UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Trabajo de Fin de Grado en Ingeniería Electrónica, Industrial y Automática

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño.

ESTUDIO DEL META-APRENDIZAJE COMO ALTERNATIVA A LA DISYUNTIVA ENTRE EL APRENDIZAJE POR REFUERZO BASADO EN MODELO Y EL APRENDIZAJE POR REFUERZO SIN MODELO.

AUTOR

IMANOL MIGUEZ QUINTELA

TUTORES

ALICIA HERRERO DEBÓN

JOAN SALVADOR ARDID RAMÍREZ

Curso Académico: 2021/2022

Resumen

Se comparará la eficiencia, ventajas y desventajas, de diferentes algoritmos de aprendizaje por refuerzo ("reinforcement learning", RL), tales como el refuerzo basado en modelo (model-based RL), el refuerzo sin modelo (model-free RL), el refuerzo híbrido (que surge de combinar los algoritmos con y sin modelo, hybrid model RL) y una alternativa a estos, basada en meta-aprendizaje. La optimización de los diferentes algoritmos se hará en base a dos contextos distintos: por un lado, cuando lo que se pretende es maximizar la recompensa y, por otro lado, cuando se pretende reproducir el comportamiento humano a través de su desarrollo cognitivo.

Abstract

In this TFG, I will analyze and compare the performance, advantages and limitations, of the following reinforcement learning (RL) algorithms: model-based RL, model-free RL, a hybrid RL algorithm that arises from combining the previous two, and an alternative RL algorithm based on meta-learning. These RL algorithms will be optimized, either according to reward maximization, or by the extent to which each RL algorithm is capable of accounting for human behaviour throughout cognitive development.

ÍNDICE

ÍNDICE DE FIGURAS

I. MEMORIA

1. Introducción

1.1 Contexto

Aprender interactuando con el entorno es probablemente el primer método de aprendizaje en el que la gente piensa cuando se habla de la naturaleza del aprendizaje. A un bebé nadie le enseña a agitar los brazos, jugar o gatear, sin embargo, tiene una conexión sensoriomotora con su entorno que produce una gran cantidad de información sobre la causa y el efecto. A lo largo de la vida de las personas esas interacciones ayudan a comprenderse mejor a sí mismos y a su entorno. Tanto si se aprende a tocar un instrumento como a mantener una conversación, las personas son conscientes de cómo responde el entorno a lo que hacen y tratan de influir en el resultado a través de su comportamiento [1].

Pese a que desde edades tempranas los niños muestran la capacidad de tomar decisiones simples basadas en la retroalimentación recibida por su entorno, las diferencias en la forma de tomar decisiones a medida que las personas se desarrollan han demostrado ser evidentes. Los individuos más jóvenes tienden a tener mayor perseverancia en acciones que antes proporcionaban resultados beneficiosos, disminuyendo esta con la edad [1]. Además, tanto niños como adolescentes suelen priorizar las acciones con recompensas inmediatas, mientras que los adultos tienen mayor visión de futuro en sus elecciones [2].

Por lo que, en el contexto del aprendizaje, a la hora de tomar decisiones, los modelos teóricos distinguen dos tipos de procesos de evaluación. Por un lado, un proceso lento y deliberativo, enfocado en comparar y prever las consecuencias entre diversas elecciones para identificar la acción con mejores resultados. Por otro lado, un proceso automático que vincula las acciones recompensadas a señales y contextos asociados, permitiendo la repetición de comportamientos previamente exitosos [3].

El aprendizaje a partir de la interacción es una idea fundamental que subyace en casi todas las teorías del aprendizaje y la inteligencia, en particular en la informática, dentro del área de la inteligencia artificial, el aprendizaje automático es el campo en el que se estudian algoritmos con el objetivo de proveer a los ordenadores la capacidad de aprender.

Dentro de lo que es el aprendizaje automático, los problemas de aprendizaje por refuerzo implican aprender qué acción es la más adecuada en función de la situación en la que se

encuentra, con el objetivo de maximizar una señal de recompensa (*reward*). En esencia, son problemas recursivos porque las acciones del sistema influyen en sus entradas posteriores. Además, hay que tener en cuenta que el agente no es informado de las acciones a realizar, como en muchas formas de aprendizaje automático, sino que debe descubrirlas tras diversos intentos y atendiendo al objetivo de obtener la mayor recompensa posible.

Teniendo en cuenta todo lo anteriormente mencionado, el aprendizaje por refuerzo es el enfoque que exploramos en este trabajo, ya que esta más centrado en el aprendizaje por objetivos a partir de la interacción que otros enfoques del aprendizaje automático.

1.2 Antecedentes

El propósito de esta sección es el de aportar al lector una base teórica del aprendizaje por refuerzo y sus elementos.

1.2.1 Aprendizaje por refuerzo

El aprendizaje por refuerzo es un área de la inteligencia artificial basada en la exploración de una serie de acciones y selección de las acciones óptimas para maximizar la señal de recompensa.

De esta manera, cuando se presenta una situación nueva, no se sabe cuál de las opciones disponibles permite mayor eficacia, por lo que se deben utilizar mecanismos de exploración para indagar qué opción es la más indicada para maximizar la recompensa en base a la opción más relevante. No obstante, se puede estar frente a contextos dinámicos y/o estocásticos, por eso es conveniente probar las diferentes opciones múltiples veces para estimar la probabilidad de recompensa y como ésta cambia en el tiempo. Esto hace que la combinación y/o alternancia entre los mecanismos de exploración y explotación no sea obvia y pueda llegar a ser compleja. El aprendizaje por refuerzo es por tanto un algoritmo que, al intentar maximizar la recompensa, optimiza el balance entre exploración y explotación de las diferentes opciones.

Otra característica clave del aprendizaje por refuerzo es que considera explícitamente todo el problema de un agente dirigido por un objetivo que interactúa con un entorno incierto. El enfoque del aprendizaje por refuerzo es comenzar con un agente completo e interactivo que busca un objetivo, por lo que, todos los agentes de aprendizaje por refuerzo tienen objetivos explícitos, pueden percibir aspectos de su entorno y pueden elegir acciones para influir en él.

Además, se suele suponer desde el principio que el agente actuará a pesar de la existencia de una incertidumbre significativa sobre el entorno al que se enfrenta. Si el aprendizaje por refuerzo implica también una planificación entonces tiene que abordar la interacción entre la planificación y la selección de acciones en tiempo real, así como la cuestión de cómo se adquieren y mejoran los modelos del entorno.

Asimismo, el aprendizaje por refuerzo también es considerado el tercer paradigma del aprendizaje automático, junto con, el aprendizaje supervisado y el aprendizaje no supervisado. Estos tres tipos de aprendizaje son diferentes entre sí puesto que el aprendizaje supervisado hace uso de un conocimiento a priori para generar una función que dé el resultado deseado y en el no supervisado no se conocen los resultados que se deben obtener por lo que el algoritmo construye un modelo en función de los patrones en los que detecta una relación entre las entradas.

Aunque los humanos hacen uso de estos tres tipos de aprendizajes, el aprendizaje por refuerzo es el más similar al tipo de aprendizaje que realizan tanto seres humanos como otros animales cuando se relacionan con el entorno y aprenden de la experiencia. Muchos de los algoritmos centrales del aprendizaje por refuerzo se inspiraron originalmente en los sistemas de aprendizaje biológicos [4].

1.2.2 Elementos del aprendizaje por refuerzo

A continuación, se realiza un breve resumen de los elementos involucrados en la elaboración de modelos de aprendizaje por refuerzo.

1.2.3 Agente y entorno

Un agente es un programa o ser vivo que tiene la función de tomar decisiones y, por lo tanto, la capacidad de realizar diferentes acciones que le ayuden en dicha de toma de decisiones.

En contraposición, un entorno o ambiente es la representación de un problema, es decir, el universo en el que se encuentra el agente y con el cual interactúa (Fig. 1). Así, debido a las acciones tomadas por el agente, el entorno genera estímulos o contextos, denominados estados, y recompensas, que pueden ser positivos o negativos.

Ambos componentes están en continua interacción y, como resultado, el agente pretende generar una influencia en el entorno a través de la realización de acciones, mientras que el entorno responde a esas acciones. En función de si al agente tiene o no conocimiento del modelo definido en el comportamiento del entorno, se pueden diferenciar dos tipos de modelos [5]:

- En el enfoque basado en modelo (model-based, MB) se utiliza un modelo predictivo del mundo para elegir la mejor acción.
- En el enfoque sin modelo (model-free, MF), el sistema trabaja solo mediante ensayo y error, recayendo todo el peso de la elección de una acción en la recompensa obtenida.

Fig. 1. Esta imagen muestra la interacción entre el agente y el ambiente. El agente realiza una acción que afecta al ambiente y este responde generando un estado y una recompensa (Imagen obtenida del blog Aprende Machine Learning, *https://www.aprendemachinelearning.com/aprendizaje-por-refuerzo/)*

1.2.4 Estado

El entorno se interpreta como un conjunto de variables, que cambian en función del problema a resolver. Así, el conjunto de variables y sus posibles valores se denomina espacio de estados. Los estados son, por lo tanto, los indicadores del ambiente, de cómo se encuentran los diferentes elementos que componen el entorno en un momento determinado.

1.2.5 Acción y función de transición

Para cada estado, existe la posibilidad de realizar un conjunto de acciones entre las cuales el agente elegirá una. Por lo que, el agente tiene influencia en el entorno a través de las acciones escogidas, y a su vez, el entorno puede establecer cambios de estado como respuesta a la acción. La función de transición o probabilidad de transición entre estados es la responsable de llevar a cabo la exploración mencionada [5].

1.2.6 Recompensa

Como respuesta a las acciones ya mencionadas, el entorno proporciona al agente una recompensa, la cual es una retroalimentación sobre la última acción llevada a cabo para lograr completar satisfactoriamente la tarea del agente. En este caso, la función responsable del proceso se denomina función de recompensa. Debido a que la tarea del agente es alcanzar la mayor recompensa posible, esta función es el impulso necesario para que el agente actúe con el comportamiento deseado.

Como, generalmente, sólo al terminar el proceso es posible conocer si la acción seleccionada por el agente es correcta o no, la definición de la función de recompensa es uno de los elementos más complejos en la modelización del problema [5].

2. Estudio de necesidades

2.1 Objetivo

Este trabajo tiene un doble objetivo. Por un lado, entender mejor el comportamiento humano en el proceso de toma de decisiones, observando como seleccionan sus acciones en base a la retroalimentación generada al recibir o no recompensa. Por otro lado, comprobar hasta donde pueden llegar los algoritmos cuando se busca maximizar la recompensa.

Los datos que usamos para la realización de la tarea están disponibles en internet y pertenecen al estudio realizado por Decker et al., 2016 [6]. Por lo tanto, estos consisten en una población de 59 sujetos y en torno a 200 *trials* por sujeto. Luego, al tratar de explicar el comportamiento humano, puede generar ruido en la obtención de datos, ya que se trata de una muestra relativamente pequeña. Además, para la comprobación de los algoritmos cuando se trata de maximizar la recompensa, utilizamos la tarea creada en dicho artículo, la cual consiste en dos etapas de elección, la primera con un estado y la segunda con dos estados.

Así mismo, para los modelos model-free y model-base utilizados en este trabajo, nos basaremos en los generados por Otto et al., 2013 [7].

2.2 Metodología de obtención de datos del estudio

En el estudio de Decker et al., 2016 [6] participaron 59 sujetos: 20 niños (11 de los cuales eran niñas), 20 adolescentes (12 de los cuales eran chicas) y 19 adultos (11 de los cuales eran mujeres). Además, se adaptó una tarea de aprendizaje secuencial [8] diseñada para disociar el

refuerzo basado en modelo (MB) y el refuerzo sin modelo (MF), utilizando una narrativa apta para niños.

Así, el objetivo de la tarea consistía en alcanzar una recompensa compuesta por dos etapas de decisión (Fig. 2). En la primera etapa se partía de un estado 1 en el cual se debía escoger entre dos acciones (nave verde o azul), una con un 70% de probabilidades de ir al estado 2 (planeta rojo) y un 30% de probabilidades de ir al estado 3 (planeta morado) y la otra con los porcentajes invertidos (30% - 70%, respectivamente). En la segunda etapa, el sujeto tenía que escoger entre dos acciones diferentes (alienígenas rojos y morados) en función del estado alcanzado tras la elección de la primera etapa. Esta última acción tenía una probabilidad de proporcionar recompensa, la cual variaba (Fig. 2, Probability of Winning) en cada uno de los ensayos (*trials*).

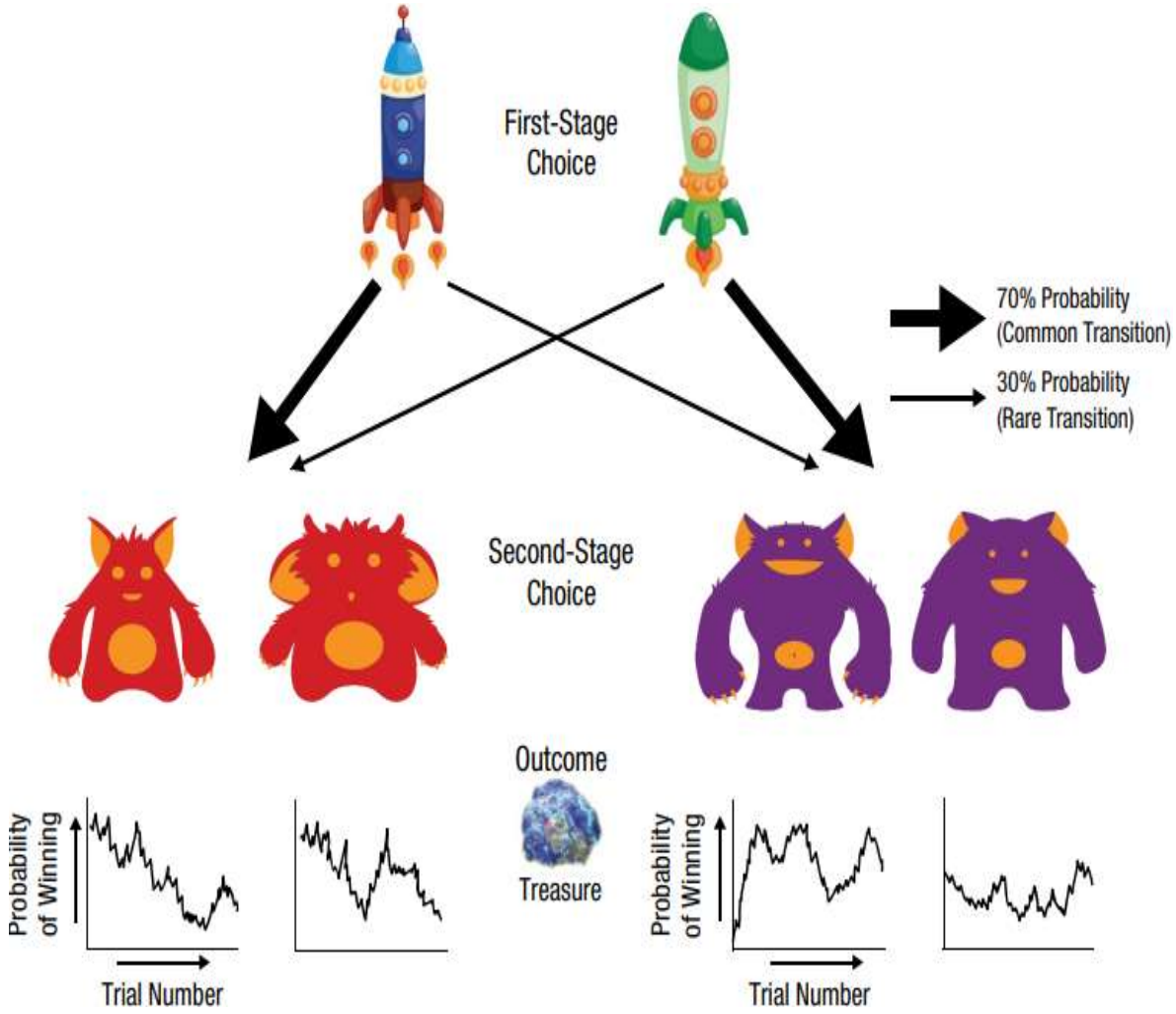


Fig. 2. Imagen del esquema del funcionamiento de la tarea. Se puede ver que en la primera etapa hay dos opciones, elegir el cohete azul o el cohete verde. Cada uno de ellos con unas probabilidades estáticas (70% vs 30%) de ir al planeta de los alienígenas rojos y al de los morados. La elección de la segunda etapa consiste en elegir entre uno de los dos alienígenas del planeta en el que se encuentre. Además, contiene las gráficas de la probabilidad de recompensa de cada alienígena en función del *trial*.

3. Planteamiento de alternativas y justificación de la solución adoptada

Para empezar a presentar nuestro proyecto en primer lugar describiremos los diferentes lenguajes de programación que podrían usarse, así como los algoritmos que se utilizarían en cada caso. Después comentaremos la solución que hemos adoptado en nuestro caso.

3.1 Lenguajes de programación

A continuación presentamos los lenguajes de programación más utilizados en este ámbito, los softwares asociados y sus principales características.

R.

Es un lenguaje de programación interpretado, que no necesita una compilación previa, por lo que ejecuta las instrucciones directamente y es utilizado generalmente para la computación estadística y gráfica. Además, es apto tanto para Windows, como MacOS y plataformas UNIX (Linux). También, hay que remarcar, que es código abierto desarrollándose de manera colaborativa y abierta, publicando los usuarios paquetes que amplían su configuración base.

Python.

Es un lenguaje de programación de propósito general y de alto nivel, que, además, es código abierto. Destaca por su código sencillo, legible y limpio, así como, la gran cantidad de librerías que posee. Al ser multiplataforma, tiene compatibilidad con Windows, MacOs y Linux.

C++.

Es un lenguaje de programación que proviene de una ampliación del lenguaje C, añadiéndole la funcionalidad de manipular objetos. Sin embargo, a pesar de ser un lenguaje con muchos años, su gran potencia lo convierte en uno de los lenguajes de programación más demandados.

Matlab.

Es un entorno de programación para el desarrollo de algoritmos, análisis de datos, visualización y calculo numérico. Usa un lenguaje de programación propio llamado M. El entorno hace uso de cajas de herramienta que permiten aprender y aplicar tecnologías específicas, las cuales están disponibles en áreas como el procesamiento de señales, los

sistemas de control, las redes neuronales y las simulaciones. A diferencia de los otros entornos mencionados en esta sección, éste es gestionado por la empresa privada MathWorks® y, al igual que los demás, es compatible con Windows, MacOs y Linux.

Tras sopesar los posibles lenguajes y entornos disponibles, optamos por el entorno de trabajo Matlab. La elección se debe a la cantidad de librerías enfocadas a la optimización que posee. Concretamente, utilizaremos BADS (*Bayesian Adaptive Direct Search)*, que es un algoritmo de optimización bayesiana diseñado para resolver problemas de optimización difíciles, en particular los relacionados con el ajuste de modelos computacionales, como los que se generarán en este trabajo.

Otra de las razones que nos ha hecho decantarnos por este entorno es que BADS también ha sido probado para ajustar modelos conductuales, cognitivos y neuronales, además de haberse comparado su rendimiento frente a otros optimizadores comunes y de última generación de Matlab, como fminsearch, fmincon y cmaes, comportándose a la par o mejor que estos. Asimismo, tampoco requiere ningún ajuste específico y se ejecuta del mismo modo que otros optimizadores integrados en MATLAB, como fminsearch.

3.2 Modelos

Existen dos modelos de aprendizaje por refuerzo a partir de los cuales generamos el resto de los modelos de modelos de aprendizaje por refuerzo.

Model-Free.

El algoritmo obtiene información solo de la recompensa final obtenida al terminar la tarea que se le haya encomendado.

Model-Based.

A diferencia del modelo MF, el MB, además de obtener información de la recompensa recibida, tiene embebida información del entorno. Por ejemplo, en el caso descrito anteriormente por la Fig. 2, las probabilidades de transición (70% vs 30%) entre las acciones del estado 1 y los estados a los que se llega (estados 2 y 3) se añaden a mano (*hard-coded*), no por inferencia.

Dado que el objetivo es observar hasta dónde llegan los modelos cuando se trata de maximizar la recompensa y explicar el comportamiento humano, optamos por usar ambos modelos base

(MF y MB) para así comparar los resultados obtenidos en ambos. Además de estos dos modelos básicos, creamos tres modelos más para nuestro análisis. El primero consiste en un modelo híbrido basado en la unión de los dos anteriores. Pensamos que en principio este modelo debería tener un comportamiento al menos igual o incluso mejor que el mejor de los iniciales. Los otros dos modelos serán, un modelo meta-model-free (meta-MF) y otro meta-model-based (meta-MB), basados en MF y MB respectivamente, pero cuyo ratio de aprendizaje, que es el parámetro que indica el porcentaje de cambio con el que se actualizan en cada iteración, es dinámico en vez de estático como en el resto de los modelos.

Estos cinco modelos generados para el trabajo los describiremos en mayor profundidad en las siguientes secciones. Finalmente, se compararán los resultados obtenidos para cada objetivo con cada uno de los modelos, analizando su comportamiento.

4. Descripción detallada de la solución adoptada

En esta sección exponemos detalladamente la metodología seguida para la implementación de la solución adoptada, que es el uso de algoritmos *SARSA* (*State-Action-Reward-State-Action*) y *meta Q-Learning*.

Para ello, comenzamos con el desarrollo del método de adquisición de los datos, así como sus características. Seguiremos con la exposición de los modelos en sí y concluiremos con la descripción de los procesos de los dos objetivos, es decir el proceso de maximización de la recompensa y el proceso de imitación del comportamiento humano.

Una vez obtenidos los resultados, en función del objetivo, procedemos a su análisis para comprobar la habilidad de cada uno de los modelos en el logro una mayor recompensa (4.3 Maximización de la recompensa) y la capacidad de explicar el comportamiento humano (4.4 Comprensión del comportamiento humano). Además, estudiaremos para cada caso el modelo más apto.

4.1 Adquisición de datos

Partimos del conjunto de datos resultantes del estudio de Decker et al., 2016 [6], que se encuentran divididos en una serie de documentos en formato “csv”, los cuales incluyen datos sobre los sujetos del estudio, las elecciones realizadas por los sujetos y la probabilidad de recompensa de cada elección final.

Para este trabajo, los archivos de interés serán aquellos que contengan la información necesaria para la realización de los objetivos y que mostramos en la Figura 3. Por un lado, el archivo de Decker_choices_for_RL.csv (Fig. 3A) nos proporciona los siguientes datos:

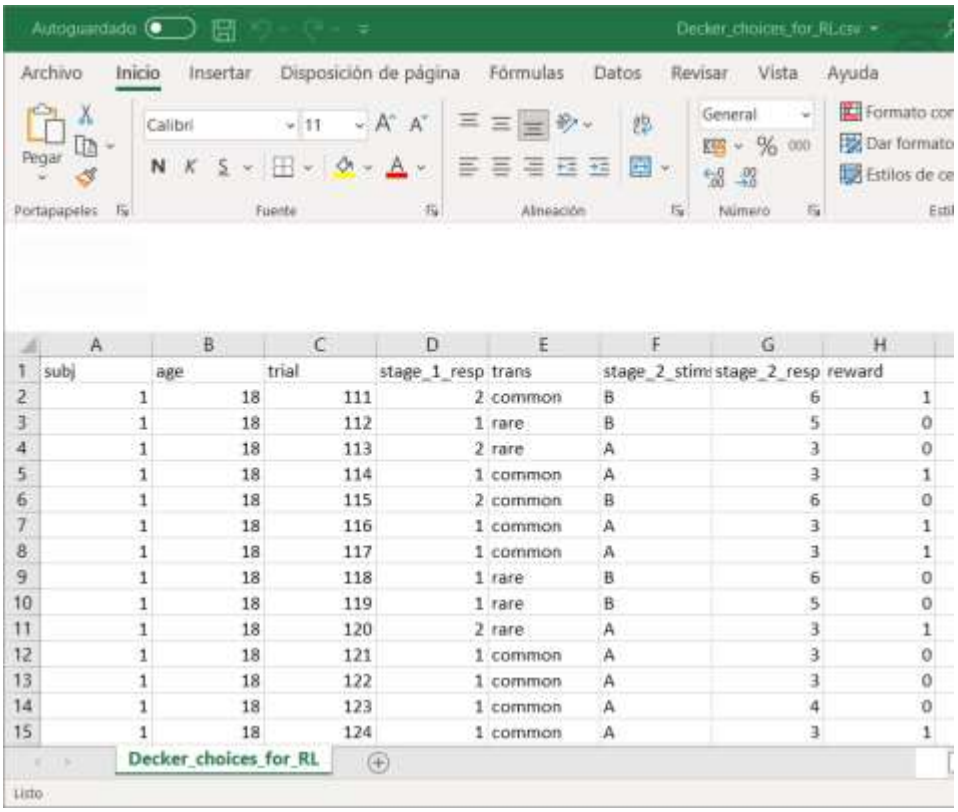
- **subj.** Proporciona el número que identifica al sujeto (1-59)
- **age.** Informa de la edad del sujeto.
- **trial.** Informa del número de ensayo.
- **stage_1_resp.** Informa de la acción realizada en la etapa 1 (1: nave azul; 2: nave verde).
- **trans.** Informa del tipo de transición de la etapa 1 (common: la transición con 70%; rare: la transición con 30%).
- **stage_2_stims.** Informa del estado en la etapa 2 (A: planeta rojo; B: planeta morado)
- **stage_2_resp.** Informa de la acción realizada en la etapa 2 (3: alienígena estrecho rojo; 4: alienígena ancho rojo; 5: alienígena estrecho morado; 6: alienígena ancho morado).
- **reward.** Informa si se consigue recompensa o no con la decisión final (0: No hay recompensa; 1: Hay recompensa)

Por otro lado, masterprob4.csv contiene las probabilidades de recompensa para cada decisión final (alienígena), de manera que cada columna identifica a un alienígena (Fig. 3B):

- **Columna 0:** alienígena estrecho rojo.
- **Columna 1:** alienígena ancho rojo.
- **Columna 2:** alienígena estrecho morado.
- **Columna 3:** alienígena ancho morado.

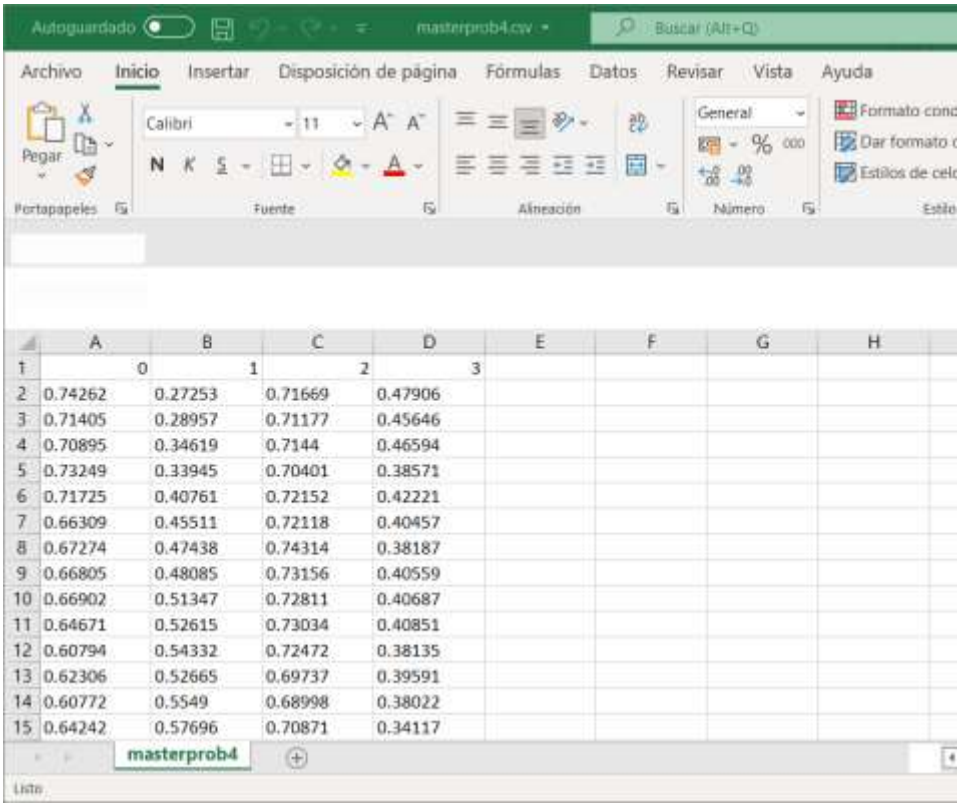
Como elegimos utilizar Matlab, la información de cada documento es guardada en formato MAT, ya que es el formato usado por el programa para guardar el workspace.

A)



	A	B	C	D	E	F	G	H
1	subj	age	trial	stage_1_resp	trans	stage_2_stim	stage_2_resp	reward
2	1	18	111	2	common	B	6	1
3	1	18	112	1	rare	B	5	0
4	1	18	113	2	rare	A	3	0
5	1	18	114	1	common	A	3	1
6	1	18	115	2	common	B	6	0
7	1	18	116	1	common	A	3	1
8	1	18	117	1	common	A	3	1
9	1	18	118	1	rare	B	6	0
10	1	18	119	1	rare	B	5	0
11	1	18	120	2	rare	A	3	1
12	1	18	121	1	common	A	3	0
13	1	18	122	1	common	A	3	0
14	1	18	123	1	common	A	4	0
15	1	18	124	1	common	A	3	1

B)



	A	B	C	D	E	F	G	H
1	0	1	2	3				
2	0.74262	0.27253	0.71669	0.47906				
3	0.71405	0.28957	0.71177	0.45646				
4	0.70895	0.34619	0.7144	0.46594				
5	0.73249	0.33945	0.70401	0.38571				
6	0.71725	0.40761	0.72152	0.42221				
7	0.66309	0.45511	0.72118	0.40457				
8	0.67274	0.47438	0.74314	0.38187				
9	0.66805	0.48085	0.73156	0.40559				
10	0.66902	0.51347	0.72811	0.40687				
11	0.64671	0.52615	0.73034	0.40851				
12	0.60794	0.54332	0.72472	0.38135				
13	0.62306	0.52665	0.69737	0.39591				
14	0.60772	0.5549	0.68998	0.38022				
15	0.64242	0.57696	0.70871	0.34117				

Fig. 3. Datos contenidos en el archivo Decker_choices_for_RL.csv. B) Datos contenidos en el archivo masterprob4.csv

4.2 Creación de los modelos

Como hemos comentado anteriormente, en este trabajo vamos a comparar cinco modelos de aprendizaje por refuerzo: model-free, model-based, híbrido, meta-model-free y meta-model-based. Los tres primeros basados en los utilizados por Otto et al., 2013 [7], mientras que los dos últimos serán una modificación del MF y el MB.

En esta subsección procedemos a exponer el funcionamiento de los modelos desde el más simple al más complejo.

4.2.1 Model-Free

Como ya hemos expuesto en la sección 1.2 Antecedentes, en este algoritmo model-free todo el peso de la elección de una acción recae en la recompensa obtenida. Para construirlo, nos basamos en el utilizado por Otto et al., 2013 [7] que es un algoritmo MF de diferencia temporal *SARSA* (λ).

El algoritmo *SARSA* actualiza de forma incremental un valor fijo para la elección de una acción de acuerdo con el historial de recompensas, asignando a continuación las probabilidades de escoger cada acción en cada estado.

En la práctica, el código consta de dos matrices 3x2, Q y P. La primera corresponde al valor a actualizar por el algoritmo y la segunda a las probabilidades de elección. En ambas matrices las filas representan los estados, mientras que las columnas representan la acción de cada estado. Al iniciar el proceso de optimización del modelo, asignamos el valor inicial de 0.5 a todas las posiciones de Q y P, por lo que tanto la función de valor de cada acción como su probabilidad de elección, son la misma para todas las acciones (0.5).

A continuación, entra en un bucle en el que cada iteración corresponde a un trial. El primer paso de cada iteración consiste en dar las correspondientes probabilidades a cada acción de cada etapa. Esta probabilidad viene dada en función de la matriz Q por la expresión:

$$\mathrm{P(state)} = \frac{\exp\left[\beta(\mathrm{state})\cdot Q(\mathrm{state,action}) + p\cdot \mathrm{rep(state,action)}\right]}{\sum_{action=1}^{2} \exp\left[\mathrm{beta(state)}\cdot Q(\mathrm{state,action}) + p\cdot \mathrm{rep(state,action)}\right]} \quad \textbf{(1)}$$

Donde p es el parámetro de "adherencia", que refleja la perseverancia en la acción de primera etapa (p >0) o la alternancia (p <0), *rep(:,:)* se define como 1 para una acción de primera etapa que repite la acción del ensayo anterior y beta (β) es un parámetro que permite ajustar el efecto de la función de valor en el cálculo de las probabilidades. Así, la repetibilidad de cada acción viene dada por el producto $p \cdot rep(:,:)$.

Tras escoger las acciones de las dos etapas, ya sea mediante los datos de los sujetos o en función de las probabilidades, el siguiente paso es actualizar la Q de la acción realizada en cada una de las etapas.

En la primera etapa la función que define su comportamiento es la siguiente:

$$Q(1, action) = Q(1, action) + \alpha\delta \quad (\mathbf{2})$$

donde alfa (α) es el parámetro del ratio de aprendizaje y

$$\delta = Q(state\_, action\_) - Q(1, action) \quad (\mathbf{3})$$

Indica el error de predicción de recompensa (EPR). Así, la función de valor de acción de estado Q (1, action) la actualizamos para la acción realizada del estado 1 (recordamos que hay 2 posibles acciones en cada estado) de acuerdo con la propia Q de la acción escogida y el producto de α con δ (delta). En estas expresiones estamos denotando por Q(1, action) la Q de la acción escogida en la primera etapa y por Q(state_, action_) la Q de la acción escogida en la segunda etapa, siendo *state_ y action_* el estado alcanzado y la acción realizada en la segunda etapa, respectivamente.

En esta segunda etapa la actualización de la Q la realizamos mediante la función:

$$Q(state\_, action\_) = Q(state\_, action\_) + \alpha\delta \quad (\mathbf{4})$$

pero ahora δ la obtenemos en función de la Q y la recompensa:

$$\delta = reward - Q(state\_, action\_) \quad (\mathbf{5})$$

El state_ puede ser el estado 2 o el 3, ya que la Etapa 2 consta de dos estados a los que se puede llegar mediante la elección de primera etapa. Además, debido a que la Etapa 1 no recibe ninguna recompensa de manera directa, tras actualizar el Q de la Etapa 2, realizamos una actualización extra de la Q de la acción realizada en el Estado 1, de acuerdo con:

$$Q(1, action) = Q(1, action) + \alpha\lambda\delta \quad (\mathbf{6})$$

donde añadimos la EPR multiplicada por el trazado de elegibilidad, lambda, para que la recompensa que logramos en la Etapa 2 tenga mayor efecto en la Etapa 1.

Como podemos comprobar, hay 4 parámetros a optimizar: α, β, λ y p. Estos parámetros los optimizamos mediante la función *rmsearch* que hace uso de BADS. Además, realizamos estas optimizaciones para dos configuraciones distintas del uso de β, una en la que optimizados una única β para todos los estados y otra en la que optimizamos una β distinta para cada estado, ya

que con una β para cada estado el grado de adaptación del modelo a la tarea es mayor. De esta manera, podemos comparar los resultados y discernir si compensa añadir mayor complejidad al modelo al aumentar el número de betas a 3, una por estado, con el correspondiente aumento de coste computacional, o si compensa mantener el modelo más simple.

4.2.2 Model-Based

El algoritmo MB utiliza un modelo para elegir la mejor acción. Específicamente, en este trabajo utilizamos un MB de "arboles de búsqueda" (cálculo explícito de la ecuación de Bellman), que representa todas las opciones de elección posibles y los resultados asociados (4), ya que existen dos posibles acciones en la Etapa 1 y se conocen las probabilidades de cada acción de alcanzar dos posibles estados.

Este modelo, por tanto, es muy similar al anterior, pero difiere en la función de aprendizaje de la primera etapa, que tiene en cuenta la estructura de probabilidad de transición de 70/30 y calcula los valores de acción de estado acumulados de todos los resultados posibles. Como tal, en el algoritmo MB actualizamos los valores de acción de la primera etapa de acuerdo con lo siguiente:

$$En\ caso\ de\ escoger\ la\ accion\ 1\ \rightarrow 0.7 \cdot \max\big(Q(2)\big) + 0.3 \cdot \max\big(Q(3)\big) \quad (\mathbf{7})$$

$$En\ caso\ de\ escoger\ la\ accion\ 2\ \rightarrow 0.3 \cdot \max\big(Q(2)\big) + 0.7 \cdot \max\big(Q(3)\big) \quad (\mathbf{8})$$

Aunque la estimación del valor de acción de la segunda etapa la realizamos de la misma manera que el algoritmo MF, como la Etapa 1 tiene en cuenta los valores de Q de la segunda etapa, ya no es necesaria la actualización extra de la Etapa 1 realizada en el model-free.

4.2.3 Modelo híbrido

Ahora nos planteamos la realización de un modelo híbrido que considere ambos, el MF y el MB. Este tipo de modelos ya ha sido utilizado por Otto et al., 2013 [6], en ese trabajo los autores utilizaban la combinación de los valores de acción de MF y MB para calcular las probabilidades. Sin embargo, en este trabajo, nosotros vamos a desarrollar un modelo híbrido distinto al presentado por estos autores. Nuestro modelo simplifica la optimización, ya que no se necesita ejecutar el MB y el MF a la vez, sino que va a depender de los resultados obtenidos de cada uno de los modelos calculados previamente.

Para ello, usamos las P de cada uno de los dos modelos MF y MB, calculados previamente, y mediante un parámetro de peso ω, asignamos el valor de las P híbridas siguiendo la siguiente ecuación:

$$P = \omega \cdot \mathrm{P}_{MF} + (1 - \omega) \cdot \mathrm{P}_{MB} \quad (\mathbf{9})$$

4.2.4 Modelos Meta

La diferencia entre los modelos Meta-MF y Meta-MB respecto a los modelos MF y MB, se encuentra en que el ratio de aprendizaje (*learning rate*), alfa, de estos últimos es estático mientras que el de los modelos con meta aprendizaje es dinámico. De manera que, tras actualizar los valores de las Q escogidas en la primera y la segunda etapa, en estos modelos añadimos una actualización del *learning rate*.

Por lo tanto, tenemos una incertidumbre, unReduct (*uncertainty reduction*), que calculamos a partir de cuánto se aleje del centro (0.5) la variable Q. Así, como los valores de las entradas de Q están entre 0 y 1, el valor absoluto del doble de la resta de Q menos el valor central (0.5) da como resultado un numero entre -1 y 1, al cual, se le aplica el valor absoluto. Esto se debe a que tanto la correlación positiva como negativa entre la acción y la recompensa son informativas. Cuando la Q tiene el mínimo valor (0.0) la resta multiplicada por dos da -1, siendo así una acción que sistemáticamente no proporciona recompensa. Mientras que cuando el valor de Q es máximo (1.0) la resta multiplicada por dos da 1, lo que quiere decir que sistemáticamente se consigue recompensa con esa acción y, por lo tanto, ambos extremos informan de si hay o no recompensa.

$$unReduct = abs\,(2 \cdot (Q(state, action) - 0.5)) \quad (\mathbf{10})$$

Teniendo en cuenta lo mencionado hasta ahora, y tal como se puede comprobar en la expresión (10), en el caso de que la Q tenga un valor intermedio de 0.5 el resultado sería 0, ya que dicho valor implica que no hay una correlación entre la acción y la recompensa, por lo que genera una incertidumbre y no proporciona información. Por lo tanto, este valor de incertidumbre, que se actualiza con cada Q utilizada, junto al parámetro constante (μ) y el *learning rate* (α) del trial anterior, los usamos para actualizar el *learning rate* actual:

$$\alpha = \alpha + \mu \cdot (uncReduct - \alpha) \quad (\mathbf{11})$$

Consiguientemente, con el *learning rate* dinámico, buscamos priorizar las características que sean más predictivas y penalizar las que produzcan una mayor incertidumbre. Con lo que, cuanto más se alejan los valores de Q de 0.5, lo cual implica más información, el *unReduct*

aumenta provocando a su vez que el *learning rate* aumente. Sin embargo, cuanto más se acercan a 0.5, hay mayor incertidumbre y el *unReduct* disminuye haciendo a su vez que el *learning rate* lo haga. Como resultado, con el *learning rate* dinámico conseguimos mejorar la relación entre señal, lo que contiene información relevante para la toma de decisión, y ruido, los valores fluctuantes no sistemáticos.

4.3 Maximización de la recompensa

La obtención de la recompensa máxima nos permite comparar qué modelos resultan más eficientes. En este sentido, la maximización de la recompensa por parte de los agentes la separamos en dos partes.

Primero, para comprobar el correcto funcionamiento de los modelos y no sufrir posibles interferencias en la optimización, debido a las probabilidades cambiantes de recompensa, optamos por sustituir las probabilidades de recompensa de la última acción por unas probabilidades constantes. En segundo lugar, una vez comprobamos el correcto funcionamiento de los modelos en el caso anterior, usamos las probabilidades de los datos originales, los cuales van variando con una lenta deriva, para poder comprobar su desempeño con las probabilidades reales.

4.3.1 Probabilidades constantes

En este caso decidimos sustituir la probabilidad de recompensa de cada elección final, para pasar de tener unas probabilidades dinámicas a unas probabilidades estáticas y, de esta manera, seamos capaces de comprobar el correcto funcionamiento de los modelos sin interferencias debidas a la estocasticidad de las probabilidades de recompensa. De manera que las probabilidades ahora en cada *trial* serán:

- **Elección final 1**: 0.2% de probabilidades de recompensa.
- **Elección final 2**: 0.8% de probabilidades de recompensa.
- **Elección final 3**: 0.6% de probabilidades de recompensa.
- **Elección final 4**: 0.4% de probabilidades de recompensa.

Todos los modelos comienzan asignando los valores de los parámetros a optimizar. Tras esto, se inicia el bucle en el que cada *trial* corresponde a una iteración. Al inicio del trial concedemos a las diferentes P de cada acción sus valores en función de las Q actuales. A partir de la asignación de estos valores de P, el procedimiento de cada uno de los modelos presenta divergencias.

Model-Free y Model-Based: con las probabilidades estáticas mencionadas, mediante softmax, una función que permite seleccionar aleatoriamente un parámetro entre varios teniendo en cuenta sus probabilidades de selección, conseguimos la acción de primera etapa. A continuación, conociendo la acción de primera etapa y las probabilidades de alcanzar los distintos estados de la segunda, llegamos a un estado de la segunda etapa y finalmente, generamos la acción de segunda etapa de acuerdo con sus probabilidades de elección.

Tras ello, comprobamos si hay recompensa o no en función de la probabilidad de recompensa de la última elección realizada, basándonos en los porcentajes de probabilidad de recompensa ya mencionados. En cada etapa, actualizamos los valores de la Q correspondiente a la acción escogida en función de las ecuaciones del modelo. Además, el modelo MF, actualiza en mayor medida la Q de la primera etapa al finalizar ambas, de acuerdo con la ecuación (6) presentada en la sección 4.2.1 Model-Free, en la que a la Q de la acción realizada de primera etapa se le suma el EPR de la segunda etapa multiplicado por λ.

Híbrido: en este caso, como el modelo depende de los resultados obtenidos en los modelos MF y MB, el valor de las P lo asignamos en función de las P guardadas de los modelos MF y MB en cada trial, de esta manera el peso de cada modelo depende de ω, la cual optimizamos para encontrar el mejor valor posible en cada caso (comportamiento humano vs. maximizar el reward). Tras esto, realizamos los mismos pasos que en los modelos anteriores, pero sin la actualización de las Q, ya que este modelo trabaja sobre los modelos MF y MB que ya han sido optimizados.

Meta-Model-Free y Meta-Model-Based: los pasos que seguimos son los mismos que los de los modelos MF y MB respectivamente, pero en este caso, tras actualizar Q realizamos una actualización del *learning rate*.

Cuando con uno de los modelos terminamos los pasos descritos anteriormente, guardamos en la variable optMetric la media de la recompensa obtenida que es la variable objetivo a maximizar y, finalmente, guardamos los datos de interés, que son los mismos para cada modelo (Fig. 4):

- **target**. El objetivo del modelo (Human Behaviour / Reward).
- **trialsPerSubj**. Numero de *trials* del agente.
- **set_rep**. si se ha repetido o no la elección de primera etapa en cada *trial*.
- **Q**. Las Q finales.
- **set_Q**. Las Q registradas en cada *trial*.
- **Q_chosen**. La Q escogida en la primera etapa y en la segunda etapa de cada *trial*.
- **P**. Las probabilidades finales de cada acción.
- **set_P**. Las probabilidades registradas en cada *trial*.
- **P_chosen**. La P escogida en la primera etapa y en la segunda etapa de cada *trial*.
- **stage_1_resp**. La acción escogida de primera etapa de cada trial.
- **stage_2_stims**. El estado de segunda etapa de cada trial.
- **stage_2_resp**. La acción escogida de segunda etapa de cada trial.
- **ret_reward**. El reward obtenido en cada *trial*.
- **cum_reward**. el reward acumulado en cada *trial*.
- **Avg_reward**. La media de recompensa obtenida.
- **learning_rate**. El *learning rate* de cada *trial*.
- **optMetric**. El valor objetivo a maximizar.
- **model_tag**. El modelo usado ('model-free', 'model-based', 'hybrid', 'meta_learning_mf', 'meta_learning_mb').
- **f_xopt**. Los valores de las variables a optimizar.
- **f_opt**. El valor de la variable objetivo tras la optimización.

Con esta configuración, si los modelos funcionan correctamente, los agentes tenderán a seleccionar la elección 2, ya que tiene la mayor probabilidad (0.8), y en menor medida la opción 3 con la segunda mayor probabilidad (0.6). Una vez comprobamos que los modelos se comportan correctamente, procedemos a realizar los modelos con las probabilidades dinámicas.

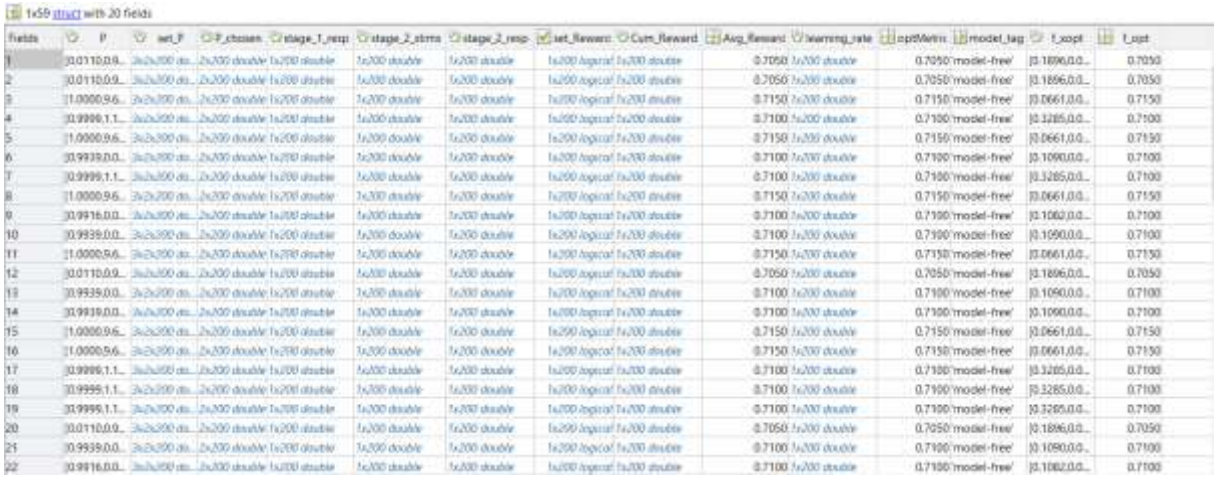


Fig. 4. Sección de la tabla generada con los datos obtenidos del modelo MF.

4.3.2 Probabilidades dinámicas

El proceso para maximizar la recompensa con las probabilidades dinámicas de elección final es el mismo que seguimos con las probabilidades estáticas, con la diferencia de que, en este caso, las probabilidades de cada elección final de cada *trial* son dinámicas y no estáticas, con lo que la probabilidad de recompensa de cada una de las acciones finales varia en cada *trial* (Fig. 5).

Para el proceso, empleamos los valores dinámicos contenidos en el archivo *masterprob4.mat,* de manera que cada vez que realizamos una elección final, le asignamos la probabilidad de recompensa correspondiente en función de qué elección se lleve a cabo y del *trial*. Tras ello, empleamos la función de *rand* con la que decidimos si hay o no recompensa, para lo cual, con esta función generamos un numero pseudoaleatorio entre 0.0 y 1.0, con el que hacemos una comparación con la probabilidad de recompensa de la última acción elegida, si el número es menor a la probabilidad hay recompensa, sino no se obtiene recompensa.

Fig. 5. Panales extraídos de la Fig 2. Representan la probabilidad de recompensa dinámica de cada una de las acciones finales

4.4 Comprensión del comportamiento humano

Para la comprensión del comportamiento humano el objetivo es maximizar las probabilidades de realizar las mismas acciones llevadas a cabo por los sujetos. Por ello, lo que pretendemos es inferir cómo los humanos tomamos nuestras decisiones, para lo cual los modelos de aprendizaje por refuerzo ofrecen una aproximación mecanicista al comportamiento: los valores "Q(state,action)" y su dinámica, las probabilidades de las transiciones entre etapas, la combinación entre mecanismos MF y MB, o una posible dinámica del learning rate.

Por tanto, procedemos a medir qué valores y probabilidades hay de elegir las mismas acciones que escogieron los sujetos bajo sus mismas transiciones entre estados, optimizando así el comportamiento de los modelos. Tras ello, comparamos los diferentes modelos para hallar aquellos que son más capaces de reproducir el comportamiento humano.

De la misma manera que en el caso de maximizar la recompensa, en el proceso de optimización, todos los modelos comienzan asignando los valores de los parámetros a optimizar, ya mencionados previamente (4.2 Creación de modelos). Tras esto, iniciamos el bucle para correr cada uno de los *trials*, de manera que, para cada iteración, es decir en cada *trial*, comenzamos concediendo a las diferentes P de cada acción sus valores en función de las Q actuales, extrayendo los estados y las acciones de la tabla DeckerchoiceforRL. A partir de la asignación de estos valores P, el procedimiento de cada uno de los modelos presenta divergencias.

Model-Free y Model-Based: del conjunto de datos con los estados y acciones de los sujetos para cada ensayo, extraemos los estados y las acciones de cada etapa en cada *trial*. Con estos datos extraídos, en función de la ecuación de cada uno de los modelos y de las acciones escogidas, actualizamos el valor de la Q en cada etapa. Además, en el modelo MF actualizamos en mayor medida la Q de la primera etapa al finalizar ambas etapas (5.2.1, ecuación 6).

Híbrido: en este caso, partimos de los modelos ya optimizados de MF y MB, como hemos registrado para cada *trial* cuáles han sido las P de cada acción, el valor de las P del modelo híbrido lo asignamos en función de las P guardadas de los modelos MF y MB en cada trial. Para ello, el peso de cada uno de los modelos depende de una variable llamada ω. Es por esto que realizamos los mismos pasos que en los modelos anteriores, pero sin la actualización de las Q, ya que no se necesita calcular las P mediante las Q.

Meta-MF y Meta-MB: los pasos son los mismos que en los modelos MF y MB respectivamente, pero tras actualizar Q se realiza una actualización del *learning rate*.

Cuando terminamos los pasos correspondientes con uno de los modelos, guardamos en la variable optMetric la media de la media del logaritmo de las P escogidas en cada *trial*, la cual es la variable objetivo a maximizar. Finalmente, guardamos los datos de interés:

- **target**. El objetivo del modelo (Human Behaviour / Reward).
- **trialsPerSubj**. Numero de *trials* del agente.
- **set_rep**. si se ha repetido o no la elección de primera etapa en cada *trial*.
- **Q**. Las Q finales.
- **set_Q**. Las Q registradas en cada *trial*.
- **Q_chosen**. La Q escogida en la primera etapa y en la segunda etapa de cada *trial*.
- **P**. Las probabilidades finales de cada acción.
- **set_P**. Las probabilidades registradas en cada *trial*.
- **P_chosen**. La P escogida en la primera etapa y en la segunda etapa de cada *trial*.
- **stage_1_resp**. La acción escogida de primera etapa de cada trial.
- **stage_2_stims**. El estado de segunda etapa de cada trial.
- **stage_2_resp**. La acción escogida de segunda etapa de cada trial.
- **ret_reward**. El reward obtenido en cada *trial*.
- **cum_reward**. el reward acumulado en cada *trial*.
- **Avg_reward**. La media de recompensa obtenida.
- **optMetric**. El valor objetivo a maximizar.
- **model_tag**. El modelo usado ('model-free', 'model-based', 'hybrid', 'meta_learning_mf', 'meta_learning_mb').
- **f_xopt**. Los valores de las variables a optimizar.
- **f_opt**. El valor de la variable objetivo tras la optimización.

5. Justificación de la solución adoptada

5.1 Resultados

Una vez obtenidos los resultados, comprobamos qué modelo explica mejor el comportamiento humano en función de las probabilidades de escoger por parte de los agentes las mismas acciones que los sujetos. Además de realizar la comparación con los datos globales de los sujetos, también dividimos los resultados en los distintos grupos de edad del estudio (niño, adolescente, adulto) para comprobar la capacidad de explicar el comportamiento humano por parte de los modelos dependiendo de la edad.

5.1.1 Resultados en la maximización de la recompensa

En la comprobación del correcto funcionamiento de los modelos, mediante el uso de elecciones finales con una probabilidad estática de recompensa, obtenemos los diagramas de la frecuencia con la que los modelos han elegido una acción en cada uno de los tres estados posibles (Fig. 6).

Los resultados obtenidos son los esperados y similares para todos los modelos. En el estado uno la acción seleccionada con mayor frecuencia es la primera (el cohete azul) la cual tiene un 70% de probabilidades de alcanzar el Estado 2 (planeta rojo). A su vez en el estado 2 la acción más repetida es la segunda (alienígena rojo ancho). Esto se debe a que la elección final con mayor probabilidad de recompensa es la segunda acción del estado 2 (80%). Además, en los casos en los que se ha alcanzado el estado 3 (planeta morado) la acción 2 (alienígena morado ancho) es mucho más frecuente que la acción 2 (alienígena morado estrecho), ya que para la acción 1 hay un 40% de probabilidad de recompensa y para la acción 2 un 60%.

Además, el modelo MB parece seleccionar mejor la opción correcta en el Estado 1 frente al modelo MF. Nada sorprendente puesto que la cualidad característica del modelo MB es conocer el funcionamiento del entorno. En este caso, el modelo conoce las probabilidades que tiene cada acción del Estado 1 de llegar al estado 2 o 3. Identificando con mayor rapidez que la opción correcta para llegar a la mejor elección final (estado 2, acción 2) es seleccionar la acción 1 del estado 1. Asimismo, la diferencia en recompensa final no es significativa y viene dada por la estocasticidad del sistema.

Aunque el modelo MB transiciona mejor desde el estado 1 al estado más adecuado (el estado 2), la recompensa final es menor que en el MF debido a que, una vez superado el estado 1, ambos modelos son iguales, ya que hay una probabilidad de conseguir *reward* y el *reward* acumulado varia según la realización. Esto tambien se aprecia en la figura 8, donde MB funciona mejor que MF.

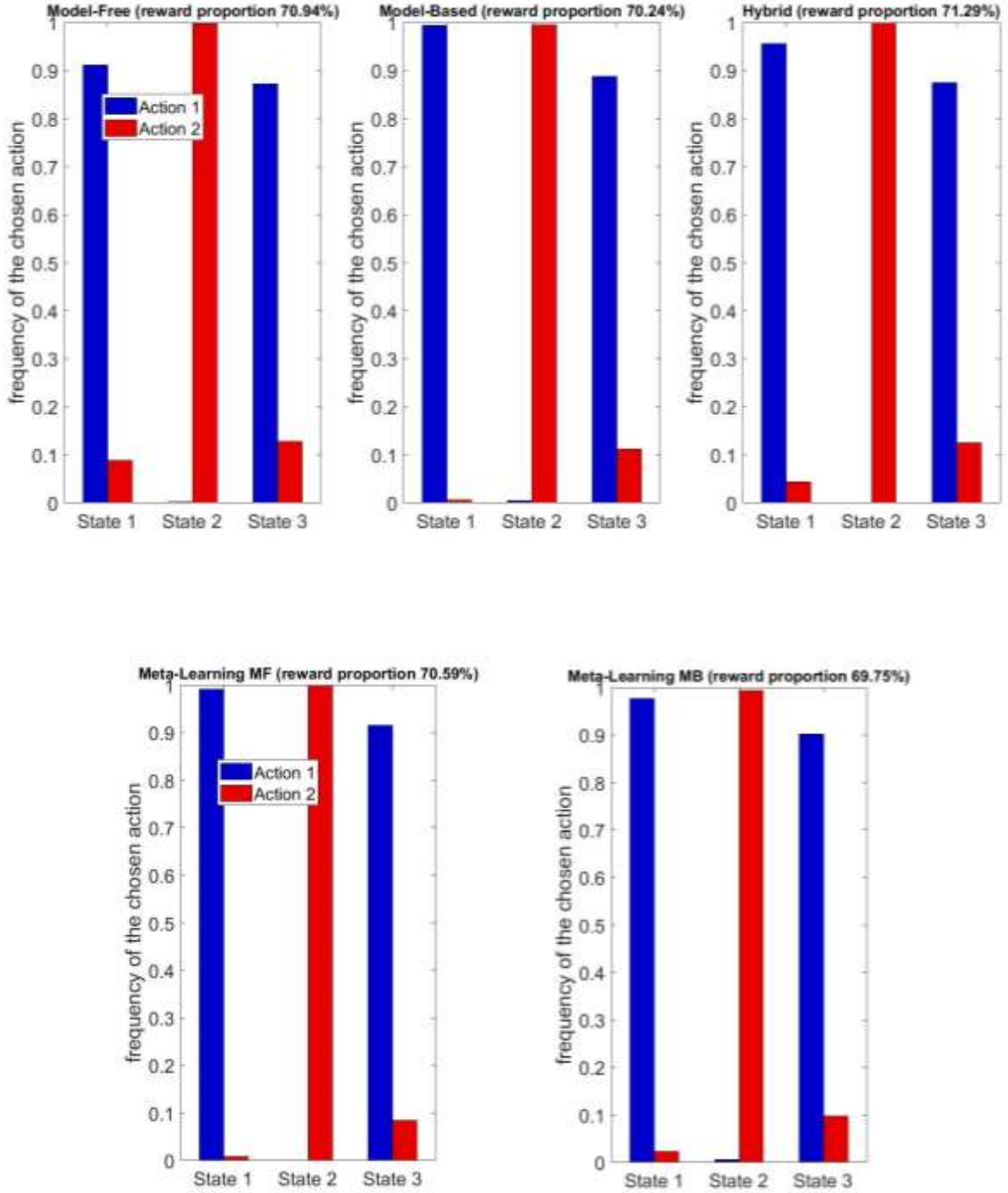


Fig. 6. Diagramas de barras de cada modelo con las probabilidades de recompensa estáticas y usando cada modelo una beta. Los diagramas muestran la frecuencia con la que se selecciona la acción 1 y 2 para cada estado, siendo está similar para todos los casos.

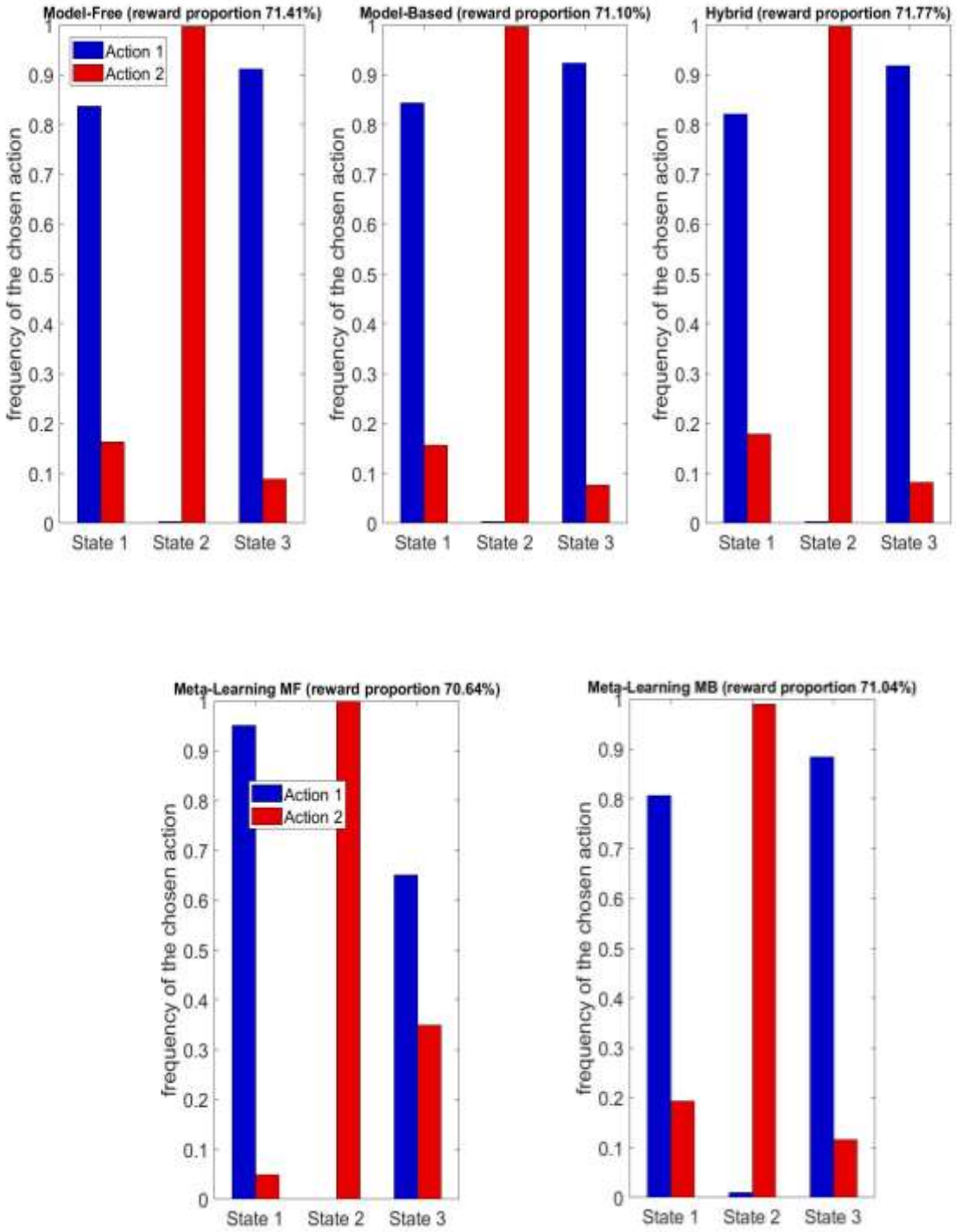


Fig. 7. Diagramas de barras de cada modelo con las probabilidades de recompensa estáticas y usando cada modelo tres betas. Los diagramas muestran la frecuencia con la que se selecciona la acción 1 y 2 para cada estado, siendo está similar para todos los casos.

Tras estos resultados, que confirman que los modelos realizan las acciones que deberían realizar, comprobamos el comportamiento de los modelos con las probabilidades reales del estudio. Para ello, hemos hallado la media de las probabilidades variables de recompensa de cada una de las elecciones finales, y así podamos tener una idea aproximada de cuáles deberían ser las mejores acciones. Estas medias son:

- Estado 2, Acción 1: 48,79%
- Estado 2, Acción 2: 59,85%
- Estado 3, Acción 1: 58,29%
- Estado 3, Acción 2: 37,83%

Los diagramas de barras muestran como para todos los modelos, ya sea con una o tres betas, la elección más frecuente en el Estado 2 es la Acción 2 y en el Estado 3 la Acción 1 (Fig. 8 y 9). En función a las medias de probabilidad de recompensa de cada acción final, estos resultados tienen sentido, ya que ambas acciones son las de mayor probabilidad en sus respectivos estados, por lo tanto, aun variando ligeramente las frecuencias entre modelos es lógico pensar que esas acciones serán las más frecuentes.

Sin embargo, con el Estado 1 hay una mayor disparidad de resultados. Esto puede deberse a que las acciones con mayor probabilidad de recompensa, dentro de los Estados 2 y 3, tienen un porcentaje muy parecido (Estado 2, Acción 2: 59,85%; Estado 3, Acción 1: 58,29%), por ello la diferencia de recompensa media será pequeña al realizar la Acción 1 o 2 del Estado 1, siempre que se escojan después las acciones con más probabilidad de recompensa en el Estado 2 y 3. A este hecho, se le añaden las interferencias de las transiciones raras de las acciones (30%) además de que, al variar las probabilidades de recompensa final en cada *trial* (Fig. 4), en ocasiones la Acción 1 del Estado 3 tiene mayor probabilidad de recompensa que la acción 2 del estado 2, inclinando la balanza de idoneidad hacia la elección de la acción 2 del Estado 1.

Aunque el comportamiento descrito es similar tanto para la optimización de una beta como para tres betas, se observa una ligera mejora en la recompensa media de todos los modelos cuando hacemos uso de tres betas (Fig. 8 y 9).

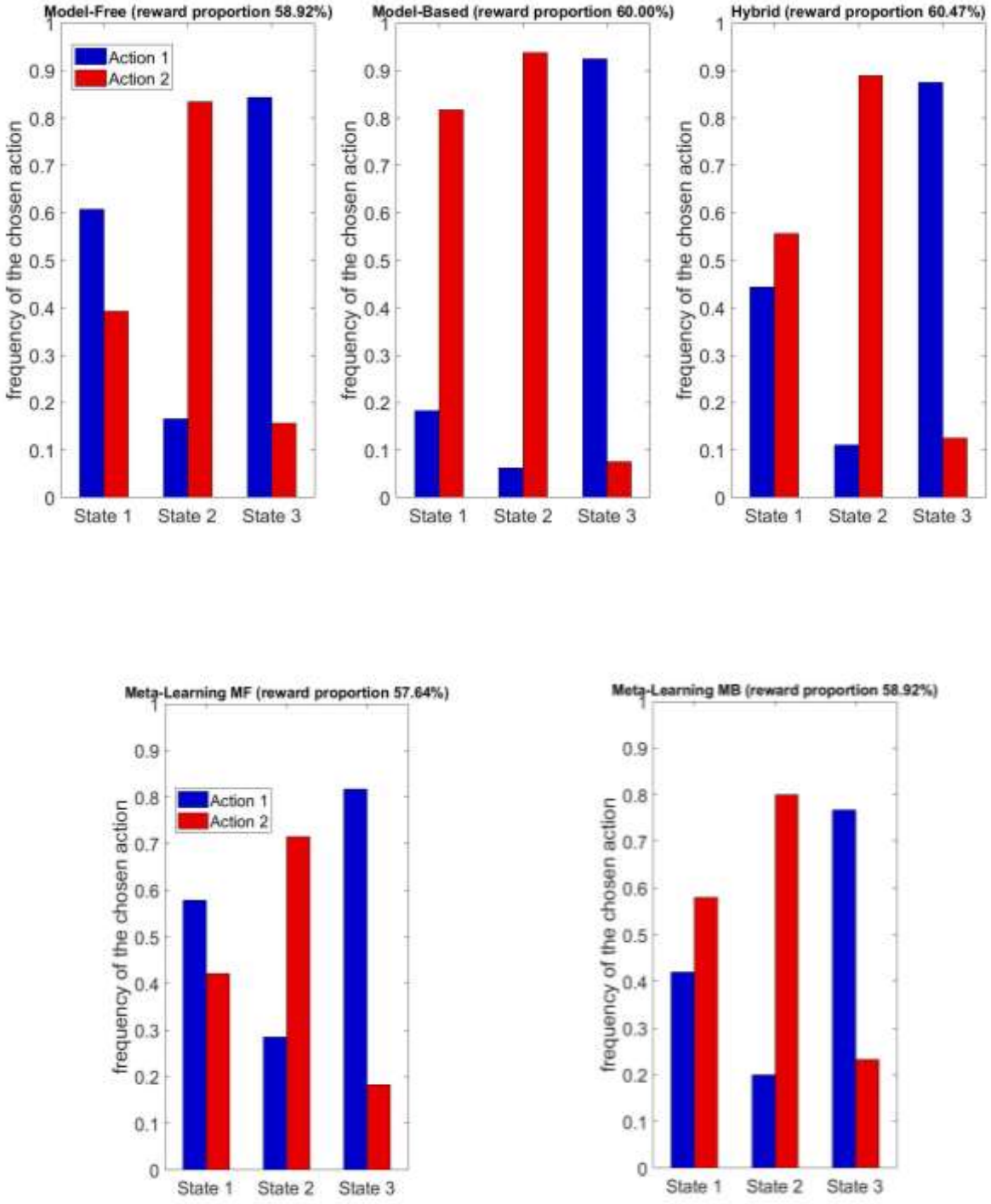


Fig. 8. Diagramas de barras de cada modelo con las probabilidades de recompensa reales y usando cada modelo una beta. Los diagramas muestran la frecuencia con la que se selecciona la acción 1 y 2 para cada estado.

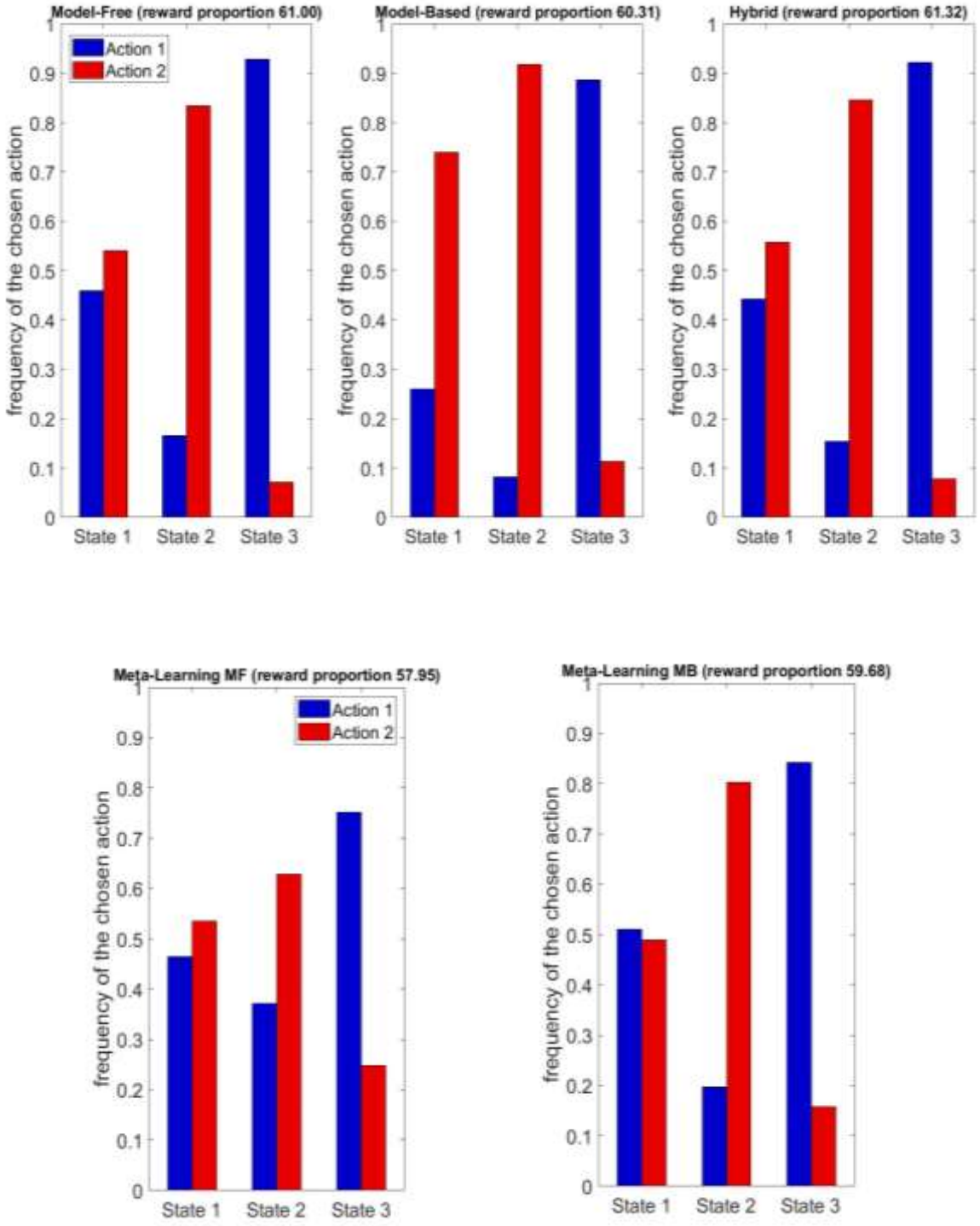


Fig. 9. Diagramas de barras de cada modelo con las probabilidades de recompensa reales y usando cada modelo tres betas. Los diagramas muestran la frecuencia con la que se selecciona la acción 1 y 2 para cada estado.

Los resultados del comportamiento de los diversos modelos los analizamos mediante graficas de densidad (Fig. 10 y 11) de la variable objetivo, representando en este caso la media de la recompensa obtenida a través de los 200 *trials*.

Los mejores resultados se han obtenido con el modelo híbrido (Fig. 10). Esto se debe a que este modelo surge de la unión ponderada de MF y MB, escogiendo las acciones para cada *trial* en función del mejor comportamiento de entre los dos modelos. Por lo tanto, los resultados del híbrido son como mínimo iguales al mejor modelo entre MF y MB.

El mejor rendimiento de MB frente a MF se explica debido a un mejor desempeño en el Estado 1, ya que, como hemos comentado, el modelo MB conoce las probabilidades de alcanzar los estados 2 y 3 en función de las acciones del Estado 1.

En cuanto a las opciones con una o tres betas, todos los modelos se ven ligeramente beneficiados cuando se hace uso de tres betas, siendo el caso MF el más beneficiado. El único resultado extraño está relacionado con MB y MF al usar tres betas. En este caso el MF se comporta ligeramente mejor que el MB, asimismo los resultados obtenidos con los modelos meta (Fig. 10) parecen indicarnos que funcionan peor que sus versiones sin actualización dinámica del *learning rate*.

Como no tiene sentido que el MB funcione peor que el MF y que los modelos meta funcionen peor que los demás modelos, ya que los modelos con más grados de libertad y mayor conocimiento de base siempre serán mejores, la razón para obtener estos resultados es la estocasticidad del problema y la dificultad de su optimización. La solución óptima es mucho más difícil a medida que crece la dimensión del espacio de búsqueda con el aumento del número de parámetros del modelo.

Adicionalmente, cuando comprobamos la recompensa media alcanzada por los sujetos, podemos apreciar que los modelos son capaces de mejorar la capacidad de las personas, consiguiendo más del 5% de recompensa media (Fig. 12).

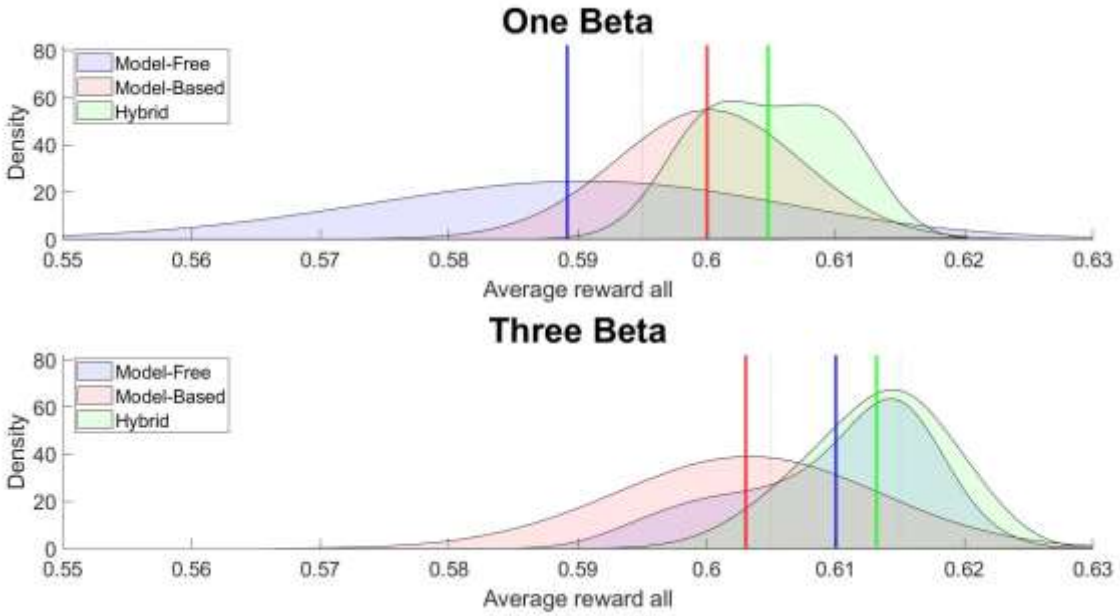
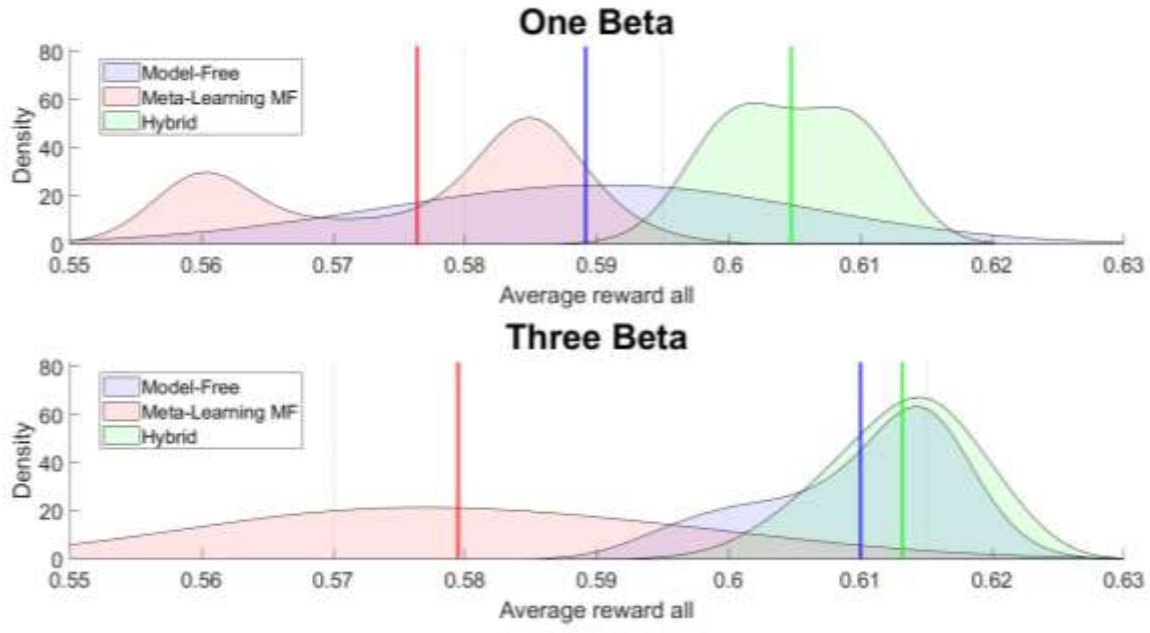


Fig. 10. Gráfica de densidad que compara la media de recompensa de los modelos MF, MB e híbrido. Con una beta la gráfica superior y con tres betas la inferior.

A)



B)

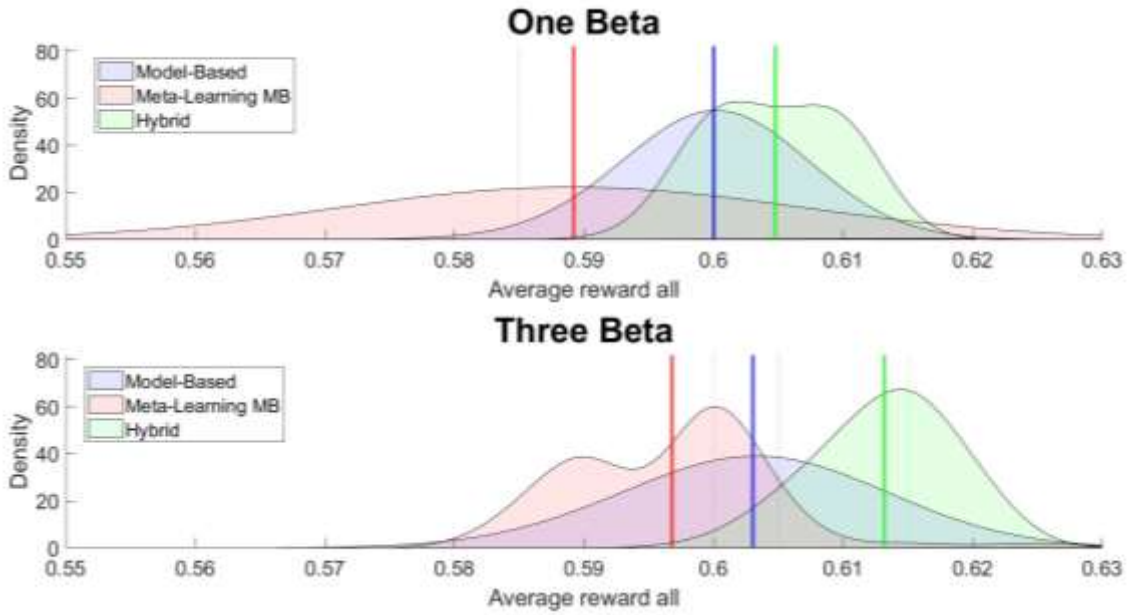


Fig. 11. A) Gráfica de densidad que compara la media de recompensa de los modelos MF, meta-MF e híbrido. Con una beta la gráfica superior y con tres betas la inferior. B) Grafica de densidad que compara la media de recompensa de los modelos MB, meta-MB e híbrido. Con una beta la gráfica superior y con tres betas la inferior.

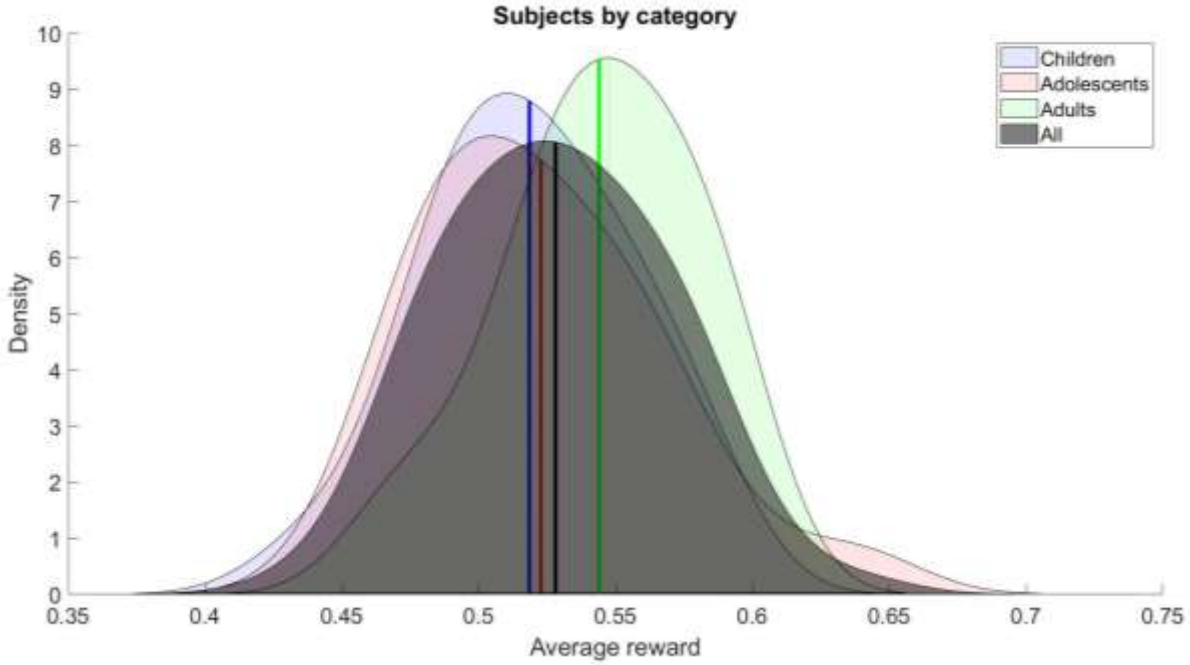


Fig. 12. Gráfica de densidad que compara la media de recompensa de los sujetos en su total y divididos por categorías de edad (niño, adulto, joven).

5.1.2 Resultados en la comprensión del comportamiento humano

Mediante la gráfica de densidades comprobamos el desempeño de cada modelo a la hora de explicar el comportamiento humano durante la toma decisiones. Estas graficas las realizamos sobre los datos del conjunto de sujetos y los datos de cada categoría de edad, ya que lo que buscamos es inferir mecanismos cerebrales mediante el uso del aprendizaje por refuerzo, por ejemplo, si los niños se comportan más como un modelo MF, los adultos más similares a uno MB y los adolescentes como un punto intermedio entre ambos modelos.

En el caso de buscar comprender el comportamiento humano, los resultados nos muestran que no hay diferencias destacables entre trabajar con una beta o con tres betas. Además, podemos apreciar que el modelo MF sirve mejor que el modelo MB para explicar las elecciones de los sujetos (Fig.13).

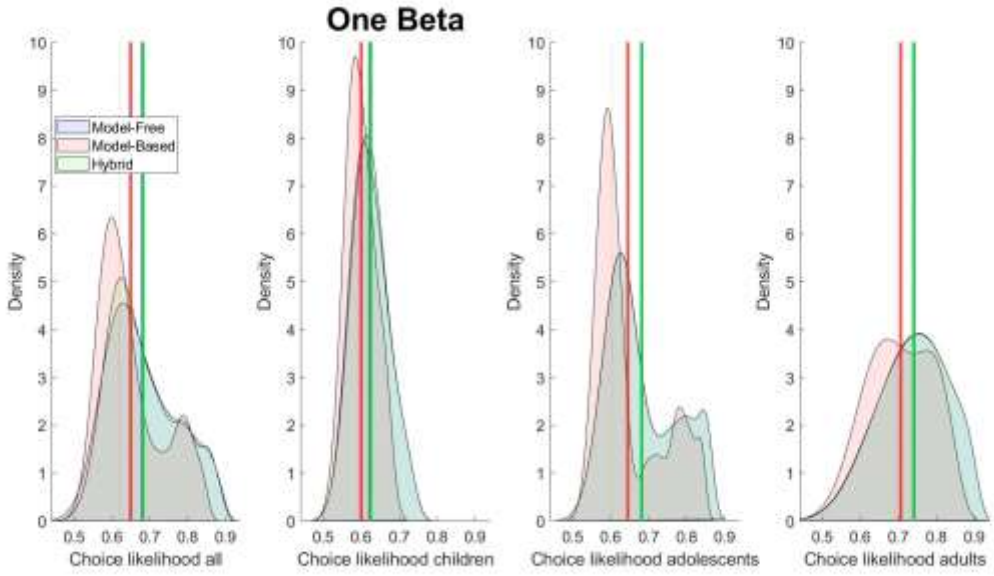
Pero, como era de esperar, el modelo híbrido rinde mejor que el Model-Free y el Model-Based, ya que, como se hemos dicho con anterioridad, este se comporta como mínimo como el mejor de los dos modelos. Cabe destacar que debido a la escala de las gráficas la línea que señala la media del modelo MF no se ve debido a que esta solapada por la del modelo híbrido. De todas formas, en caso de hacer un zoom a cada una de las gráficas, la media del modelo híbrido siempre es ligeramente mejor que el Model-Free (Fig. 13).

En los resultados comparativos de MF, MB e híbrido, también podemos extraer dos datos importantes. Por un lado, respecto a los modelos Model-Free y Model-Based, apreciamos una mejor explicación del comportamiento de los sujetos mediante el primero. Por otro lado, a mayor edad los resultados tienden a ser mejores en los tres modelos (MF, MB e híbrido).

Los modelos con *meta learning* sólo los analizamos de manera global, es decir, no los dividimos por categorías de edad, ya que solo queremos observar su comportamiento global frente a los demás modelos. En esta ocasión, observamos que los dos modelos con *meta learning* se comportan mejor que sus modelos básicos con el *learning rate* estático, sin embargo, al igual que pasaba previamente, no observamos diferencias con respecto al uso de una o tres betas (Fig. 14).

El meta-MF parece ser mejor que el hibrido y que su versión sin *learning rate* dinámico. Como el Model-Free es el modelo que mejor explica el comportamiento humano entre los dos modelos básicos y el hibrido solo es ligeramente mejor que MF, el rendimiento del modelo Model-Free con *meta learning* supone el mejor para el objetivo actual.

A)



B)

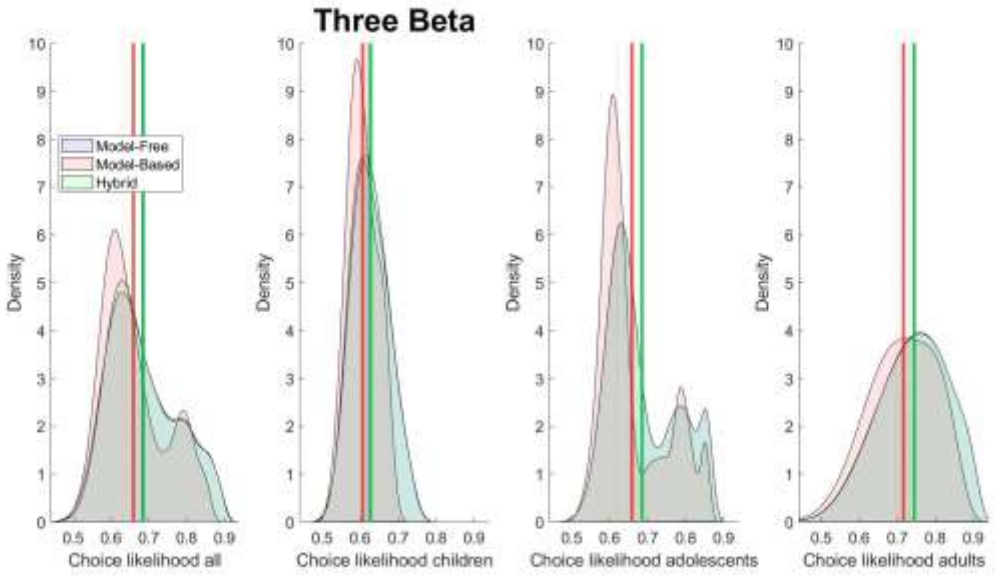


Fig. 11. Gráficas de densidad que comparan la medida de las probabilidades de la etapa 1 y 2 de escoger la misma acción que los sujetos por parte de los modelos MF, MB e híbrido. A) Los modelos con uso de una beta. B) Los modelos con el uso de tres betas.

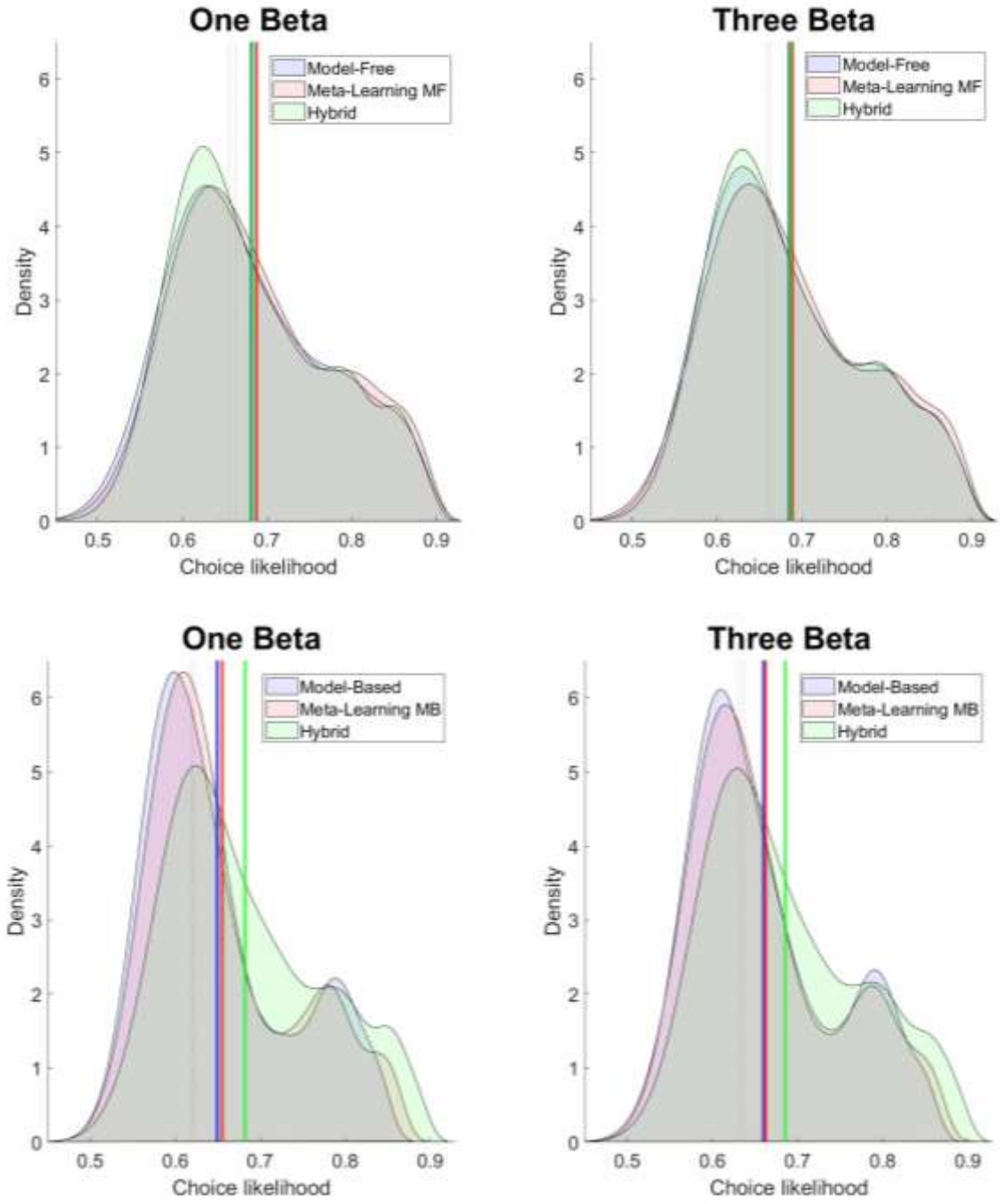


Fig. 12. Gráficas de densidad que comparan la media de las probabilidades de la etapa 1 y 2 de escoger la misma acción que los sujetos por parte de los modelos MF, meta-MF e híbrido en las gráficas superiores y MB, meta-MB e híbrido en las inferiores. A la izquierda los modelos usan una única beta y a la derecha tres.

Retomando el hecho de que el MF explica mejor el comportamiento humano que el MB hemos realizado una serie de diagramas de barras que muestran la probabilidad de escoger la misma opción que los sujetos (P_chosen) en función del modelo y de la categoría de edad (Fig. 15).

Estos resultados arrojan datos interesantes: los sujetos tienen prácticamente la misma proporción de probabilidades entre Model-Free y Model-Based. Siendo siempre las probabilidades de MF ligeramente superiores a las del modelo MB. Además, las probabilidades con el modelo hibrido son cercanas a las del Model-Free pero con una ligera mejora que no llega a apreciarse en la figura. Por lo que podemos llegar a la conclusión de que el comportamiento humano es una mezcla de elecciones tomadas en función del ensayo y error, así como, de elecciones tomadas en función de la experiencia de sucesos anteriores. Aunque, con tendencia a depender ligeramente más del ensayo y error.

También, apreciamos un aumento prácticamente lineal de las probabilidades de ambos modelos a medida que la edad aumenta. Esto se puede deber a una mayor aleatoriedad en la toma de decisiones de los más jóvenes, pudiendo influir en la elección de una acción elementos que no tienen nada que ver con el objetivo, como una preferencia por alguno de los colores de los planetas por parte del niño. Mientras que, a mayor edad, el comportamiento se vuelve más predecible y por lo tanto más fácil de explicar para los modelos de aprendizaje por refuerzo. Asimismo, este comportamiento menos predecible cuanto más joven es el sujeto puede estar relacionado con el *trade-off* entre exploración y explotación, dándose la situación de que los niños y adolescentes son más exploratorios que los adultos. Aunque otras alternativas que pueden explicar el comportamiento según el nivel cognitivo es que los niños sean menos capaces de aplicar el crédito correctamente (asociar el resultado de recompensa o no recompensa a los estímulos elegidos) o que sean menos capaces de lidiar con la volatilidad, las probabilidades cambian con el tiempo, y puede que los adultos seamos más capaces de seguir esta dinámica.

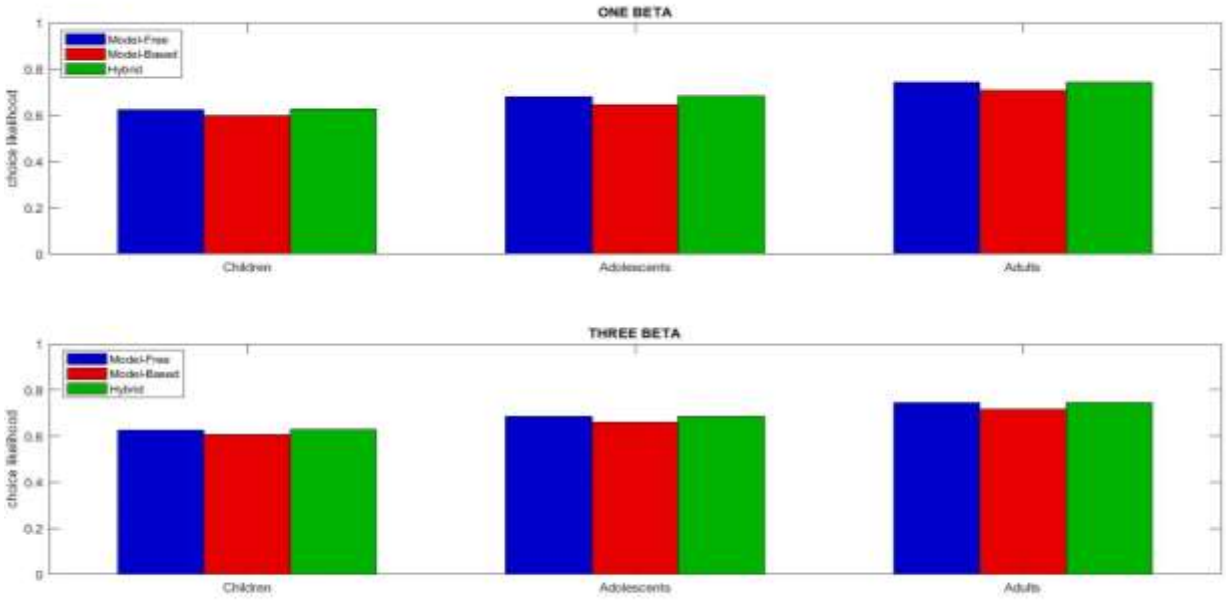


Fig. 13. Los diagramas de barras muestran la probabilidad media de escoger lo mismo que los sujetos en función de usar MF, MB o hibrido. Se dividen por categorías de edad y en las gráficas superiores se usa una beta, mientras que en las inferiores se usan tres.

6. Discusión y conclusiones

Con estos resultados y tras comparar los diferentes modelos acorde a los dos objetivos establecidos previamente, dejando de lado la configuración de 3 betas por los problemas de optimización, podemos establecer que, en cuento a la obtención de una mayor recompensa el modelo con más éxito entre MF y MB es el MB, ya que es el que mayor recompensa media proporciona (Fig. 7). Aunque el modelo híbrido arroja resultados ligeramente mejores respecto al MB (MB: 60.00%, híbrido: 60.47%), la mejora corresponde a un factor más numérico que real, ya que el mejor modelo posible por construcción en este caso es el MB. Además, el híbrido necesita realizar los modelos MB y MF de antemano, puesto que se basa en los resultados de estos. Asimismo, vemos que los modelos son capaces de alcanzar una mayor recompensa que los sujetos, estando los primeros cercanos al 0.6 de media y los segundos entorno al 0.52.

Por otro lado, en la explicación del comportamiento humano, extraemos de los resultados que el modelo con mejor desempeño entre el MF, MB e híbrido ha sido el híbrido, ya que es el que presenta una mayor probabilidad de repetir las elecciones de los sujetos (0.67 aprox.) (Fig. 12), indicando que los humanos hacemos balance entre los sistemas MF y MB.

Además, observamos un problema con el modelo MB y la implementación meta. El modelo MB asume conocimiento a priori del entorno, pero este en realidad se tiene que inferir por los agentes de alguna manera. La implementación meta pretendía llegar a este punto, creando una transición entre modelos MF y MB. Sin embargo, hemos visto que el modelo meta no es adecuado en su implementación actual y requerirá de un diseño alternativo para conseguir inferir la información que ahora se pone externamente en el modelo MB. Una alternativa para futuras líneas de investigación sería estimar las probabilidades de transición entre las etapas en base a las transiciones experimentadas.

Otros resultados de interés son los obtenidos con relación a la categoría de edad, donde vemos que a mayor edad los modelos pueden explicar con mayor exactitud el comportamiento de los sujetos. Esto podría deberse a que los sujetos más jóvenes tienden a ser más exploratorios, por lo que sus elecciones tienen un componente más aleatorio, mientras que los sujetos más adultos son menos exploratorios y más explotadores, teniendo así un comportamiento más predecible. Además, este aumento de la explotación frente a la exploración parece desarrollar una mejor proporción exploración/explotación, haciendo que los adultos consigan mayor media de recompensa que los adolescentes y estos que los niños. Sea como fuere, establecer cual pueda

ser la causa última requiere de experimentación adicional para dilucidar si la exploración, la volatilidad y/o asignación de crédito puedan ser las razones subyacentes.

.

II. PLIEGO DE CONDICIONES

1. Definición y alcance

En esta sección definimos las condiciones mínimas necesarias para poder llevar a cabo el presente proyecto de aprendizaje por refuerzo para el entendimiento del comportamiento humano en la toma de decisiones y las limitaciones de los modelos en la maximización de la recompensa.

2. Condiciones generales

Para la elaboración del trabajo hay que tener en cuenta tanto la salud y seguridad del trabajador con respecto a su puesto de trabajo, así como el posible uso adecuado de los datos, ya que estos pueden incluir información personal de los sujetos. Por lo tanto, hay que atender a dos leyes principalmente, la Directiva 90/270/CEE y la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

Los requisitos mínimos de seguridad y salud relativas al trabajo consisten en disponer de equipos que incluyen pantallas de visualización, definidos en la Directiva 90/270/CEE del Consejo de las Comunidades Europeas, de 29 de mayo de 1990. Según esta, el puesto de trabajo se define como “el conjunto que consta de un equipo con pantalla de visualización provisto, en su caso, de un teclado o de un dispositivo de adquisición de datos y/o de un programa que garantice la interconexión hombre/máquina, de accesorios opcionales, de anejos, incluida la unidad de disquetes, de un teléfono, de un módem, de una impresora, de un soporte de documentos, de una silla y de una mesa o superficie de trabajo, así como el entorno laboral inmediato” (Directiva 90/270/CEE Del Consejo de Las Comunidades Europeas).

Teniendo en cuenta lo citado anteriormente de la Directiva 90/270/CEE, los requisitos mínimos en el puesto de trabajo relativos al equipo, entorno e interconexión hombre/máquina se clasificarán y establecerán de la siguiente manera, que se encuentras reflejados en el anexo de esta misma directiva.

2.1 Equipo

- **Pantalla:** la configuración de los caracteres deberá estar bien definida, ser claros y tener unas dimensiones idóneas. Asimismo, la imagen deberá ser estable y carecer de destellos, por lo que, el usuario, deberá poder adaptarla a sus necesidades.
- **Teclado:** la disposición del teclado permitirá una postura cómoda evitando el cansancio en brazos y manos. Además, las teclas presentarán símbolos claramente legibles.
- **Superficie de trabajo:** ha de ser poco reflectante y de amplia dimensión, para permitir la colocación del resto del equipo, además de documentación y material accesorio.
- **Asiento de trabajo:** debe facilitar libertad de movimiento y una postura cómoda. Teniendo esto en cuenta, la altura deberá de ser regulable y el respaldo ajustable. También, se deberá poner a disposición del usuario un reposapiés.

2.2 Entorno

- **Espacio:** deberá tener una dimensión suficiente de manera que permita cambios de postura y el movimiento durante la realización del trabajo.
- **Iluminación:** tanto la iluminación general como la especial han de garantizar un contraste adecuado pantalla/entorno. Por tanto, el acondicionamiento ha de ser ajustable con el fin de evitar deslumbramientos y reflejos.
- **Reflejos y deslumbramientos:** la instalación del puesto de trabajo debe evitar las fuentes de luz que puedan llegar a provocar deslumbramiento directo y los reflejos en la pantalla.
- **Ruido:** en el diseño de los puestos de trabajo deberá de tenerse en cuenta el ruido producido por los mismos, de manera que el diseño a ser óptimo para evitar la perturbación de la atención.
- **Calor:** los puestos de trabajo no han de producir calor adicional que pueda ocasionar molestias en los usuarios.
- **Emisiones:** toda radiación deberá reducirse a los niveles mínimos con el fin de proteger la salud de los usuarios.
- **Humedad:** ha de mantenerse en niveles óptimos para el trabajo.

2.3 Interconexión ordenador/hombre

Los factores a tener en cuenta para la elaboración, elección, compra y modificación de los programas, son los siguientes:

- Adaptación a la tarea a realizar.
- Los trabajadores deberán tener la formación necesaria para usar el programa y estos deben poder adaptarse al nivel del usuario.
- Los sistemas tienen que facilitar a los usuarios indicaciones sobre el proceso en un formato y ritmo adaptado a los operadores.
- El principio de ergonomía tiene que aplicar sobre todo en el tratamiento de la información por parte del trabajador.

3. Condiciones especificas

En la siguiente sección se enumeran las condiciones específicas a nivel técnico, las cuales se refieren tanto a los elementos físicos del sistema informático (*hardware*) como a los programas (*software*) que se han empleado para la realización de este trabajo.

3.1 Hardware

Las características serán las utilizadas por el escritorio remoto, estas son necesarias para poder hacer uso de la computación en paralelo con hasta 32 tareas ejecutadas al mismo tiempo. Todo esto agiliza la optimización de los modelos.

- CPU: 128 AMD Ryzen Threadripper PRO 3995WX 64-Cores 4.3GHz
- Tarjeta gráfica: NVIDIA Quadro P2200
- RAM: 512 GB
- Disco SSD: 1 TB.
- Sistema Operativo: Linux 5.10.0-1052-oem (64 bits).

3.2 Software

Para la realización del proyecto será necesario hacer uso de *software* especifico. La tabla 1 muestra el software utilizado.

Nombre	Versión	Función
Matlab	R2021b	Entorno de desarrollo integrado (IDE), con un lenguaje de programación propio (lenguaje M), que se usará para la programación de los modelos.
Bayesian Adaptive Direct Search (BADS)	v1.0.6	Algoritmo de optimización bayesiana diseñado para resolver problemas de optimización difíciles, en particular los relacionados con el ajuste de modelos computacionales.
Parallel Computing ToolBox	7.5	Herramienta que permite el uso de la computación en paralelo en Matlab.
Filezilla	1.2.0	Aplicación FTP libre y de código abierto que consta de un cliente y un servidor. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS). Permitirá el intercambio de archivos entre el escritorio remoto y el PC.

Tabla 1. Enumeración de las diferentes aplicaciones y librerías específicas para el desarrollo del proyecto.

III. PRESUPUESTO

1. Introducción

En esta sección mostramos los costes para el desarrollo del proyecto. Los precios de algunos de los materiales son estimativos. Los materiales descritos en este presupuesto son los necesarios para un rendimiento adecuado que reduzca el coste computacional en el tiempo de la optimización de los modelos. De esta manera, los modelos desarrollados tardan unos minutos en ejecutarse tras cada cambio realizado para su mejora. El uso de otro tipo de materiales implicaría un aumento en el tiempo de ejecución, pudiendo ser de hasta varias horas, con el consecuente aumento en número de horas de trabajo. Así, la reducción de costes en cuanto a materiales, escogiendo algunos con especificaciones reducidas en el hardware del usuario, resultaría en un incremento en las horas de trabajo.

2. Costes de hardware

Unidad	Denominación	Cantidad	Precio	Total
Ordenador de torre y su ensamblaje				
Ensamblaje del dispositivo hardware customizado para desarrollar e implementar proyectos basados en *Reinforcement Learning*.				
Materiales				
U	Teclado Approx APPKBECO Teclado USB Negro	1	4,79 €	4,79 €
U	Raton Inphic M2B Wireless	1	10,62 €	10,62 €
U	Placa Base ASUS ROG Strix TRX40-E Gaming	1	609,99 €	609,99 €
U	CPU AMD Ryzen Threadripper PRO 3995WX	1	5.490,00 €	5.490,00 €
U	RAM Kingston Server Premier DDR4 2666MHz PC4-21300 64GB CL19	8	932,08 €	7.456,64 €
U	Samsung 980 Pro SSD 2TB PCIe 4.0 NVMe M.2	1	396,63 €	396,63 €
U	Gigabyte Aorus WaterForce X 360 Kit de Refrigeración Líquida	1	232,79 €	232,79 €
U	GPU PNY Quadro P2200 5GB GDDR5X	1	656,98 €	656,98 €
U	Caracasa Thermaltake V200 Tempered Glass	1	63,69 €	63,69 €
U	Fuente de alimentación Corsair HX1000	1	203,01 €	203,01 €
U	Monitor Asus VP229HE 21.5" LED IPS FullHD FreeSync	1	119 €	119 €
Mano de obra				
h	Técnico informático	2	9,29 €	18,58 €
Costes directos complementarios				
%		2%	15.262,72 €	305,25 €
Total capitulo				
U		1	15.567,97 €	15.567,97 €

Tabla 2. Presupuesto de la partida del ensamblaje del ordenador de torre desglosado en materiales, mano de obra y costes directos complementarios (2%).

3. Costes de software

Unidad	Denominación	Cantidad	Precio	Total
Instalación y configuración del *software*				
Instalación de los programas y librerías específicas en los ordenadores.				
Materiales				
U	Linux 64 bits	1	0,00 €	0,00 €
U	Matlab Educativa Perpetual Licence	1	500,00 €	500,00 €
U	Bayesian Adaptive Direct Search (BADS)	1	0,00 €	0,00 €
U	Parallel Computing ToolBox	1	0,00 €	0,00 €
U	Filezilla	1	0,00 €	0,00 €
Mano de obra				
h	Técnico informático	2.5	9,29 €	23,23 €
Costes directos complementarios				
%		2%	523,23 €	10,46 €
Total capitulo				
U		1	533,69 €	533,69 €

Tabla 3. Presupuesto de la partida de la instalación del software específico en el ordenador de torre desglosado en materiales, mano de obra y costes directos complementarios (2%).

4. Coste del desarrollo del proyecto

Unidad	Denominación	Cantidad	Precio	Total
Desarrollo del proyecto				
Presupuesto del desarrollo de los diversos modelos de aprendizaje por refuerzo.				
Mano de obra				
h	Ingeniero de datos	480	15,00 €	7.200,00 €
Costes directos complementarios				
%		4%	7.200,00 €	288,00 €
Total capítulo				
U		1	7.288,00 €	7.288,00 €

Tabla 4. Presupuesto de la partida de desarrollo del proyecto desglosado en mano de obra y costes directos complementarios (4%). Se asume que el estudio tendrá una duración aproximada de 6 semanas.

5. Resumen del presupuesto

Capítulo	Importe
Capítulo 1. Ensamblaje de ordenador de torre	15.567,97 €
Capítulo 2. Instalación y configuración del software	533,69
Capítulo 3. Desarrollo del proyecto	7.288,00 €
TOTAL PRESUPUESTO DE EJECUCIÓN MATERIAL	23.389,66 €
medios auxiliares (4%)	935,59 €
TOTAL PRESUPUESTO DE EJECUCIÓN POR CONTRATA	24.325,25 €
IVA (21%)	5.108,30 €
TOTAL PRESUPUESTO BASE DE LICITACIÓN	29.433,55 €

Tabla 5. Presupuesto de la partida del desarrollo del proyecto desglosado en mano de obra y costes directos complementarios (4%). Se asume que el proyecto tendrá una duración aproximada de 12 semanas.

IV. ANEXO CÓDIGO

Enlace del código en GitHub:

https://github.com/LofNaDI/TFG_MF_MB_metaQL.git

V. BIBLIOGRAFÍA

1. Klossek UMH, Russell J, Dickinson A. The Control of Instrumental Action Following Outcome Devaluation in Young Children Aged Between 1 and 4 Years. 2008;
2. Mischel W, Shoda Y, Rodriguez ML. Delay of Gratification in Children Downloaded from [Internet]. 1989. Available from: http://science.sciencemag.org/
3. Daw ND, Niv Y, Dayan P. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nature Neuroscience 2005 8:12 [Internet]. 2005 Nov 6 [cited 2021 Aug 11];8(12):1704–11. Available from: https://www.nature.com/articles/nn1560
4. Sutton RS, Barto AG. Reinforcement Learning: An Introduction Second edition, in progress. 2018.
5. Torres J. Introducción al aprendizaje por refuerzo profundo. 2015.
6. Decker JH, Otto AR, Daw ND, Hartley CA. From Creatures of Habit to Goal-Directed Learners: Tracking the Developmental Emergence of Model-Based Reinforcement Learning. Psychological Science. 2016 Jun 1;27(6):848–58.
7. Otto AR, Raio CM, Chiang A, Phelps EA, Daw ND. Working-memory capacity protects model-based learning from stress. Proc Natl Acad Sci U S A [Internet]. 2013 Dec 24 [cited 2022 Feb 21];110(52):20941–6. Available from: https://www.pnas.org/content/110/52/20941
8. Daw ND, Gershman SJ, Seymour B, Dayan P, Dolan RJ. Model-based influences on humans' choices and striatal prediction errors. Neuron. 2011 Mar 24;69(6):1204–15.
9. Directiva 90/270/CEE del Consejo de las Comunidades Europeas.