurísticas ganadoras) o malas heurísticas (heurísticas perdedoras). Teniendo en cuenta que las heurísticas dicen cómo valorar las piezas y su posición en el tablero, este conocimiento podría ser una herramienta importante a la hora de entrenar a nuevos jugadores. Es decir, en este caso,

Figura 8.

Evolución de la puntuación asignada por la heurística aprendida a la posición de la figura 7 conforme aumenta el número de partidas



Fuente: Elaboración propia.

y a diferencia de lo que decíamos en la introducción, los humanos sí que podemos entender cómo juega la máquina y entender el por qué de cada uno de sus movimientos.

3.2. Aplicación en mejora genética vegetal

Otro campo en el que las redes bayesianas han sido aplicadas con éxito es el de la mejora genética vegetal. En concreto, describiremos el papel de las redes bayesianas en el contexto de un sistema de ayuda a la decisión para genetistas, a la hora de decidir cruces para obtener nuevas variedades de tomate con unas determinadas características deseadas (Nielsen *et al.*, 2014). Las empresas de semillas trabajan constantemente en la obtención de nuevas variedades vegetales con ciertas características que las hagan resistentes a enfermedades o que posean cualidades atractivas desde el punto de vista comercial (color, calibre, etc). Estas empresas disponen de bases de datos en las que figura el resultado de los cruces que han probado a la hora de obtener nuevas variedades, así como el resultado obtenido.

Además del conocimiento contenido en dichos datos, las empresas cuentan también con genetistas que atesoran experiencia y conocimiento. En ese sentido, de nuevo las redes bayesianas se muestran como una herramienta adecuada, pues son capaces de obtener conocimiento a partir de datos y combinarlo con conocimiento aportado por expertos humanos.

En el caso que nos ocupa, una vez construida la red bayesiana se trata de usarla para encontrar la combinación de variedades con mejores perspectivas de dar lugar a tomates con determinadas características, lo que se traduce en resolver la ecuación:

$$p^* = *\text{arg máx}_{P \in UP}(P = p \mid E = e),$$

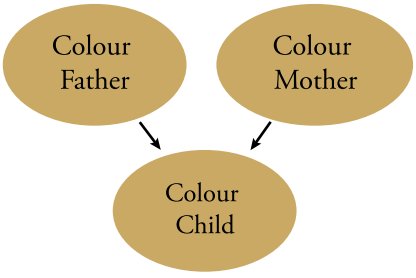
donde p son los posibles cruces y e las características deseadas.

La solución de la ecuación anterior, p^*, se corresponde con la combinación de variedades que maximiza la probabilidad de obtener las características deseadas. Sin embargo, dado el carácter *generativo* de las redes bayesianas, sería posible obtener mucha más información. Podríamos obtener datos sintéticos sobre posibles características de especies resultado de un determinado cruce, o ejemplos de variedades que se puedan cruzar con una variedad concreta para obtener ciertas características, etc.

Respecto a la incorporación de conocimiento experto procedente de los genetistas, se hizo por dos vías diferentes. En primer lugar (figura 9) se fijaron algunos de los valores de probabilidad de las distribuciones condicionadas. En este caso, hay reglas de combinación genética que determinan cómo se expresan determinadas características en función de su presencia o ausencia en los progenitores, como es por ejemplo, el color de los tomates. En ese caso, los valores determinados con certeza por la genética se incluyen por defecto, y el resto de parámetros se estiman a partir de los datos.

Figura 9.

Incorporación de conocimiento experto en forma de algunos valores de probabilidad



Father	Green		Red	
Mother	Green	Red	Green	Red
Child = Green	?	0.67	?	?
Child = Red	?	0.33	?	?

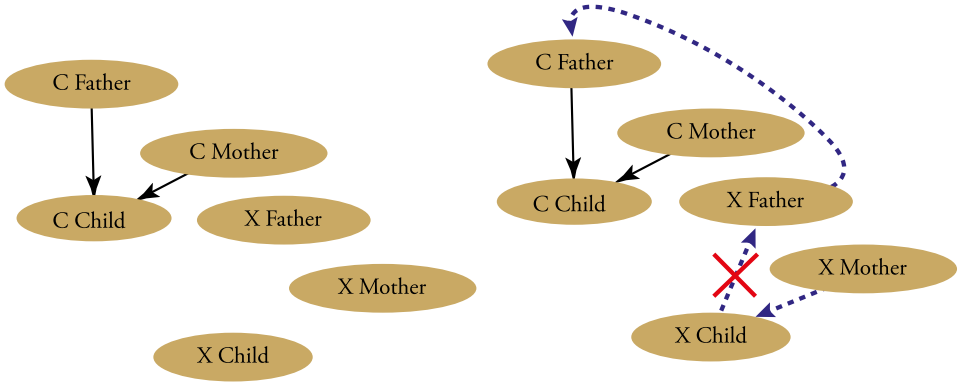
Fuente: Nielsen *et al.* (2014).

Por otro lado, hay también algunas conexiones entre variables que pueden incluirse sin necesidad de aprenderlas a partir de los datos, como son las que van desde las características de los progenitores (por ejemplo color) a las de los hijos. Igualmente, existe la posibilidad de prohibir conexiones que no tengan sentido desde el punto de vista del problema que estamos analizando. En este caso, prohibimos las conexiones desde la característica de un indivi-

duo a la de sus progenitores (en todo caso, debería estar en sentido contrario). En la figura 10, hemos indicado con líneas continuas las conexiones obligatorias, tachadas con una cruz las prohibidas, y en línea discontinua las conexiones aprendidas a partir de los datos.

Figura 10.

Incorporación de conocimiento experto en forma de conexiones obligatorias y conexiones prohibidas



Fuente: Nielsen *et al.* (2014).

3.3. Aplicación a la predicción de morosidad en créditos particulares

Disponer de una solución eficiente para la *predicción del riesgo* de crédito es crucial para reducir las pérdidas debido a procesos de negocio ineficientes, y de hecho el riesgo de crédito tiene impacto en las provisiones de fondos que las entidades financieras deben hacer en base a la regulación de los supervisores (como el Banco Central Europeo). Tales soluciones pueden ser usadas para *monitorizar la evolución de los clientes en términos de riesgo en operaciones de crédito* de cara a incrementar la solvencia de las instituciones. Desde el punto de vista del *machine learning*, la predicción del riesgo se ha afrontado como un problema de *clasificación supervisada*, en el que a partir de un histórico de datos, se construye un modelo orientado a predecir si un cliente entrará en mora en base al valor de una serie de variables acerca de ese cliente. Sin embargo, la predicción del riesgo de crédito presenta varios *retos diferenciadores* en relación con un problema estándar de clasificación supervisada:

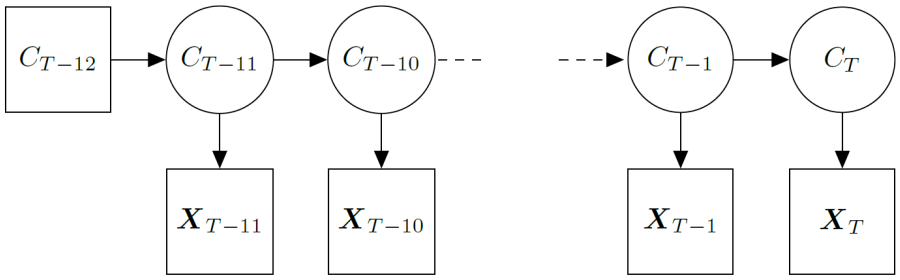
- La clasificación ha de realizarse en un contexto de *streaming*. La información disponible para cada cliente es un flujo (*stream*) de múltiples secuencias de datos sobre el tiempo. Es decir, en cada instante temporal t, recibimos un dato D_t relativo a una variable multidimensional con información sobre todos los clientes.
- *Feedback* sobre el valor de la clase retardado. El valor de la variable clase indica si un cliente entrará o no en morosidad *a 12 meses vista*, lo que dificulta la predicción.

- *Concept drift.* Este término se refiere al cambio de escenario. En nuestro caso abarca dos dimensiones. Por un lado, la distribución de los datos puede cambiar a lo largo del tiempo, y por otro lado, la relevancia de las variables a la hora de influir en la predicción también evoluciona.

En la aplicación que nos ocupa, los datos fueron proporcionados por el Banco de Crédito Cooperativo (BCC). Contienen información agregada por meses para un conjunto de clientes durante el período de abril de 2007 a marzo de 2014. Solo se consideran clientes activos, que son aquellos entre 18 y 65 años con al menos una operación en el período. Se excluyen los empleados de BCC porque tienen condiciones especiales. En total disponemos de información sobre 50.000 clientes por mes. Sobre cada cliente, se usaron 44 variables predictoras, denotadas por X_t, de las cuales 11 variables describen el *status* financiero del cliente y 33 variables son de carácter sociodemográfico. Cada cliente u tiene asociada una variable clase $C_t^{(u)}$ para cada instante temporal t que indica si el cliente en particular entrará en mora durante los próximos 12 meses. El esquema del modelo temporal se muestra en la figura 11, donde los rectángulos/círculos indican información disponible/no disponible cuando se hace la predicción en el instante T.

Figura 11.

Esquema del modelo temporal usado para la predicción de la probabilidad de entrar en morosidad



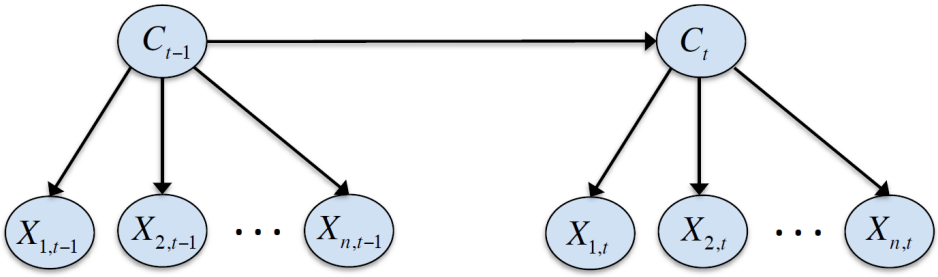
Fuente: Elaboración propia.

Dicho modelo temporal lo representamos como una red bayesiana como la de la figura 12, donde asumimos que solo las variables clase están conectadas a lo largo del tiempo y todas las variables predictoras en el instante t son condicionalmente independientes dada la variable clase. Esta suposición de independencia condicional queda habitualmente compensada por la drástica reducción del número de parámetros a estimar a partir de los datos. Así, esta estructura tipo Naive Bayes puede resultar adecuada en problemas con un elevado número de variables y gran volumen de datos (Friedman *et al.,* 1997). Con estas suposiciones, la distribución conjunta factoriza como:

$$p(c_{1:T}, \boldsymbol{x}_{1:T}) = \prod_{t=1}^{T} p(c_t \mid c_{t-1}) \prod_{i=1}^{n} p(x_{i,t} \mid c_t)\,.$$

Figura 12.

Red bayesiana codificando el modelo temporal de la figura 11



Fuente: Elaboración propia.

Las distribuciones $p(xi,t \mid ct)$ se estiman a partir de los datos etiquetados $\mathbf{D}_{T-\lambda}$, mientras que las distribuciones $p(c_t \mid c_{t-1})$ se estiman usando las transiciones de la clase de $\mathbf{D}_{T-\lambda-1}$ a $\mathbf{D}_{T-\lambda}$. A partir de la red de la figura 12, podemos realizar predicciones sobre la probabilidad de entrar en mora calculando la distribución condicionada de la variable clase para cada cliente u en el instante T dada toda la información recolectada hasta el momento, $\mathbf{D}_{1:T}$:

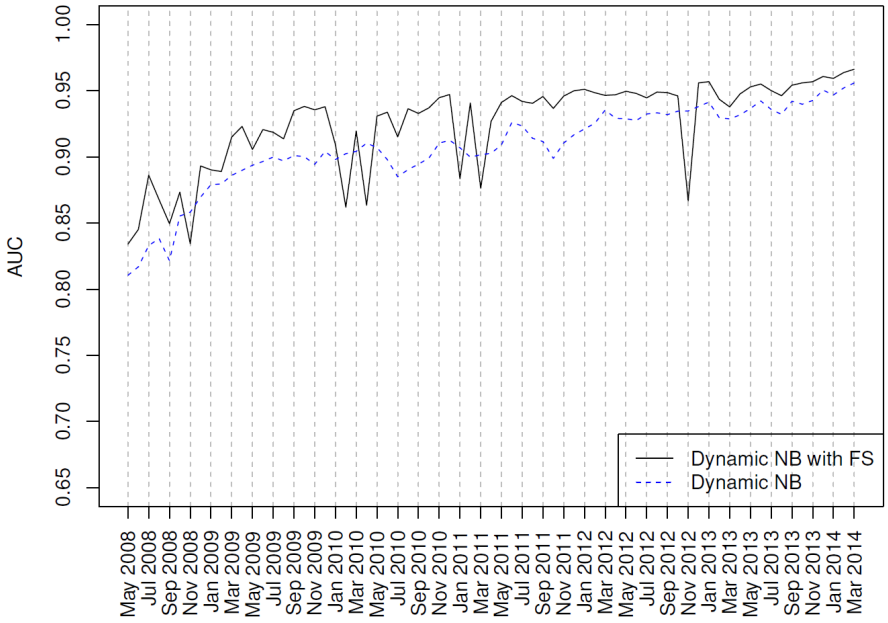
$$p\left(c_t^{(u)} \mid \boldsymbol{x}_{t-\lambda+1:t}^{(u)}, c_{t-\lambda}^{(u)}\right) \propto p\left(\boldsymbol{x}_t^{(u)} \mid c_t^{(u)}\right) \Sigma_{c_{t-1}^{(u)}} p\left(c_t^{(u)} \mid c_{t-1}^{(u)}\right) p\left(c_{t-1}^{(u)} \mid \boldsymbol{x}_{t-\lambda+1:t-1}^{(u)}, c_{t-\lambda}^{(u)}\right).$$

Aunque la expresión anterior puede resultar compleja desde un punto de vista matemático, lo cierto es que, al estar el modelo representado mediante una red bayesiana, los cálculos pueden hacerse usando algoritmos estándar de inferencia sobre las redes (Salmerón *et al.,* 2018). En realidad, esa es otra de las ventajas de usar este tipo de modelos, y es que, una vez hemos representado nuestro modelo como una red bayesiana, podemos aplicar las múltiples herramientas que se han desarrollado en ese contexto (Murphy, 2023). El rendimiento del modelo descrito se muestra en la figura 13, donde se observa el comportamiento con y sin selección de variables. La selección de variables se hizo construyendo los modelos con diferentes conjuntos de variables en cada instante temporal, eligiendo el que mejores resultados ofrecía. Se observa cómo la selección de variables es, en general, positiva (Borchani *et al.,* 2015). La precisión del modelo se ha medido usando el *área bajo la curva ROC* (AUC), que es la medida habitual de bondad de un modelo de clasificación cuando nos enfrentamos a modelos no balanceados, como es este caso, donde la gran mayoría de clientes nunca entran en mora.

Más allá de ser capaces de predecir si un cliente entrará en mora en un determinado horizonte temporal, puede ser de gran interés para una institución financiera el ser capaz de determinar en qué momento se produce un cambio significativo en la distribución de probabilidad de ciertas magnitudes de interés. Este fenómeno se conoce habitualmente como *concept drift* (Lu *et al.,* 2020). A continuación mostraremos cómo es posible sacar partido de la flexibilidad para el modelado de las redes bayesianas para detectar el *concept drift*. En con-

Figura 13.

Rendimiento del modelo de predicción en términos de área bajo la curva ROC



Nota: La línea continua muestra el modelo con selección de variables.
Fuente: Borchani *et al.* (2015a).

creto, veremos que esto es posible mediante la introducción de las llamadas *variables latentes*, que son variables que introducimos artificialmente en el modelo y que no son observables, es decir, que no tenemos datos acerca de ellas. Sin embargo, el hecho de que no tengamos datos acerca de ellas no significa que no podamos hacer inferencias sobre esas variables. De hecho, uno de los grandes avances de la estadística en las últimas décadas ha venido motivado por la posibilidad de hacer inferencias acerca de aquello que no podemos observar. La forma de hacerlo es a través de los llamados *modelos de variables latentes* (Blei, 2014), que son en realidad redes bayesianas con algunas de sus variables de tipo latente. La utilidad de las variables latentes es que nos permiten descubrir patrones en los datos que de otra forma podrían permanecer inadvertidos.

En la aplicación que estamos describiendo, introduciremos una variable oculta con la que trataremos de descubrir la posible existencia de *concept drift* en el *stream* de datos sobre los clientes que estamos analizando (Borchani *et al.,* 2015; Salmerón, 2020)]. La figura 14 muestra la estructura del modelo (red bayesiana) empleado. Las variables etiquetadas con las letras griegas α, β y θ con sus respectivos subíndices, se corresponden con parámetros de las diferentes distribuciones de probabilidad. La variable a predecir es *Y* y las variables

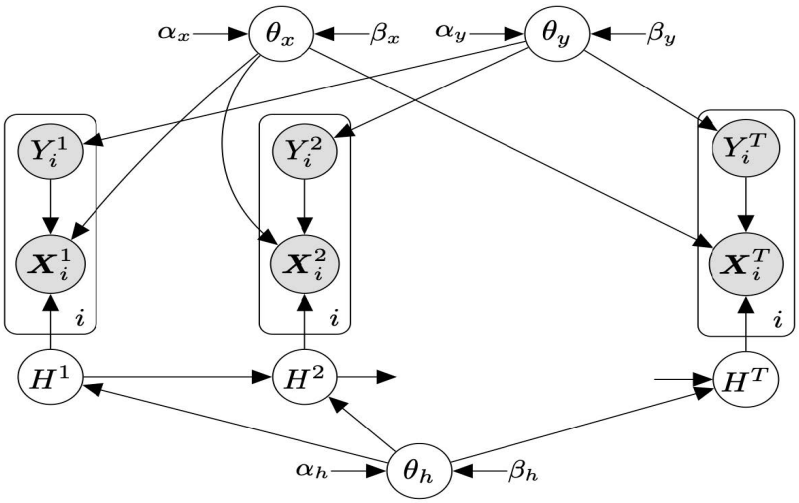
predictoras están representadas como el vector X. El modelo de predicción empleado es una red bayesiana de tipo Naive Bayes, que hemos ampliado para indicar que la secuencia temporal viene determinada por una variable oculta que evoluciona junto con el *stream* de datos $(H^1,...,H^T)$.

Después de estimar todos los parámetros de las distribuciones de la red bayesiana de la figura 14, comprobamos que la evolución de H^t sobre el tiempo captura el *concept drift* y refleja el efecto estacional en los datos, lo que puede verse en la figura 15. Es interesante destacar que la evolución de la variable oculta guarda muchas similitudes con la de la tasa de desempleo en la provincia de Almería (de donde son la mayoría de los clientes en la base de datos) durante el mismo período, que aparece en la figura 16, lo que llama la atención, pues esa variable no estaba incluida en la base de datos. Aunque no se ha realizado análisis de cointegración de ambas series temporales (la tasa de desempleo no forma parte del modelo), parece razonable pensar que la tendencia global de H^t ilustra el clima económico en Almería durante el período de estudio.

Para concluir con esta aplicación, hemos de añadir que, además de la capacidad de las redes bayesianas para predecir la morosidad y para monitorizar el cambio de distribución (*concept drift*), de nuevo podemos sacar partido de su carácter de modelo generativo. Podemos, a modo de ejemplo, usar la red bayesiana para obtener datos sintéticos de clientes que

Figura 14.

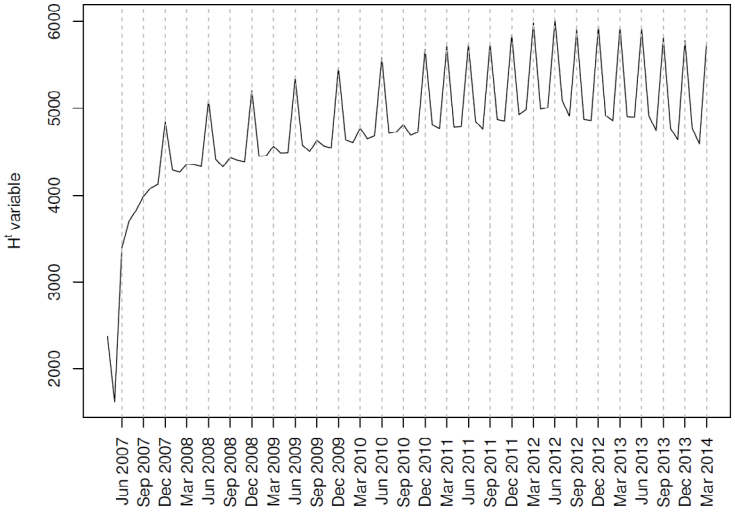
Red bayesiana para la detección del cambio de distribución de probabilidad para la predicción de morosidad



Fuente: Elaboración propia.

Figura 15.

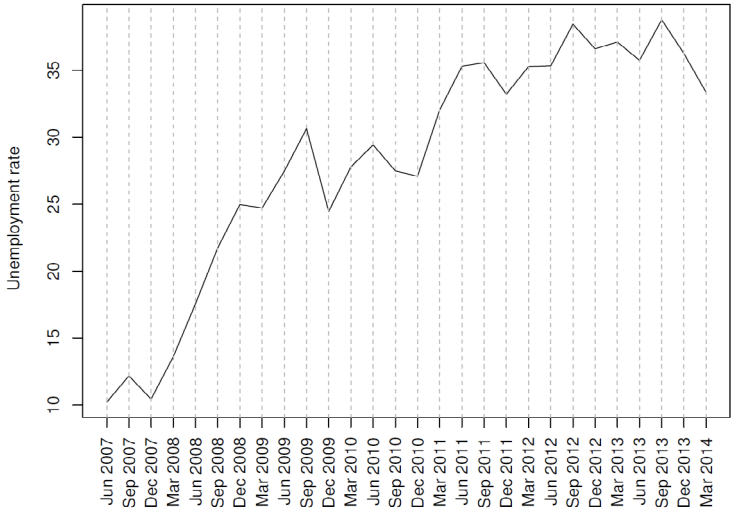
Evolución de la variable oculta a lo largo del tiempo



Fuente: Borchani *et al.,* 2015b.

Figura 16.

Evolución de la tasa de desempleo en la provincia de Almería a lo largo del tiempo



Fuente: Borchani *et al.,* 2015b.

previsiblemente van a entrar en morosidad, en términos de perfiles socio-demográficos o de actividad como cliente. Esto podría servir para que la institución financiera tome medidas mitigadoras del riesgo. Igualmente, podríamos generar ejemplos de buenos clientes, en términos de factores sociodemográficos, que podrían ser útiles a la hora de diseñar nuevas campañas de *marketing*.

4. CONCLUSIONES

En este trabajo hemos puesto de manifiesto que las redes bayesianas son una herramienta suficientemente flexible como para abordar aplicaciones de diferente naturaleza, en las que es posible combinar la potencia de métodos de *machine learning* con la incorporación de conocimiento procedente de expertos humanos. A su vez, la forma de proceder de este tipo de modelos es interpretable, lo que ayuda a sostener la confianza en las decisiones que se tomen en base a las predicciones del modelo.

Un aspecto importante es la amplia disponibilidad de *software* libre implementando los principales avances metodológicos desarrollados alrededor de las redes bayesianas (Masegosa *et al.,* 2019; Pérez-Bernabé *et al.,* 2020; Scutari, 2010). Este hecho permite un rápido prototipado y puesta en marcha de nuevas aplicaciones sin necesidad de volver a implementar los métodos necesarios de inferencia y aprendizaje.

Por otro lado, las redes bayesianas son, de forma natural, modelos generativos, ya que representan el conocimiento en forma de distribuciones de probabilidad a partir de las cuales se pueden simular ejemplos con determinadas características. La importancia de que un modelo cuente con carácter generativo ha quedado suficientemente patente en las aplicaciones que han surgido en los últimos años (generación de texto, vídeo, imágenes, audio, etc.) y aquí hemos tratado de ilustrar que en aplicaciones de diferente naturaleza, el hecho de disponer de un modelo generativo puede ser un importante valor añadido con nuevas posibilidades, por ejemplo, para la formación de personal usando el conocimiento representado por el modelo.

Finalmente, las redes bayesianas son una de las herramientas fundamentales del razonamiento causal (Pearl, 2009). El razonamiento causal, que va más allá de la simple predicción, abre la puerta a realizar inferencias sobre escenarios hipotéticos y a determinar las relaciones de causa-efecto entre las variables del problema, de forma que los modelos sean aún más interpretables por los humanos.

Referencias

Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1, 203–232.

Borchani, H., Martínez, A. M., Masegosa, A. R., Langseth, H., Nielsen, T. D., Salmerón, A., Fernández, A., Madsen, A. L., y Sáez, R. (2015a). Dynamic Bayesian modeling for risk prediction in credit operations. *The 13th Scandinavian Conference on Artificial Intelligence*. Halmstad, Sweden, November 5-6,. 72-83.

Borchani, H., Martínez, A. M., Masegosa, A. R., Langseth, H., Nielsen, T. D., Salmerón, A., Fernández, A., Madsen, A. L., y Sáez, R. (2015b). Modeling concept drift: A probabilistic graphical model based approach. *IDA'2015. Lecture Notes in Computer Science,* 9385, 72-83.

Fernández A., y Salmerón, A. (2008). BayesChess: A computer chess program based on Bayesian networks. *Pattern Recognition Letters*, 29, 1154-1159.

Friedman, N., Geiger, D., y Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, 131-163.

Jensen. F. V., y Nielsen, T. D. (2007). *Bayesian networks and decision graphs.* Springer.

Jumper, J., Evans, R., Pritzel, A. *et al.* (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596, 583–589.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., y Zhang, G. (2020). Learning under concept drift. A review. arXiv:2004.05785

Masegosa, A. R., Martínez, A. M., Ramos-López, D., Cabañas, R., Salmerón, A., Langseth, H., Nielsen, T. D., y Madsen, A. L. (2019). AMIDST: a Java toolbox for scalable probabilistic machine learning. *Knowledge Based Systems,* 163, 595-597.

Masegosa, A. R., Martínez, A. M., Ramos-Lopez, D., Langseth, H., Nielsen, T. D., y Salmerón, A. (2020). Analyzing concept drift: a case study in the financial sector. *Intelligent Data Analysis*, 24, 665-688.

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill.

Murphy, K. P. (2023). *Probabilistic Machine Learning. Advanced Topics*. MIT Press.

Nielsen, J. D., Gámez, J. A., y Salmerón, A. (2014). A tool based on Bayesian networks for supporting geneticists in plant improvement by controlled pollination. *International Journal of Approximate Reasoning*, 55, 74-83.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.

Pearl, J. (2009). *Causality. Models, inference and reasoning*. Second edition. New York: Cambridge University Press.

Pérez-Bernabé, I., Maldonado, A. D., Nielsen, T. D., y Salmerón, A. (2020). MoTBFs: An R package for learning hybrid Bayesian networks using mixtures of truncated basis functions. *The R Journal*, 12, 342-358.

Salmerón, A., Rumí, R., Langseth, H., Nielsen, T. D., y Madsen, A. L. (2018). A review of inference algorithms for hybrid Bayesian networks. *Journal of Artificial Intelligence Research,* 62, 799-828.

Scutari, M. (2010). Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35, 1–22.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., y Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arXiv:1712.01815

Sobre los autores

Amparo Alonso Betanzos

Es catedrática de Universidad en el área de Ciencias de la Computación e Inteligencia Artificial, en la Universidade da Coruña (UDC). Sus líneas de investigación actuales son el desarrollo de modelos de Aprendizaje Máquina Escalables, la Inteligencia Artificial Responsable y Sostenible, y el modelado basado en agentes inteligentes para la sostenibilidad de procesos, entre otras. Fue presidenta de la Asociación Española de Inteligencia Artificial (AEPIA) desde 2013-2021. Actualmente es miembro del Consejo Asesor en IA del Ministerio de Transformación Digital y Función Pública y miembro del Comité Español de Ética de la Investigación, del Ministerio de Ciencia, Innovación y Universidades. Es *Senior Member* de IEEE y de ACM y académica correspondiente de la Real Academia Española de Ciencias Exactas, Físicas y Naturales.

Ha recibido varios premios, como Helena Rubinstein-UNESCO "Women in Science" en España y finalista europea (1998), Premio Galicia TIC a la Innovación Digital (2004), Premio Galicia TIC a la trayectoria profesional (2019), Premio Josefa Wonenburger de la Xunta de Galicia (2020), Premio Gallega del año (2020), y Premio de la Asociación Española para la IA (2024).

Humberto Bustince

Es catedrático de Ciencia de la Computación e Inteligencia Artificial en la Universidad Pública de Navarra y profesor honorario de la Universidad de Nottingham. Fellow de la Asociación IEEE y de la asociación IFSA, que preside desde 2023. Miembro de Jakiunde desde 2018. Galardonado con la Cruz de Carlos III El Noble por el Gobierno de Navarra en 2017; con el Premio Nacional de Informática José García Santesmases 2019 y con el Premio a la Excelencia Científica de EUSFLAT en 2019.

José Duato

Ha sido catedrático en la Universidad Politécnica de Valencia durante más de 30 años, donde ha desarrollado soluciones que se han patentado y/o incorporado en múltiples procesadores, supercomputadores y estándares de comunicaciones. Actualmente es *Chief Technology Officer* (CTO) de la empresa Qsimov Quantum Computing y académico numerario de la Real Academia de Ciencias Exactas, Físicas y Naturales de España. Ha sido galardonado con el Premio Rey Jaime I de Nuevas Tecnologías, el Premio Nacional de Informática Aritmel y el Premio Nacional de Investigación Julio Rey Pastor en Matemáticas y Tecnologías de la Información y las Comunicaciones. Es autor de más de quinientas publicaciones que cuentan con más de 18.500 citas y de una decena de patentes internacionales.

Carlos Gómez Rodríguez

(Murcia, 1982). Es catedrático del área de Ciencias de la Computación e Inteligencia Artificial en la Universidade da Coruña. Su investigación se centra en el campo del procesamiento de lenguaje natural, con más de 150 artículos publicados. Ha dirigido un proyecto *ERC Starting Grant* y una *ERC Proof-of-Concept Grant*, además de distintos proyectos nacionales y autonómicos. En 2022, recibió el Premio Nacional de Investigación para Jóvenes "María Andresa Casamayor", el máximo reconocimiento en España para jóvenes investigadores en Matemáticas y Tecnologías de la Información; y fue reconocido como socio de honor de la Real Sociedad Matemática Española.

Bertha Guijarro-Berdiñas

Es doctora en Informática. Actualmente es profesora titular en la Facultad de Informática de la Universidade da Coruña, miembro de su Laboratorio de I+D en Inteligencia Artificial (LIDIA) y del Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC). Sus intereses de investigación se centran en aspectos aplicados y teóricos del aprendizaje automático (aprendizaje en línea, distribuido y federado, IA verde e IA Frugal), sistemas basados en conocimiento y modelado basado en agentes. Ha participado en más de 30 proyectos de I+D+I financiados por agencias europeas, nacionales y regionales, así como en varios contratos con empresas, y es coautora de más de cien publicaciones en libros, revistas internacionales y conferencias en el campo de la Inteligencia Artificial.

Ibai Laña

Es doctor en Inteligencia Artificial. Vive en Bilbao y trabaja en Tecnalia como analista de datos. Se licenció en Ingeniería Informática por la Universidad de Deusto (España) en 2006, el máster en Inteligencia Artificial Avanzada por la UNED (España) en 2014, y el doctorado en Inteligencia Artificial por la UPV/EHU en 2018. Actualmente es investigador principal en Tecnalia y profesor en el grado de Ciencia de Datos e Inteligencia Artificial en la Universidad de Deusto. Sus intereses de investigación se enmarcan en la intersección de los Sistemas Inteligentes de Transporte (ITS), el aprendizaje automático, el análisis de datos de tráfico y la ciencia de datos. Se ha ocupado de problemas de predicción de tráfico urbano, donde ha aplicado modelos de aprendizaje automático y algoritmos evolutivos para obtener predicciones a más largo plazo y más precisas. También está interesado en otros retos relacionados con el tráfico, como la estimación de matrices origen-destino o la detección de puntos de interés y trayectorias. Además del dominio de la movilidad, Ibai ha trabajado en el dominio del mantenimiento predictivo para diversas aplicaciones industriales, así como en la predicción de la demanda energética y la optimización de las redes eléctricas. El trabajo con grandes volúmenes de datos le ha llevado a dominar las herramientas de procesamiento de *big data* y las bases de datos noSQL. También ha sido investigador visitante en el Instituto de Investigación de Ingeniería del Conocimiento y Descubrimiento (KEDRI) de la Universidad Tecnológica de Auckland (AUT).

Pedro Larrañaga

Es catedrático de Universidad en el área de Ciencias de la Computación e Inteligencia Artificial en la Universidad Politécnica de Madrid. Colidera el Computational Intelligence Group, y codirige la ELLIS Unit Madrid. Su investigación se centra en las redes bayesianas, la selección de variables y los algoritmos de estimación de distribuciones, con aplicaciones a neurociencia computacional e industria. Es *Fellow* de la Asociación Europea para la Inteligencia Artificial, Premio Nacional Aritmel en Ciencias de la Computación, Premio de la Asociación Española para la Inteligencia Artificial, *Fellow* de la Academia Europea, académico de Jakiunde, e IEEE *Fellow*.

Alejandro Rodríguez Arias

Es estudiante de doctorado en Informática en la Universidad de A Coruña. Es ingeniero informático por la misma facultad y actualmente investiga sobre sistemas multiagente, modelado basado en agentes y la integración de ambas tecnologías.

Antonio Salmerón Cerdán

Es catedrático de Estadística de la Universidad de Almería especializado en modelos probabilísticos y aplicaciones en *big data* e inteligencia artificial. Más de sesenta artículos en revistas, más de setenta en congresos, Premio José Cuena de la CAEPIA 2001. *WP leader* del proyecto europeo, "Analysis of Massive Data Streams". IP de proyectos nacionales desde el año 2001. Socio fundador de la EBT Soluciones de Biología Computacional S. L. Codirector de siete tesis doctorales. Director del Centro para el Desarrollo y Transferencia de Investigación Matemática a la Empresa de la UAL. Exdirector del Departamento de Matemáticas de la Universidad de Almería.

Noelia Sánchez-Maroño

Es profesora titular en la Universidad de A Coruña en el área de Ciencias de la Computación e Inteligencia Artificial y es miembro de su Laboratorio de I+D en Inteligencia Artificial en el Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC). Sus intereses de investigación incluyen el diseño de métodos de aprendizaje automático y modelos basados en agentes aplicados a diversos problemas reales.

Funcas

Caballero de Gracia, 28
28013 Madrid
Teléfono: 91 596 54 81
Fax: 91 596 57 96

publica@funcas.es
www.funcas.es

ISBN 978-84-17609-94-8