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

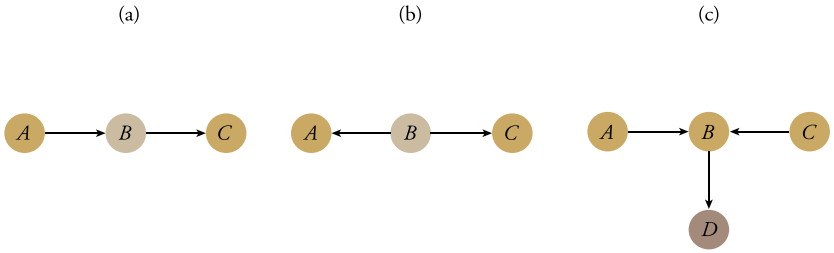
Las redes bayesianas (Jensen y Nielsen, 2007; Pearl, 1988) son modelos probabilísticos que tratan de cubrir los requisitos necesarios para ser operativas en entornos de *big data*. Formalmente, una red bayesiana es un grafo (un conjunto de vértices y aristas que los unen) dirigido (las aristas tienen 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).

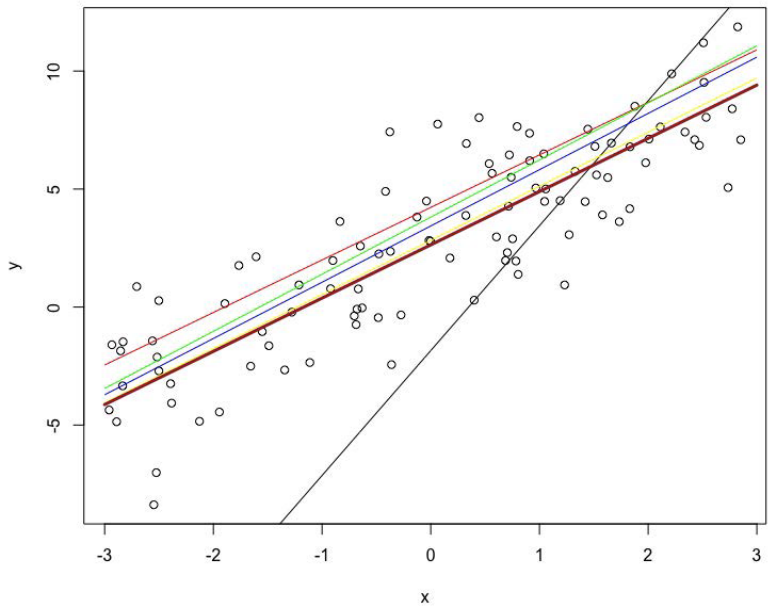
Consideremos un sencillo experimento consistente en generar un conjunto de 100 pares de valores (*X*,*Y*), correspondientes a la nube de puntos de la figura 2, y ajustar diferentes modelos lineales usando subconjuntos de datos, desde dos elementos (línea negra) hasta los 100 generados (línea gruesa, 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.

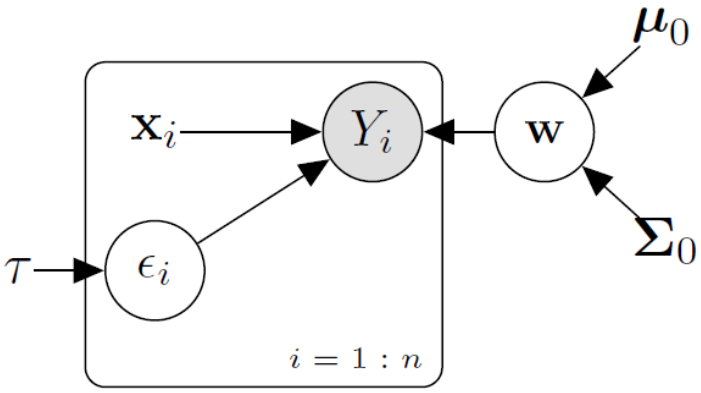
Descrito de esta forma, el modelo resulta aún más difícil de interpretar. Sin embargo, tiene la ventaja de que puede representarse como la red bayesiana de la figura 3, de manera que la propia estructura de la red indica cuáles son las interacciones entre los diferentes componentes del modelo. Es decir, 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

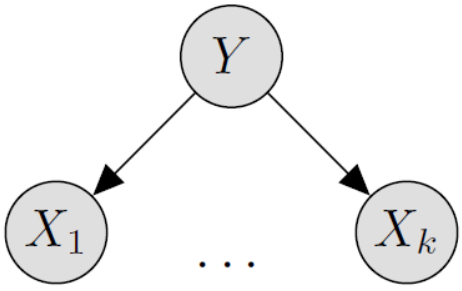


Fuente: Elaboración propia.

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.

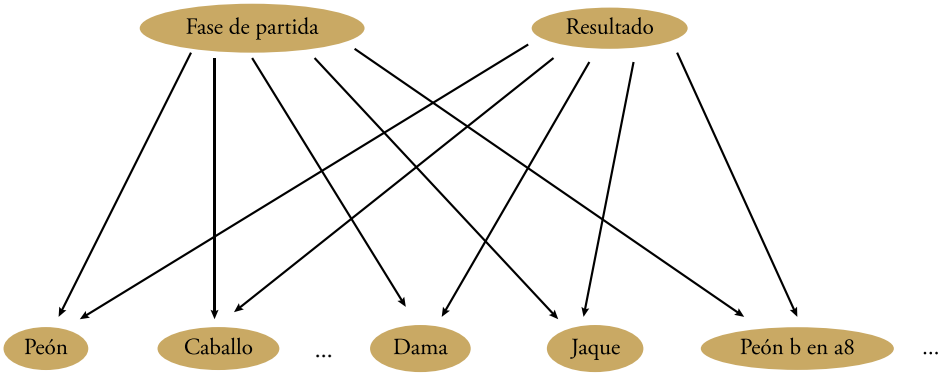
Puntuación de las piezas en la heurística empleada

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.

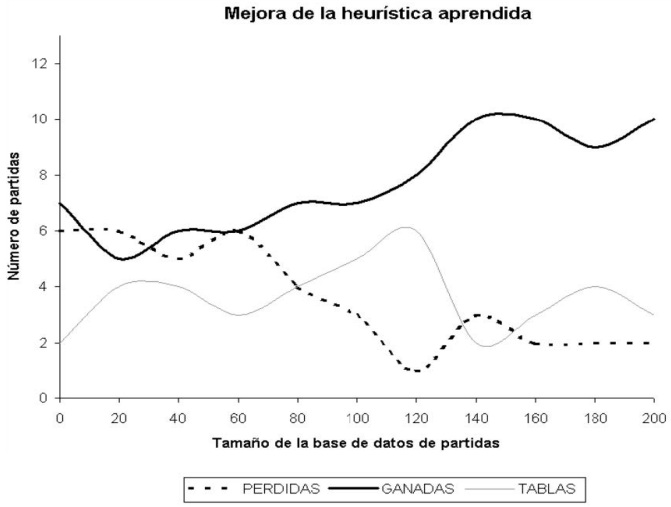
es precisamente el alto número de variables, ya que el uso de una estructura más compleja aumentaría drásticamente el tiempo necesario para evaluarla, lo que ralentizaría la evaluación de las posiciones durante la exploración del árbol de búsqueda.

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



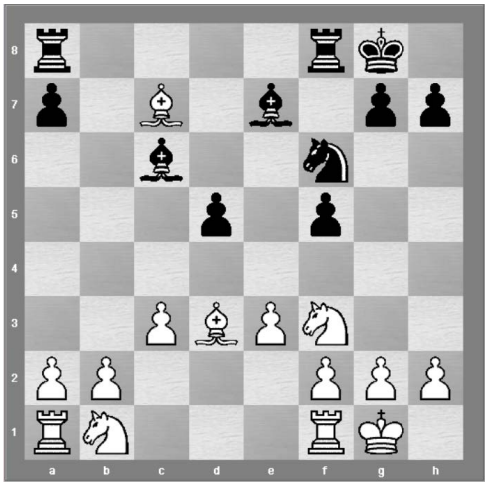
Fuente: Elaboración propia.

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



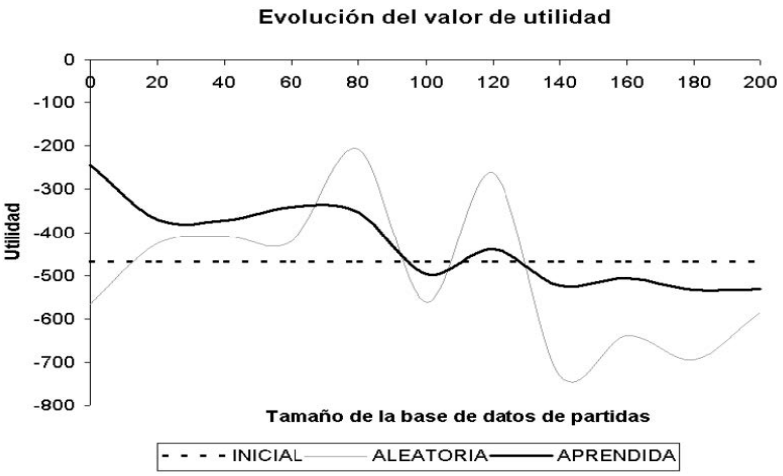
Fuente: Elaboración propia.

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:

$$p^* = *\text{arg máx}_{P \in UP}(P = p \mid E = e),$$

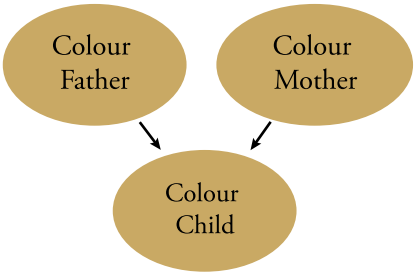
donde p son los posibles cruces y e las características deseadas.

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	?	?

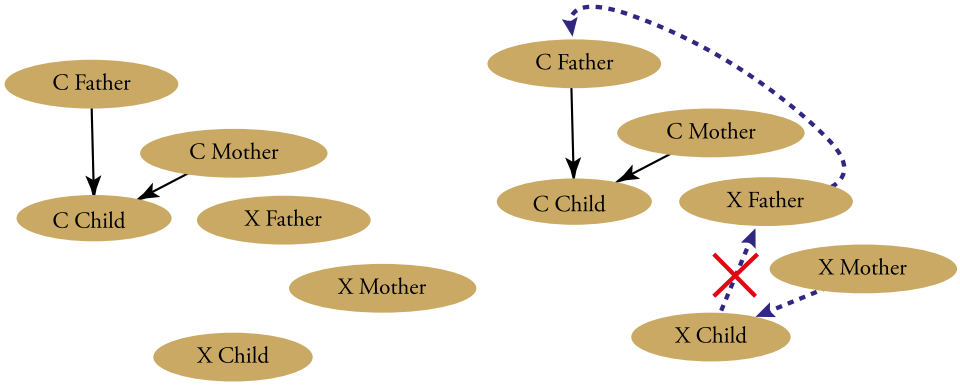
Fuente: Nielsen *et al.* (2014).

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:

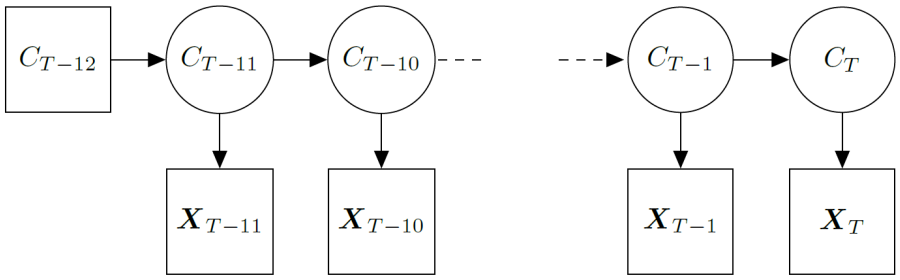
- La clasificación ha de realizarse en un contexto de *streaming*. La información disponible para cada cliente es un flujo (*stream*) de múltiples secuencias de datos sobre el tiempo. Es decir, en cada instante temporal t, recibimos un dato D_t relativo a una variable multidimensional con información 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



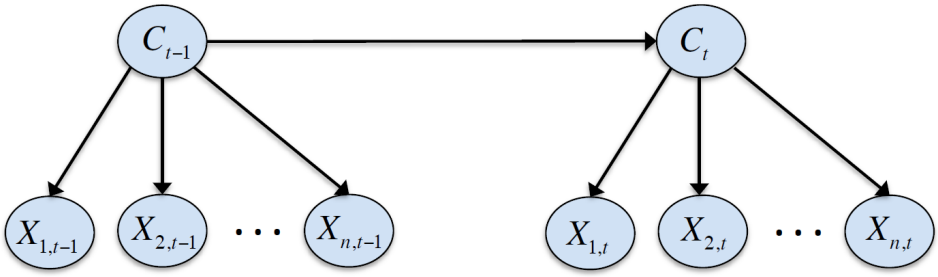
Fuente: Elaboración propia.

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, $\mathbf{D}_{1:T}$:

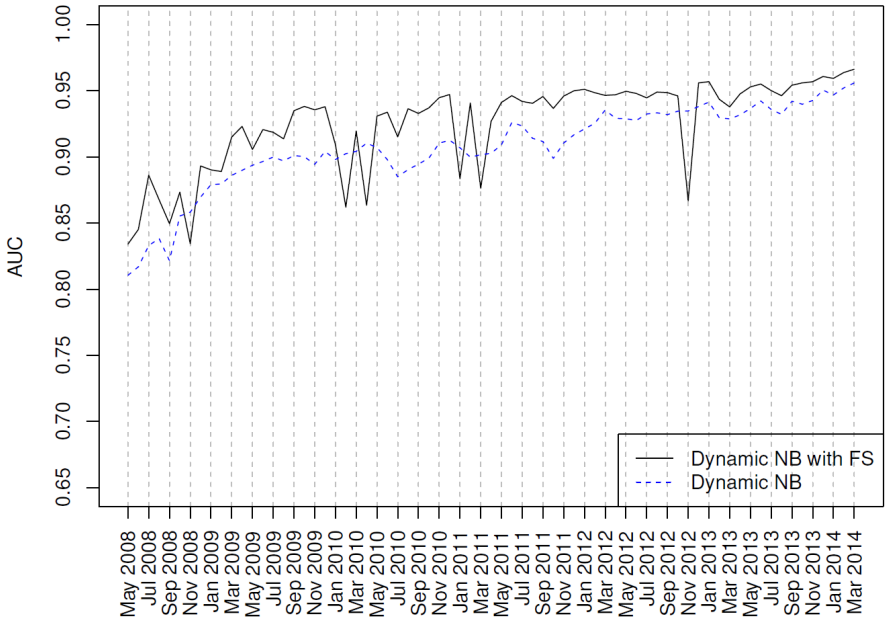
$$p\left(c_t^{(u)} \mid \boldsymbol{x}_{t-\lambda+1:t}^{(u)}, c_{t-\lambda}^{(u)}\right) \propto p\left(\boldsymbol{x}_t^{(u)} \mid c_t^{(u)}\right) \Sigma_{c_{t-1}^{(u)}} p\left(c_t^{(u)} \mid c_{t-1}^{(u)}\right) p\left(c_{t-1}^{(u)} \mid \boldsymbol{x}_{t-\lambda+1:t-1}^{(u)}, c_{t-\lambda}^{(u)}\right).$$

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

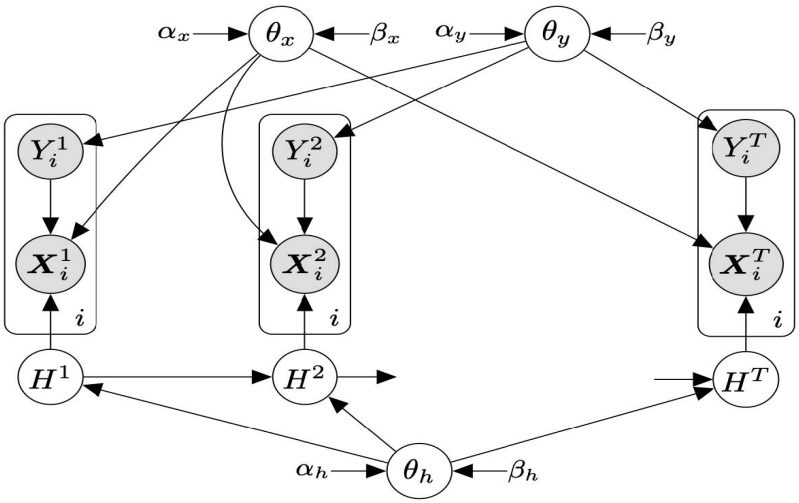
predictoras están representadas como el vector X. El modelo de predicción empleado es una red bayesiana de tipo Naive Bayes, que hemos ampliado para indicar que la secuencia temporal viene determinada por una variable oculta que evoluciona junto con el *stream* de datos $(H^1,...,H^T)$.

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.

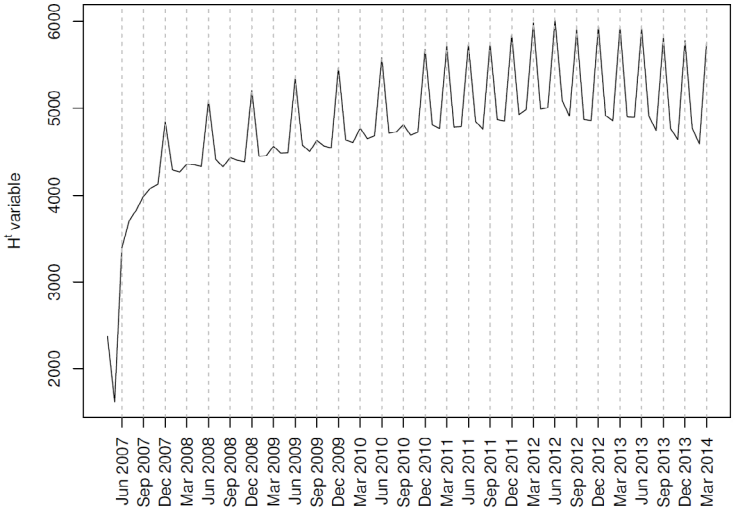
Red bayesiana para la detección del cambio de distribución de probabilidad para la predicción de morosidad



Fuente: Elaboración propia.

Figura 15.

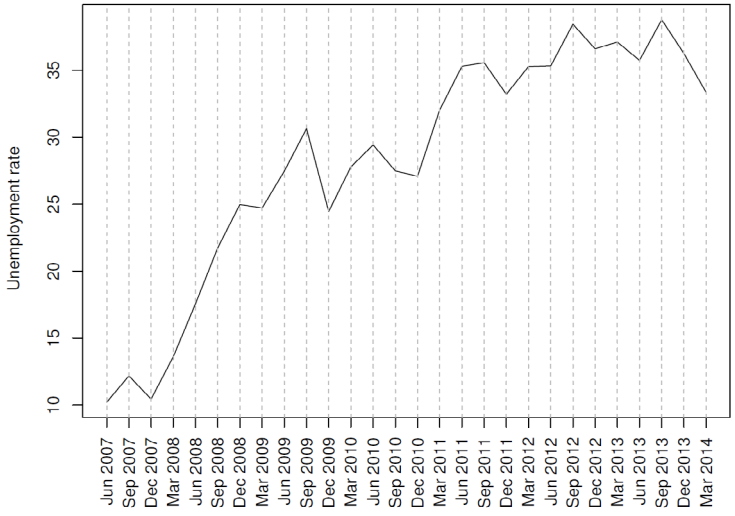
Evolución de la variable oculta a lo largo del tiempo



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