Práctica 5.
Modelos causales de historia completa.

Docente: Gustavo Landfried

Inferencia Bayesiana Causal 1
2do cuatrimestre 2024
UBA - UNSAM

Índice

1. Introducción **2**

2. TrueSkill Through Time **4**

3. Monty Hall temporal **6**

1. Introducción

En la práctica 1 implementamos varios modelos relacionados con el juego Monty Hall. En particular, el ejercicio 1.9 implementamos un modelo extendido que nos permitió resolver el problema que se generaba al evaluar modelos (Monty Hall vs Base) cuando en nuestra base datos había al menos un episodio en el que la persona que da la pista señala la caja que habíamos elegido. Como el modelo Monty Hall considera que éste caso era imposible, implementamos un modelo causal alternativo, en el que agregábamos una variable común a todos los episodios: la probabilidad de acordarse de actuar de acuerdo con las reglas del Monty Hall. Esta probabilidad podía ir de 0 a 1, y en cada episodio la persona que daba la pista se acordaba o no se acordaba de actuar de acuerdo a las reglas del Monty Hall. En la siguiente figura se agrega el modelo gráfico.

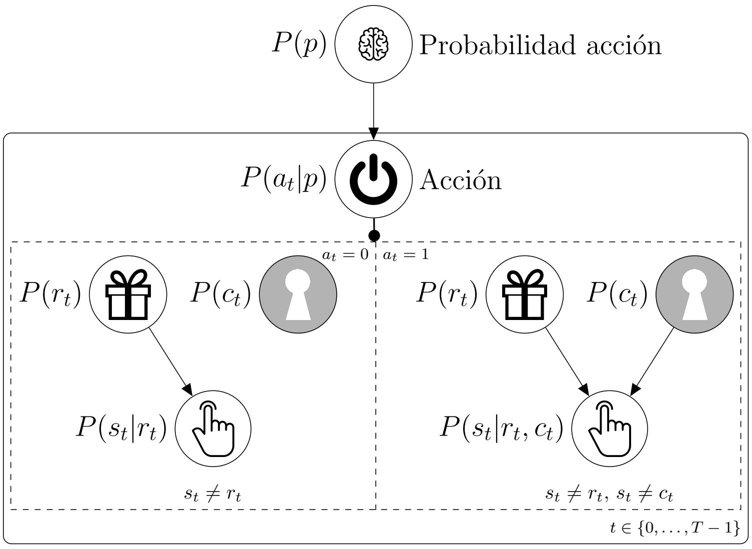


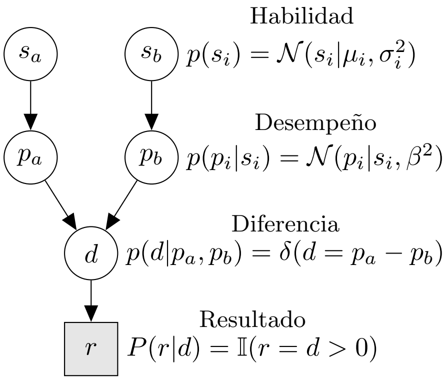
Figura 1: Modelo alternativo al Base y MontyHall. Dependiendo de si la persona se acuerda o no, actúa en función de un modelo u otro.

La distribución de probabilidad conjunta de este modelo luego de ver varios episodios es.

$$P(\boldsymbol{r}, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{a}, p|M_2) = P(p) \prod_{i \in \{0,\ldots,T-1\}} P(r_i)P(c_i)P(s_i|r_i, M_0)^{1-a_i} P(s_i|r_i, c_i, M_1)^{a_i} P(a_i|p)$$

Para evaluar la predicción a priori que realiza este modelo lo que hicimos fue calcular la predicción de cada evento usando el último posterior disponible de la variable común a todos los episodio, $P(p|\text{Episodios}_{0:t-1})$, y usar ese posterior como prior para predecir el siguiente evento. En este caso, usar el último posterior como prior del siguiente evento era el procedimiento correcto. Verificamos la validez de este procedimiento en la unidad 3, cuando calculamos los mensajes usando el algoritmo de suma-producto

Por otra parte, en la práctica 3 implementamos las clases `Gaussiana` y `Evento` que nos permitieron actualizar la estimación de habilidad luego de observar un evento. En la siguiente figura recordamos el modelos gráfico.



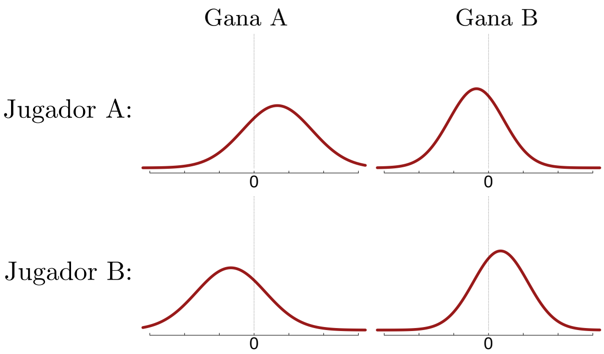
En la clase `Evento` calculamos la gaussiana que mejor aproxima al posterior exacto (sección **??**),

$$\widehat{p}(s_i|r, s_j) = \underset{\mu,\sigma}{\arg\min} \quad \text{KL}(\, p(s_i|r, s_j) \,||\, \mathcal{N}(s_i|\mu, \sigma^2)\,) \tag{1}$$

en términos de minización de la divergencia Kullback-Leibler entre la distribución verdadera y la familia de distribuciones gaussianas. Tener una aproximación gaussiana del posterior para usar como prior del siguiente evento es necesario para que la clase Evento pueda aprovechar las propiedades de la distribución gaussiana para actualizar eficientemente las creencias. Debido a que las habilidades cambian en el tiempo, es importante agregar alguna incertidumbre γ luego de cada paso.

$$\widehat{p}(s_{i_t}) = \mathcal{N}(s_{i_t}|\mu_{i_{t-1}}, \sigma^2_{i_{t-1}} + \gamma^2) \tag{2}$$

Sin embargo, a diferencias de lo que ocurre en el modelo Monty Hall extendido arriba señalado, en el caso de TrueSkill no es del todo correcto usar el último posterior como prior del siguiente evento. Propagar la información únicamente del pasado al futuro es intuitivamente una mala idea si pensamos el siguiente evento. Supongamos que dos personas desconocidas compiten en un video juego, y la primera la se gana a la segunda. Si luego la segunda le gana a las tres personas con mayor habilidad en ese video juego, intuitivamente diríamos que la primera también tiene mucha habilidad. Sin embargo, cuando usamos el último posterior como prior del siguiente evento estamos trasmitiendo la información en una dirección a través del sistema (del pasado al futuro) y por la tanto no actualizaremos la creencia sobre la habilidad de la primera persona, que se encuentra en el pesado. Este problema surge incluso cuando tenemos tan solo dos personas que se ganan entre sí mutuamente. En la siguiente figura mostramos el posterior de ambos eventos. Si ambas personas



se ganan mutuamente (sin que transcurra tiempo entre los eventos) intuitívamente diríamos que las personas tienen la misma habilidad. Sin embargo, cuando usamos el último posterior como prior del siguiente evento naturalmente, veremos naturalmente que: el posterior del primer evento tiene mayor incertidumbre que el posterior del segundo evento, pues a medida que acumulamos eventos estamos incorporando mayor información; y que no tienen misma media como ocurre en el primer evento, para el que solo tenemos un solo resultado sesgo que afecta al segundo evento.

2. TrueSkill Through Time

En la práctica 3 realizamos la implementación de las clases `Gaussiana` y la clase `Evento` En esta sección implementaremos la clase `Habilidad` y la clase `Historia`.

En esta sección explicamos cómo la información provista por los datos se propaga por toda la red bayesiana que contiene la historia de los eventos. TrueSkill Through Time crea un modelo causal único en el que todas las actividades históricas están vinculadas. La conectividad entre eventos surge de suponer que la habilidad de un jugador en un tiempo t depende de su propia habilidad en un tiempo anterior $t-1$. El modelo admite que en un mismo paso temporal t (e.g., día, semana, mes, año) un agente i participe en más de un evento. La figura 2 mostramos una parte de la factorización gráfica de la historia de eventos: los nodos vecinos a una variable de habilidad y los mensajes entre estos nodos. Por el *sum-product algorithm* sabemos que la distribución marginal

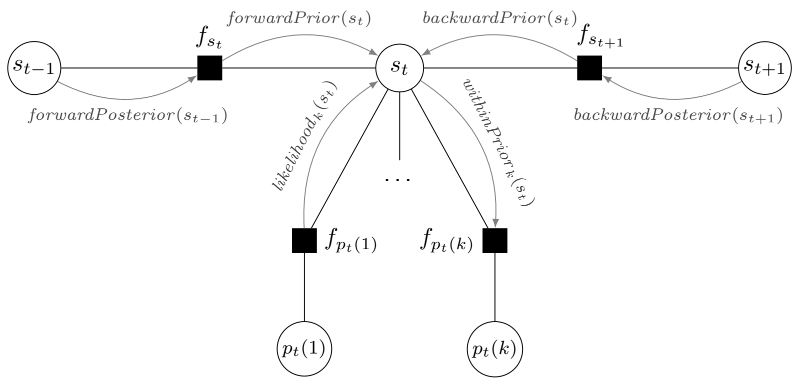


Figura 2: Nodos del grafo de factorización de la historia de eventos que son vecinos a la variable de habilidad de un agente en el paso temporal t. Las variables $p_t(j)$ representa el rendimiento p que ese agente tuvo en la j-ésima partida al interior del paso temporal t. Los nombres de las flechas representan los mensajes computados por el algoritmo de sum-product.

de cualquier variable es el producto de los mensajes que esta recibe de sus vecinos. Usando los nombres seleccionados en la figura 2, sabemos que la distribución posteriori de la habilidad de un agente i en un tiempo t es:

$$\text{posterior}(s_t) = \mathit{forwardPrior}(s_t) \cdot \mathit{backwardPrior}(s_t) \cdot \prod_{k=1}^{K_t} \mathit{likelihood}_k(s_t) \tag{3}$$

donde K_t es la cantidad de eventos en los que participa el agente en el paso temporal t. Los mensajes *likelihood* son las verosimilitudes de los eventos descritas en la unidad 3, y los mensajes *forwardPrior* y *backwardPrior* son las estimaciones de habilidad vecinas, a las que se le agrega cierta incertidumbre γ por el paso temporal.

$$\mathit{forwardPrior}(s_t) = \mathcal{N}(s_t \,|\, \mu_f, \sigma_f^2 + \gamma^2) \quad , \quad \mathit{backwardPrior}(s_t) = \mathcal{N}(s_t \,|\, \mu_b, \sigma_b^2 + \gamma^2) \tag{4}$$

donde $\mathit{forwardPosterior}(s_{t-1}) = \mathcal{N}(s_{t-1} \,|\, \mu_f, \sigma_f^2)$ y $\mathit{backwardPosterior}(s_{t+1}) = \mathcal{N}(s_{t-1} \,|\, \mu_b, \sigma_b^2)$. El mensaje *forwardPrior*(s_t) es el *forwardPosterior*(s_{t-1}) luego de que se le agrega la incertidumbre dinámica del tiempo t, f_{s_t}. Estos mensajes coinciden con las definiciones de prior y posterior del modelo básico de TrueSkill. La coincidencia surge de aplicar el *sum-product algorithm*. Pero de su aplicación también aparece la existencia del mensaje *backwardPrior*(s_t), que es el *backwardPosterior*(s_{t+1}) luego de que se le agrega la incertidumbre dinámica del tiempo $t+1$, $f_{s_{t+1}}$.

Los priors que se usan para computar las verosimilitudes de los evento son los mensajes descendentes *withinPrior*. Siguiendo el *sum-product algorithm* sabemos que los mensajes que envían las variable son la mutliplicación de los mensajes que reciben de atrás, por lo que el mensaje *withinPrior* es,

$$\begin{aligned} \textit{withinPrior}_q(s_t) &= \textit{forwardPrior}(s_t) \cdot \textit{backwardPrior}(s_t) \cdot \prod_{\substack{k=1 \\ q\neq k}}^{K_t} \textit{likelihood}_k(s_t) \\ &= \frac{\text{posterior}(s_t)}{\textit{likelihood}_q(s_t)} \end{aligned} \tag{5}$$

El *withinPrior* contiene toda la información del posterior(s_t), salvo la información que proviene de la verosimilitud del evento q para la que es prior. Para resolver la mutua dependencia entre verosimilitudes, actualizamos repetidas veces los mensajes hacia adelante y hacia atrás hasta alcanzar convergencia. En cada pasada hacia adelante guardamos el último mensajes que los agentes envían para adelante, *forwardPosterior*, y en cada pasada hacia atrás guardamos el último mensajes que los agentes envían para atrás, *backwardPosterior*.

$$\begin{aligned} \textit{forwardPosterior}(s_t) &= \frac{\text{posterior}(s_t)}{\textit{backwardPrior}(s_t)} = \textit{forwardPrior}(s_t) \cdot \prod_{k=1}^{K_t} \textit{likelihood}_k(s_t) \\ \textit{backwardPosterior}(s_t) &= \frac{\text{posterior}(s_t)}{\textit{forwardPrior}(s_t)} = \textit{backwardPrior}(s_t) \cdot \prod_{k=1}^{K_t} \textit{likelihood}_k(s_t) \end{aligned} \tag{6}$$

Los mensajes que no están todavía definidos, como por ejemplo el *backwardPrior*(s_t) en la primer pasada, deben ser remplazados por una forma neutral como la distribución gaussiana con varianza infinita. Este algoritmo requiere sólo unas pocas iteraciones lineales sobre los datos para converger y permite escalar a millones de observaciones en pocos minutos.

Esto produce una mutua dependencia entre estimaciones que nos obliga a usar iterativamente las últimas verosimilitudes disponibles hasta alcanzar convergencia. En la figura 3 mostramos como

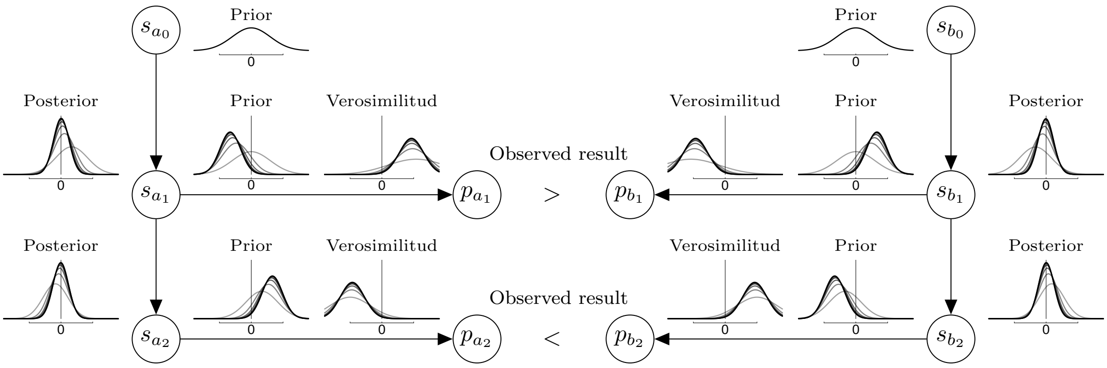


Figura 3: Convergencia de una red Bayesiana con dos eventos y dos agentes: la primera partida la gana el jugador a y la segunda la gana el jugador b. La luminosidad de las curvas indican el orden: la primera (la más clara) correponde a las estimaciones de TrueSkill, y la última (la más oscura) correponde con las estimaciones de TrueSkill Through Time.

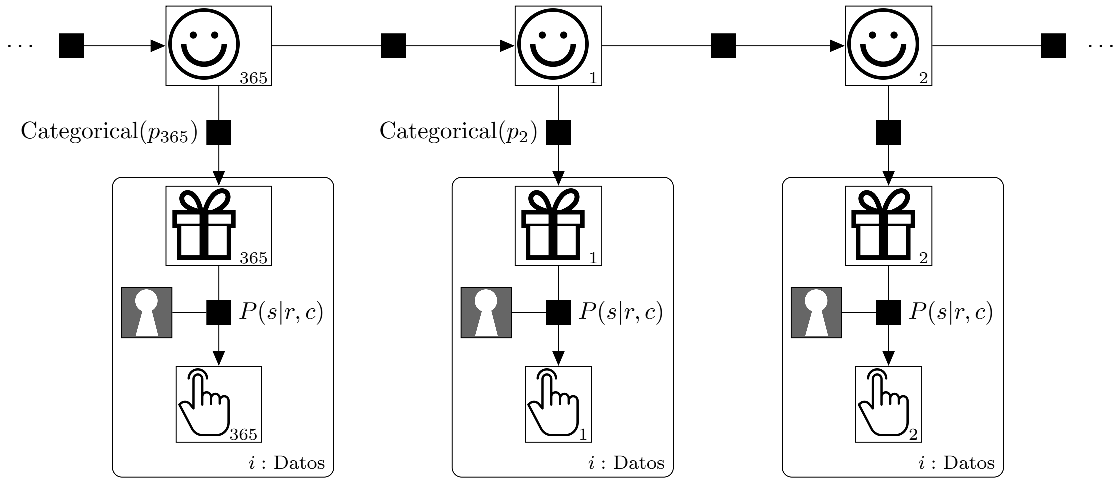
convergen las estimaciones en una red bayesiana dos agentes y dos eventos. TrueSkill Through Time recupera, de acuerdo a lo que sugieren los datos (una victoria cada uno), las verdaderas diferencias entre habilidades indicando que ambos jugadores tienen misma habilidad (posterior centrado en cero), a diferencia de TrueSkill que ofrece estimaciones sesgadas.

La ventaja de TrueSkill Through Time radica en que el modelo causal temporal permite que la información propage correctamente por todo el sistema. A diferencia de las redes neuronales que tienen estructuras regulares, estas redes bayesianas adquieren siempre una estructura compleja, creciéndo típicamente a millones de parámetros (e.g., videojuego). El procedimiento converge con unas pocas iteraciones lineales sobre los datos. La corrección de los sesgos es un paso fundamental para construir estimadores confiables que sirvan tanto para la toma de decisiones en áreas sensibles como para la evaluación de teorías científicas que utilicen la habilidad como dato observable.

3. Monty Hall temporal

El juego es un problema de inferencia con apuestas e intercambio de recursos. En esta práctica describiremos el problema de inferencia, pues el problema de toma de decisiones para apostar e intercambiar recursos se discutirá en la siguiente práctica.

El problema es un Monty Hall de cuatro puertas, en el que la persona que esconde el regalo tiene un "sesgo" que cambia a lo largo del año (de 365 pasos temporales), y el "sesgo" en un día t es el mismo en todos los años. Es decir, el modelo gráfico se puede describir mediante una red bayesiana con un ciclo, por lo tanto no es un DAG (Directed Acyclic Graph). Para estimar el sesgo de la persona que esconde el regalo, se proveé un archivo `csv` con 4 columnas (una por caja) y 365 filas (una por paso temporal) indicando la posición de los regalo en los últimos años. Es decir, las celdas del archivo contienen un numero entero $n_{t,r}$ que representa la cantidad de veces que en el día $t \in \{0, \ldots, 365\}$ del año el regalo fue escondido en la caja $r \in \{0, 1, 2, 3\}$.



El objetivo de la inferencia es estimar el sesgo de la persona para poder predecir en el futuro la posición del regalo, decidir por lo tanto qué caja elegir, y luego de recibir ver la pista apostar los recursos ω buscando maximizar la tasa de crecimiento a largo plazo. Siguiendo la idea del Monty Hall, antes de apostar la persona deberá cerrar una caja para recibir una pista, la información de que una de las otras caja no contiene el regalo. La madre naturaleza ofrece un pago $q = 2{,}75$ por cada hipótesis h, "el regalo se encuentra en la posición h". A diferencia del Monty Hall, en el que solo nos permiten elegir una sola caja, aquí debemos distribuir los recursos en proporciones b_h entre las hipótesis h, tal que $\sum_h b_h = 1$. Si en el paso temporal t la hipótesis h es verdadera y b_h es la proporción de los recursos apostados a esa hipótesis, entonces los recursos se actualiza como $\omega_{t+1} = \omega_t b_h q$. Además, entre pasos temporales, las personas podrán dar y recibir recursos entre sí.