UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Recreant la presa de decisions humana mitjançant aprenentatge per reforç

Autor: Martí Pirla Torrell

Director: Dr. Ignasi Cos Aguilera
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 11 de juliol de 2024

Abstract

In this project, we study the data collected by Michael DePass on how a group of subjects performed a task aimed at obtaining the maximum reward. The subjects repeated the task a total of 300 times, in which they had to choose between two stimuli presented on a screen. Over these 300 attempts, the subjects needed to figure out which stimuli to select to achieve the best possible reward. With this data, we will interpret their behavior and apply reinforcement learning to the same problem to compare the differences in optimal decision-making strategies. Finally, the goal is to fit the hyperparameters of a Q-learning agent to make its behavior closely resemble that of the human subjects.

Resum

En aquest projecte estudiem les dades que va recollir el Michael DePass de com un conjunt de subjectes feien un exercici intentant trobar com aconseguir la màxima recompensa. Els subjectes repetien l'exercici un total de 300 vegades, en el qual havien d'escollir entre dos estímuls en una pantalla. A través d'aquests 300 intents, els subjectes havien de descobrir quins estímuls escollir per aconseguir la millor recompensa possible. Amb aquestes dades, interpretarem el seu comportament i aplicarem aprenentatge per reforç al mateix problema per comparar les diferències en la presa de decisions entre l'algorisme de Q-learning i els subjectes. Finalment, l'objectiu és ajustar els hiperparàmetres d'un agent de Q-learning per aconseguir que el seu comportament s'asimili al màxim al dels subjectes humans.

Resumen

En este proyecto estudiamos los datos recogidos por Michael DePass sobre cómo un conjunto de sujetos realizaron un ejercicio con el objetivo de obtener la máxima recompensa. Los sujetos repitieron el ejercicio un total de 300 veces, en las cuales debían elegir entre dos estímulos presentados en una pantalla. A lo largo de estos 300 intentos, los sujetos debían descubrir qué estímulos seleccionar para conseguir la mejor recompensa posible. Con estos datos, interpretaremos su comportamiento

y aplicaremos aprendizaje por refuerzo al mismo problema para comparar las diferencias en la toma de decisiones óptimas. Finalmente, el objetivo es ajustar los hiperparámetros de un agente de Q-learning para lograr que su comportamiento se asemeje al máximo al de los sujetos humanos.

Agraïments

Vull agrair tant a l'Ignasi Cos com al Michael DePass, no només l'ajuda que m'han donat durant tot aquest temps sempre que la demanava, sinó també per donar-me l'oportunitat i les dades per tal de fer el treball de final de grau d'un tema tant interessant. També vull agrair a la meva família per tot el suport i l'ajuda, tant directa com indirecta, que m'han donat durant aquest temps.

Índex

Capítol 1

Introducció

1.1 El projecte

L'aprenentatge per reforç, ha estat profundament inspirat per la manera en què els animals aprenen i s'adapten a nous entorns. Aquest tipus d'aprenentatge és essencialment una forma de prendre decisions basada en l'experiència acumulada, on els agents autònoms aprenen a maximitzar una recompensa acumulada a través de la interacció contínua amb el seu entorn.

En aquest treball, investigarem el comportament d'un conjunt de 18 participants humans davant d'un problema específic davant el qual han de maximitzar la recompensa obtinguda en cada episodi. Paral·lelament, estudiarem el comportament d'un agent basat en l'algorisme de Q-learning enfront del mateix problema. L'objectiu és analitzar les similituds i diferències entre el comportament humà i el comportament de l'agent artificial, i finalment, crear un model de Q-learning que repliqui la manera en què els humans aprenen en aquest context.

A través d'aquesta investigació, pretenem comprendre millor com els humans i els agents de Q-learning s'enfronten a situacions d'aprenentatge i adaptació, i com aquest coneixement pot millorar els algorismes d'aprenentatge automàtic i ajudarnos a entendre millor els processos cognitius humans.

1.2 Estructura de la Memòria

Aquest treball està dividit en un total de 5 apartats, en el que es comença introduint les motivacions i objectius del treball, seguit d'una explicació de tots els conceptes teòrics que s'han fet servir. Després s'explicaran tots els processos previs a l'obtenció dels resultats que s'han fet, amb les seves justificacions. Per acabar, es donen els resultats obtinguts en cadascun dels mètodes explicats anteriorment i una conclusió, on s'explicarà el que s'ha trobat en el treball i possibles parts que es podrien ampliar en un futur. Al final hi haurà una bibliografia de les fonts d'informació que s'han fet servir per fer la recerca.

Tot el codi que s'ha utilitzat en aquest treball es troba en aquest repositori de github: https://github.com/MaPirla/TFG_Code_MartiPirla.git i les dades ja prepro-

cessades que s'han aconseguit en aquest enllaç de Kaggle https://www.kaggle.com/datasets/martpirla
del-tfg-del-mart-pirla. Per a més informació sobre el codi es recomana mirar el fitxer
README.md del repositori de github.

Capítol 2

Marc Teòric

2.1 Algorisme K-means

L'algorisme K-Means és un mètode de clústering que té com a objectiu agrupar les mostres donades per l'usuari en k grups, sent k un paràmetre que decideix l'usuari al començament de l'execució. El que l'algorisme busca és minimitzar la variància entre els elements de cada clúster, referit com a cohesió i, a la vegada, maximitzar la variància entre elements de clústers diferents, referit com a separació. Aquest algorisme és molt útil a l'hora de trobar, en un conjunt de dades, elements que comparteixin característiques comunes. Aquesta tècnica és molt utilitzada en mineria de dades i aprenentatge automàtic quan es tenen moltes dades.

A l'hora de l'execució, l'algorisme comença escollint k elements del conjunt de dades donades com a centroides inicials dels clústers. També es poden inicialitzar els centroides assignant-lis un punt aleatori dins de l'espai de les dades.

Ja amb els centroides, classifica cada element del conjunt de dades en un dels clústers calculant la distància euclidiana de l'element a cada centroide i assignant-lo al clúster amb el centroide més proper.

Ja amb tots els elements classificats, es recalculen els centroides dels clústers fent la mitjana de tots els punts de les dades dins d'aquell clúster.

Quan s'han recalculat els clústers, es torna a fer la classificació dels elements de les dades, ja que si els centroides han canviat de posició, l'assignació de grups pot ser diferent. El cicle de reclassificació dels punts i recalculació dels centroides es fa fins que es vegi que no es modifiquen les assignacions dels elelments a cada clúster o que els centroides no canvien gaire o gens de posició.

A l'hora d'escollir el valor de k, és crític trobar un valor que s'adapti perfecte al problema, ja que, si el valor és massa petit, el resultat tindrà grups massa grans, amb cohesió baixa, i si, en canvi, el valor és massa alt, l'algorisme crearà grups massa similars entre si, amb una separació massa baixa.

Per això, quan es fa servir el K-means i no es necessita que les dades estiguin agrupades en un número específic de clústers, sinó que es vol que es faci l'agrupació òptima sense importar el número de grups, és necessari fer servir altres mètodes com l'*elbow method* per tal de determinar el valor de k que ens donarà el millor

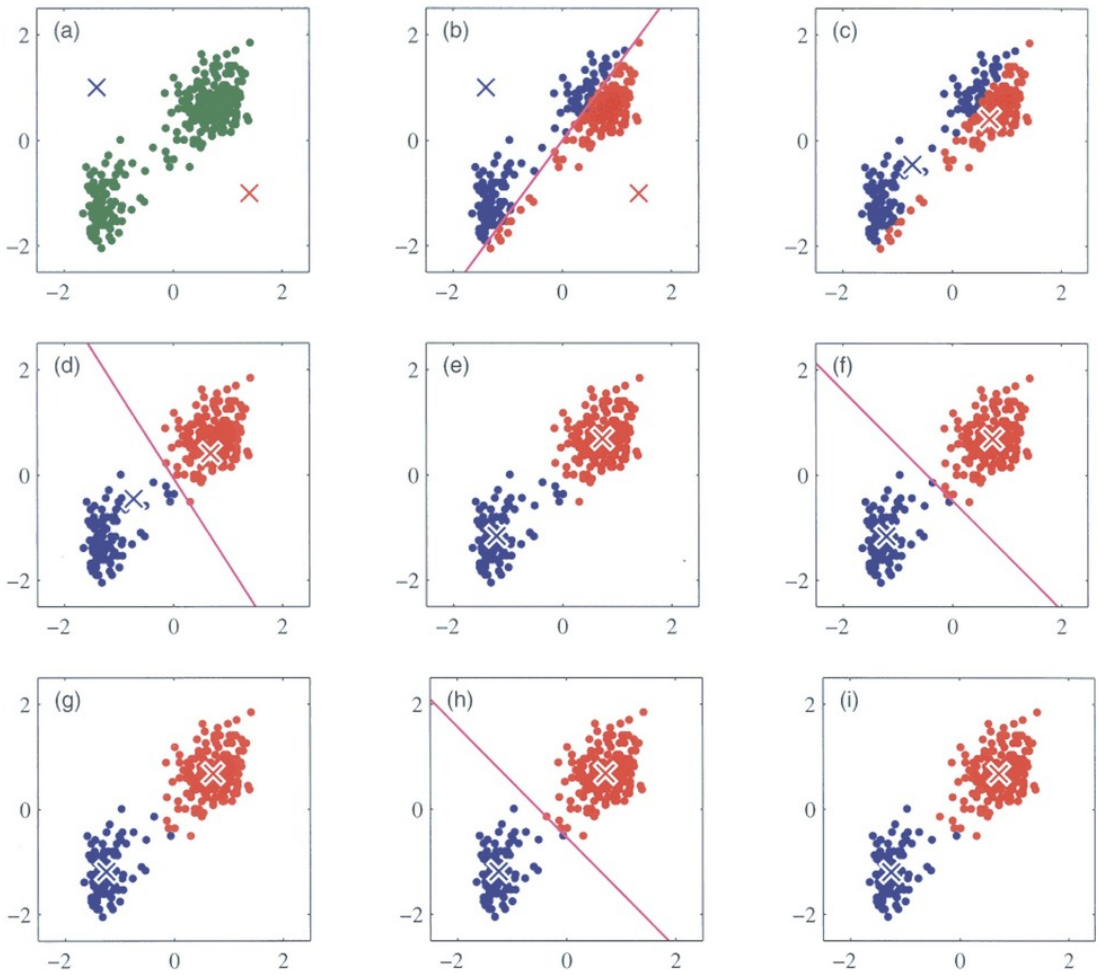

Figura 1: Visualització de l'algorisme K-means. Les X són els centroides de cada clúster, els punts són les mostres de dades i el seu color indica al clúster que s'ha assignat per última vegada. La recta indica la separació entre un clúster i un altre. En la figura a, es veuen els punts de les dades en verd, ja que encara no s'han classificat, i veiem que els dos centroides s'han inicialitzat en dos punts aleatoris de l'espai. Seguidament, en la figura b veiem com es fa la classificació dels elements en els dos clústers en base a quin centroide tenen més a prop, per després, en la figura c, recalcular-se els centroides perquè quedin en el centre del grup. Podem veure que en la figura d es torna a fer la classificació en base als nous centroides, de manera que es repeteix el procés fins que s'arriba al vuitè pas, representat en la figura i, en el que s'acaba l'execució degut a que els centroides quasi no s'han mogut a l'hora de recalcular-los respecte al quart pas, representat en la figura 4.

equilibri entre cohesió i separació abans d'executar l'algorisme K-means de forma definitiva.

L'*elbow method* és la tècnica que més es fa servir per aconseguir un valor de k òptim per un grup de dades, en el que es tingui un equilibri entre cohesió i separació dels clústers. El que es fa és executar l'algorisme K-means amb les mateixes dades per cada valor enter de k dins d'un rang determinat, el qual determina l'usuari. Per cada execució de K-means amb un valor de k diferent, es calcula la suma dels errors quadrats (WCSS) dins de cada clúster, es a dir, es sumen les distàncies euclidianes elevades al quadrat de cada element de les dades amb el centroide del seu clúster:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

On C_i és el conjunt de punts del clúster i i μ_i és el centroide del clúster i.

Ja amb les sumes dels errors quadrats per cada valor de k calculats, el que es fa després és generar una representació gràfica del WCSS per cada valor de k, on l'eix X és la k i l'eix Y és el WCSS (figura 2). Amb aquesta representació, si el valor de k òptim està dins del rang, es genera una mena de colze al voltant d'aquell valor, on el WCSS deixa de decréixer amb tanta velocitat i s'estabilitza.

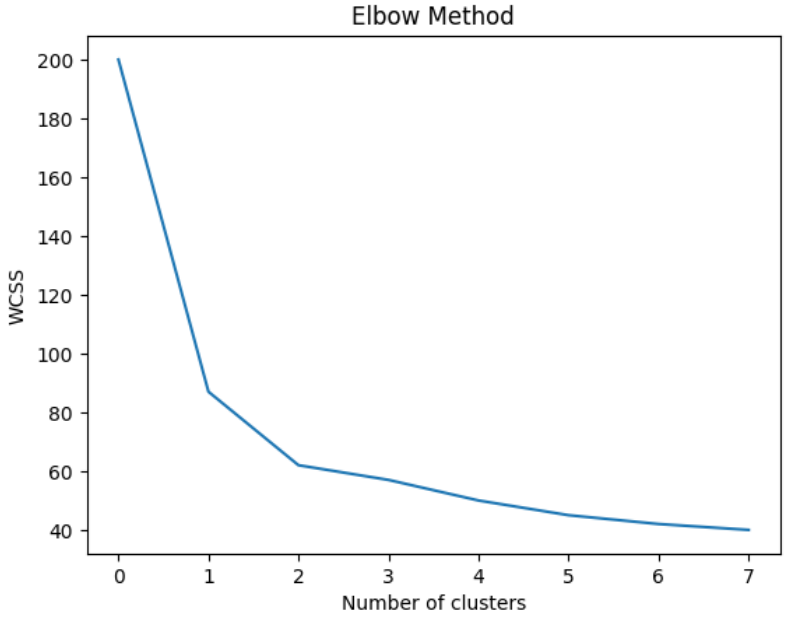

Figura 2: Recreació de com es veuria el gràfic creat sobre la suma dels errors quadrats, on es pot veure com al voltant del valor de $k = 2$ el WCSS s'estabilitza i fa aquesta forma de colze.

Aquesta forma de colze el que significa és que, a partir d'aquell valor de k,

augmentar el número de clústers no millorarà la cohesió de forma significativa, fent que aquesta petita millora no compensi l'empitjorament de la separació entre grups.

2.2 Aprenentatge per reforç

L'aprenentatge per reforç és un camp de l'aprenentatge automàtic en el que els agents tenen com a objectiu triar les accions òptimes per tal d'aconseguir la millor recompensa possible en un entorn. L'avantatge que té l'aprenentatge per reforç davant d'altres camps de l'aprenentatge automàtic com l'aprenentatge supervisat, és que pot aprendre interactuant directament amb l'entorn en comptes de necessitar de mostres ja etiquetades per el procés d'aprenentatge.

En l'aprenentatge per reforç, l'agent és presentat davant d'un entorn modelat com un procés de decisió de Markov, el qual està format per un conjunt d'estats S i un conjunt d'accions possibles en cada estat $A(s)|s \in S$. Quan l'agent es troba en un estat s i pren una acció $a \in A(s)$, l'entorn respon amb un canvi d'estat s' i una recompensa en forma d'un valor numèric. D'aquesta manera, l'aprenentatge d'un agent d'aprenentatge per reforç consisteix en un cicle en el que l'agent fa una acció sobre l'entorn, i aquest respon modificant el seu estat i donant una recompensa a l'agent, fent que l'agent torni a prendre una acció en aquest nou estat i es repeteixi aquesta reacció d'un als canvis de l'altre fins que l'usuari consideri que l'agent ja ha après suficient (figura 3).

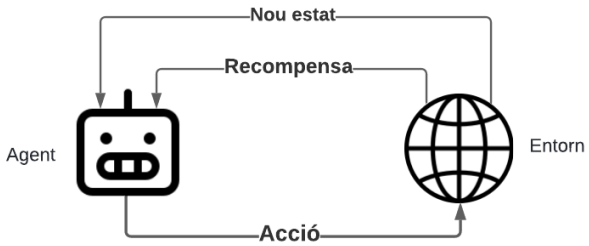

Figura 3: Representació d'un escenari d'aprenentatge per reforç. L'agent escull una acció, i l'entorn respon canviant d'estat i donant una recompensa a l'agent.

En aquest procés d'interacció amb l'entorn, l'objectiu de l'agent és trobar una política òptima en l'entorn, la qual maximitzi la recompensa acumulada donada per aquest. Per aconseguir aquest objectiu, l'agent comença amb una política subòptima, que després va modificant a mesura que obté més informació de l'entorn.

En la majoria dels casos, si un agent sempre seguís les accions dictades per la seva política, que en aquell moment segurament és subòptima, l'hi seria impossible trobar

cap política alternativa millor, ja que sempre estaria prenent les mateixes decisions i no provaria opcions que l'hi podrien donar una millor recompensa. Per això, tot agent d'aprenentatge per reforç busca mantenir un equilibri entre explotació i exploració. Quan un agent explora, té com a objectiu escollir accions que no hagi escollit tantes vegades per tal de buscar si escollint-les rep una millor recompensa. En canvi, quan explota, l'agent segueix la seva política a l'hora d'escollir les accions amb l'objectiu d'assegurar aconseguir una bona recompensa dins del que ja s'ha explorat.

L'estratègia que més es fa servir, ja que és de les més senzilles, és l'anomenada epsilon greedy, la qual consisteix en establir un paràmetre epsilon $0 < \epsilon < 1$ que, cada vegada que es vol escollir una acció, es compara amb un valor generat aleatoriament entre 0 i 1, de manera que si el valor aleatori és més alt que ϵ, l'agent explota, i, en canvi, si és inferior, s'explora escollint una acció aleatòria. D'aquesta manera es pot ajustar aquest equilibri entre exploració i explotació modificant el paràmetre ϵ.

En aquest estudi, ens centrarem en l'algorisme de Q-learning d'entre de tots els algorismes que hi ha ja que és el que farem servir en el projecte.

2.2.1 Q-learning

El Q-learning és un algorisme d'aprenentatge per reforç que busca la política òptima buscant els valors $q_*(s, a)$ per totes les accions de cada estat. Aquest valor representa la recompensa que donarà escollir l'acció a en l'estat s tant de forma immediata, com en el futur. Llavors, aconseguir $q_*(s, a)$ per totes les combinacions d'estats i accions dona el que seria la política òptima del problema.

Per trobar els valors de cada parell estat-acció, l'algorisme inicialitza els anomenats Q-valors $Q(s, a)$, els quals serviran com aproximacions dels corresponents valors $q_*(s, a)$. Els Q-valors s'acostumen a inicialitzar en un valor aleatori o a zero.

En aquest punt, l'agent té una política que probablement no sigui òptima, per la qual cosa comença a aprendre tal com hem explicat en l'apartat d'aprenentatge per reforç, interactuant amb l'entorn.

Cada vegada que l'agent esculli una acció a en un estat s, l'entorn canviarà d'estat s' i donarà una recompensa r. Amb aquesta informació, l'agent actualitza el Q-valor $Q(s, a)$ seguint la fórmula següent:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s', a') - Q(s, a)]$$

On:

- s és l'estat actual.
- a és l'acció presa en l'estat s.
- r és la recompensa rebuda després de prendre l'acció a.
- α és la taxa d'aprenentatge (learning rate), un paràmetre entre 0 i 1 que determina quant influeix cada nova informació en l'estimació actual del valor Q. Un factor de 0 faria que l'agent no modifiqués els seus Q-valors i que llavors no aprengués res, mentre que un factor igual a 1 faria que el Q-valor fos igual a la última mesura. En un entorn determinista, una taxa d'aprenentatge de 1 és el valor òptim, com més aleatori sigui el comportament de l'entorn, més baix haurà de ser el valor.
- s és el nou estat després de prendre l'acció a.
- γ és el factor de descompte (discount factor), un paràmetre superior o igual a 0 que determina la importància de les recompenses futures en comparació amb les recompenses immediates. Com més s'apropa el paràmetre a 1, l'agent donarà importància a les recompenses a llarg termini.
- $max_{a'}Q'(s', a')$ és el valor Q màxim per al nou estat s i totes les possibles accions a, que representa la millor recompensa esperada futura a partir del nou estat.

Aquesta fórmula es pot interpretar com una mitjana ponderada entre l'últim valor i la nova recompensa. Veiem com es calcula la mitjana dels primers n elements en un conjunt de valors on l'element k del conjunt és a_k:

$$m_n = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Si llavors treiem l'últim element del sumatori:

$$m_n = \frac{1}{n}(a_n + \sum_{i=1}^{n-1} a_i)$$

I també podem deduir que:

$$\sum_{i=1}^{n-1} a_i = m_{n-1}(n-1)$$

Podem substituir-ho de manera que queda així:

$$m_n = \frac{1}{n}(a_n + m_{n-1}(n-1))$$

I si desenvolupem l'equació ens queda d'aquesta manera:

$$m_n = m_{n-1} + \frac{1}{n}(a_n - m_{n-1})$$

Aquesta fórmula és molt útil per calcular la mitjana de forma incremental, de forma que pots afegir un valor al càlcul sense haver de saber tots els valors anteriors.

Si observem l'estructura de la fórmula pel càlcul incremental de la mitjana, podem veure que si $\frac{1}{n}$ es canvia per un valor fix α, es pot veure que la fórmula d'actualització del Q-learning fa una mena de mitjana entre els valors anteriors i la recompensa actual, que, degut a que α és un valor fix en comptes de $\frac{1}{n}$, dona un pes major als últims valors afegits en comptes de donar-l'hi el mateix a tots. Aquest canvi no és menys costós en termes de memòria computacional, sinó que aquest desequilibri entre els valors més antics i els últims és molt útil per agents en entorns dinàmics, en els que la política òptima pot canviar amb el temps.

2.3 Ajustament d'hiperparàmetres

L'ajustament d'hiperparàmetres o *hyperparameter fitting* és el procés de buscar els paràmetres òptims per un model d'aprenentatge automàtic. Els hiperparàmetres són el conjunt de paràmetres als que no se'ls hi assigna un valor en el procés d'aprenentatge, sinó que s'han d'establir abans de començar l'entrenament. Per exemple, en el Q-learning, la taxa d'aprenentatge o el factor de descompte són exemples d'hiperpraràmetres.

L'ajustament d'hiperparàmetres es comença determinant quins seran els hiperparàmetres dels quals es vol buscar el valor òptim i s'estableix un rang de valors que l'usuari consideri possibles valors òptims. Amb aquest espai d'hiperparàmetres establert, es decideix quin algorisme es farà servir com a estratègia de cerca, aquest algorisme determinarà quin conjunt de paràmetres trobar i s'encarregarà de determinar el conjunt que dona un millor resultat. La mesura del resultat és una funció determinada per l'usuari.

L'exemple més senzill d'estratègia de cerca és l'anomenada cerca en graella, que consisteix en provar totes les combinacions possibles de valors d'hiperparàmetres i buscar quina combinació dona un millor resultat. Aquest algorisme encara que és el més senzill, és molt costós quan es tenen rangs amb molts valors, ja que ha de provar totes les combinacions possibles.

Hi ha altres estratègies de cerca com la cerca aleatòria, que prova combinacions aleatòries dins dels rangs especificats per tal de no haver de provar totes les possibles combinacions, o els algorismes evolucionaris, que fan servir els principis de la selecció

natural per a l'optimització, En aquest treball però, ens centrarem en els algorismes de optimització bayesiana, ja que és el tipus d'algorisme que utilitzarem.

L'optimització bayesiana és un conjunt d'estratègies que fan servir models probabilístics per predir el rendiment d'altres combinacions d'hiperparàmetres que no s'han provat i a l'hora d'escollir la combinació d'hiperparàmetres, selecciona les que tenen un resultat més prometedor en el seu model.

Un dels models dins de la optimització bayesiana és el *Tree-structured Parzen Estimator* o TPE, el qual construeix un model que té com a objectiu predir el comportament de la funció objectiu a base de mostrejar l'espai d'hiperparàmetres. La funció objectiu és el comportament del resultat en funció dels canvis en els hiperparàmetres.

Aquest algorisme fa servir dues distribucions de densitat per modelar la funció objectiu, la $l(x)$, la qual intentarà modelar la densitat de combinacions de paràmetres que donaran un bon resultat, i la $g(x)$, la qual modelarà la densitat de combinacions d'hiperparàmetres amb mals resultats.

L'algorisme de TPE comença provant uns quants conjunts d'hiperparàmetres aleatoris per tenir les mostres de la funció objectiu. Després, es divideixen els conjunts d'hiperparàmetres en dos grups, aquells que han donat millors resultats i aquells que no, per després adaptar les distribucions $l(x)$ i $g(x)$ amb cadascun dels grups respectivament. Ja amb aquesta nova informació, l'algorisme escull uns altres conjunts d'hiperparàmetres basant-se en quina combinació el seu model considera que pot donar millors resultats seguint la proporció $\frac{l(x)}{g(x)}$ en tots els punts de l'espai. Després d'avaluar aquest nou grup, es tornen a adaptar les distribucions de densitat per tal d'incorporar els nous resultats, i això es repeteix tantes vegades com l'usuari consideri suficients. Quan l'algorisme ha fet totes les iteracions, llavors retorna el conjunt d'hiperparàmetres que ha provat i li ha donat un millor resultat.

L'algorisme TPE és molt millor que altres mètodes com la cerca en graella degut a la seva alta eficiència, i a la seva capacitat de gestionar espais complexos tant amb paràmetres continus com discrets, fent-lo una bona opció per la cerca d'hiperparàmetres òptims en espais molt grans.

2.4 Transfer Learning

El *transfer learning* és una tècnica que cada vegada es fa servir més en el camp de l'aprenentatge automàtic, on s'utilitzen les dades d'un model ja entrenat en una tasca per fer més ràpid l'aprenentatge en un altre tasca normalment semblant a la primera, encara que no igual. Aquesta tècnica s'acostuma a fer servir en grans

models complexos com pot ser una xarxa neuronal convolucional, de manera que si es vol crear una xarxa neuronal convolucional que faci una tasca de reconeixement d'imatges molt concreta, no cal que l'entrenis de zero, sinó que pots fer servir una ja entrenada que hagi après de forma genèrica per després només haver-la d'entrenar en la tasca concreta.

Moltes vegades, per fer que les dades del model pre-entrenat es puguin fer servir en la nova tasca, cal fer alguns ajustaments. En models grans el que s'acostuma a fer és bloquejar parts del model que se saben que no cal modificar per la nova tasca, de manera que en el nou procés d'entrenament es tingui menys paràmetres per modificar. A vegades també es fan modificacions en l'estructura del model de manera que es minimitzi el temps d'adaptació entre la tasca en la que el model va ser pre-entrenat i la nova tasca.

Capítol 3

Disseny experimental

3.1 L'experiment

L'exercici que es va demanar fer als subjectes estava dividit en 300 episodis de diferents duracions, en els que havien d'escollir, en cada intent de l'episodi, un dels dos estímuls que se'ls hi presentaven. L'objectiu que tenia el subjecte era el de maximitzar la recompensa acumulativa aconseguida en cada episodi, sent aquesta la suma de les mides dels estímuls seleccionats en cada intent del l'episodi. D'aquesta manera, l'agent haurà de descobrir a través de l'experiència, que escollir sempre l'opció que dona una recompensa immediata més alta no té perquè comportar la millor recompensa acumulativa al final de l'episodi. Això s'aconsegueix fent que si s'escull l'estímul més gran, la mida mitjana dels estímuls de els dels següents intents dins del mateix episodi baixi, i que si s'escull el més petit, la mitjana pugi.

Per dur a terme l'exercici amb els subjectes, es va utilitzar un programa informàtic que els hi presentava en cada intent una pantalla negra on, primer, apareixia una columna blava en un costat de la pantalla, sent l'alçada d'aquesta la mida d'un dels estímuls, per després, mostrar una altra columna del mateix color en el costat contrari, sent l'alçada la mida de l'altre estímul.

Fent que els estímuls no només es diferenciessin en la seva mida, sinó també en la seva posició i ordre d'aparició, es pretenia veure si els subjectes buscaven l'estratègia òptima seguint aquests criteris o si es centraven només en la seva mida.

En aquest exercici, es van fer un total de 100 episodis on l'horitzó era igual a 0, uns altres 100 amb horitzó igual a 1, i finalment 100 amb horitzó igual a 2. Cada grup de 100 episodis amb el mateix horitzó estava subdividit en 5 grups on la diferència de mida entre estímuls era de 0.01, 0.05, 0.1, 0.15 i 0.2 respectivament.

3.1.1 Generació d'estímuls

Per tal de generar els estímuls, es tenen en compte 3 paràmetres diferents:

- **Profunditat de l'horitzó (n_H):** Es pot entendre com el número d'intents posteriors de l'intent inicial, en aquest cas n_H prendrà els valors de 0, 1 i 2.
- **Diferència entre estímuls(ΔS):** Això representa la diferència entre el valor

dels dos estímuls en cada intent, prendrà els valors de 0.01, 0.05, 0.1, 0.15 i 0.2.

- **Guany/Pèrdua (G):** Aquest paràmetre determina el augment o disminució que es fa a la mitjana dels estímuls dels següents intents depenent de si s'escull l'estímul menor o major respectivament. Per $n_H = 0$ i $n_H = 1$, G prendrà el valor de 0.3, i per $n_H = 2$, el valor de G serà 0.19.

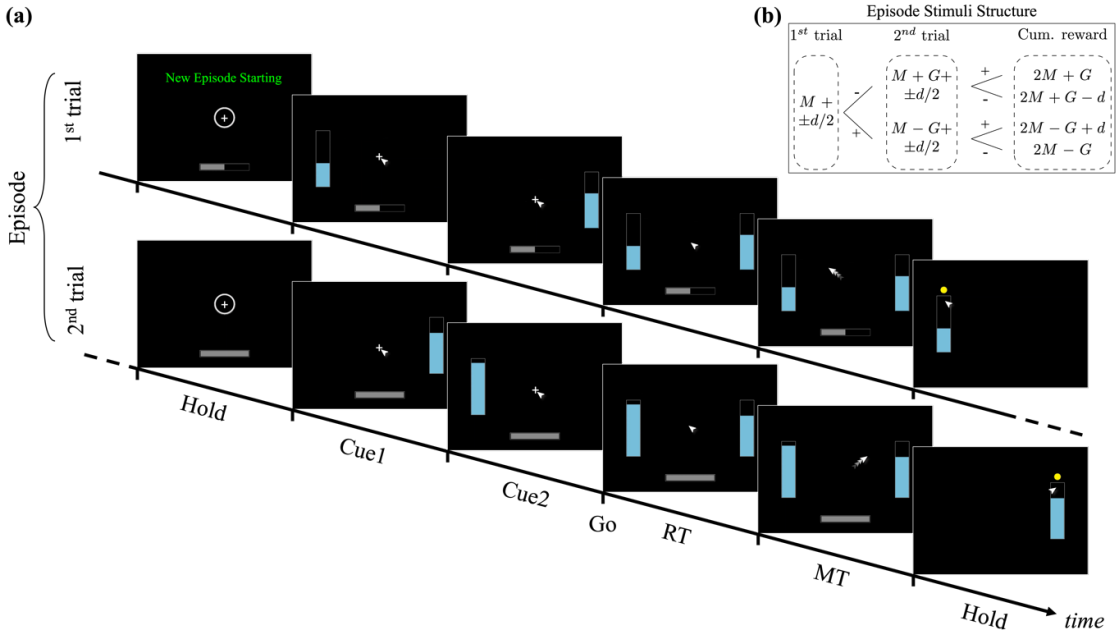

Figura 4: Diagrama de l'estructura d'un episodi de l'experiment. En aquest cas, un episodi amb horitzó 1. Podem veure tant el primer intent (trial) a la part superior, com el segon a la part inferior. Es pot veure com primer es mostra un estímul, i després l'altre, i com estan posicionats cadascun. En la part b de la figura, podem veure el càlcul de les recompenses ens donaria cada combinació d'accions sent el símbol − el que indica escollir l'estímul gran, i el símbol + el petit.

Per $n_H = 0$, el parell d'estímuls $s_{1,2}$ són generats de manera que tinguin un valor mitjà m i tinguin una diferència entre ells de ΔS, es a dir, $s_{1,2} = m \pm \Delta S/2$. De manera que els estímuls estiguin sempre entre 0 i 1, la mitjana m es un número aleatori entre $[\Delta S/2, 1 - \Delta S/2]$.

Per $n_H = 1$, cada episodi consisteix en dos intents, en el que les mides dels estímuls del segon intent dependran de la decisió presa en el primer. El primer parell d'estímuls es generen de la mateixa manera que es fa en el cas de $n_H = 0$, es a dir, $s_{1,2} = m \pm \Delta S/2$, però, com hem dit abans, la generació del segon parell dependrà de que s'esculli en el primer intent de l'episodi, de manera que

si s'escull l'estímul més petit, els següents es generaran amb la següent fórmula, $s_{1,2} = m + G \pm \Delta S/2$, mentre que si s'escull el més gran, els estímuls es generaran seguint aquesta, $s_{1,2} = m - G \pm \Delta S/2$. Per fer que tots els estímuls de l'episodi es mantinguin entre 0 i 1, la mitjana m en aquest cas s'haurà de generar dins l'interval $[G + \Delta S/2, 1 - G - \Delta S/2]$.

Finalment per $n_H = 2$, cada episodi consisteix d'un total de 3 intents, els quals són dependents de la mateixa manera que eren el primer i el segon en el cas de $n_H = 1$, sent els valors dels estímuls de cada intent dependents de l'elecció que s'hagi pres en els intents anteriors. D'aquesta manera, els estímuls del primer intent seguiran $s_{1,2} = m \pm \Delta S/2$, els del segon intent seguiran $s_{1,2} = m - G \pm \Delta S/2$ o $s_{1,2} = m + G \pm \Delta S/2$ i, finalment, els estímuls del tercer intent seguiran $s_{1,2} = m \pm G \pm G \pm \Delta S/2$ depenent de quins estímuls s'escullin en els dos primers intents. En aquest cas, per evitar una altra vegada que els estímuls no estiguin entre 0 i 1, no només s'estableix el limit de m entre $[2G + \Delta S/2, 1 - 2G - \Delta S/2]$, sinó que també canviem el valor de G de 0.3 a 0.19.

3.2 Estructura de les dades

Les dades que farem servir en aquest estudi han estat donades per Michael DePass, que les va recollir d'un total de 18 subjectes fent l'exercici que s'ha descrit anteriorment. Les dades recollides tenien molta més informació de la que farem servir en aquest treball, de manera que el primer que vam fer va ser crear un altre fitxer de dades que només contingués les dades que calen pel treball. També vam modificar algues columnes per tal de fer més còmode treballar-hi i es van fer alguns controls per evitar inconsistències en les dades.

Després de les modificacions a les dades originals i l'eliminació de dades no rellevants, la taula resultant queda de la següent manera:

- **subject:** Aquesta columna, amb un valor enter, determina quin era l'agent que va fer aquest intent.
- **horizon:** Indica la profunditat de l'horitzó de l'episodi en el que es feia l'intent.
- **nte:** Representa la profunditat dins de l'episodi de l'intent, per exemple, si $nte = 1$, vol dir que l'intent és el primer de l'episodi, i si $nte = 3$ i $horizon = 2$, vol dir que l'intent és l'últim de l'episodi.
- **big:** Un booleà que ens diu si l'estímul que va escollir el subjecte era el més gran dels dos.

- **right:** Un booleà que ens diu si l'estímul que va escollir el subjecte era el de la dreta.
- **first:** Un booleà que ens diu si l'estímul escollit per el subjecte era el primer en aparèixer.
- **performance:** Aquest valor només és diferent a *NaN* en els últims intents de cada episodi, aquesta columna ens diu la proporció de recompensa aconseguida en comparació al màxim que es podia obtenir seguint l'estratègia òptima.
- **stimuli:** Recompensa immediata aconseguida per el subjecte en aquest intent.
- **presented_stimuli:** Parell de estímuls presentats a el subjecte, posicionats de la manera que estaven les columnes en el intent.

Sabent l'estructura tant de les dades com de l'experiment, sabem que cada subjecte té un total de 600 entrades en la taula, de les quals 100 etiquetades amb *horizon* igual a 0, 200 amb *horizon* igual a 1 i 300 amb *horizon* igual a 2. Cada entrada a la taula representa un intent del subjecte, de manera que les 100 primeres entrades representen els intents de la primera etapa de l'exercici, ja que està composada per 100 episodis de 1 intent cadascun, les 200 següents representen els de la segona etapa, ja que tornen a ser 100 episodis, però cada episodi està composat de 2 intents, i les 300 últimes representen els de la tercera etapa, ja que són 100 episodis de 3 intents cadascun.

Amb aquest conjunt de dades tenim tota la informació que necessitem per comprendre cada intent, amb la columna de *subject* podem saber quin és el subjecte del que pertany aquella entrada en concret, amb les columnes de *horizon*, *nte* i *presented_stimuli* podem veure quin és l'estat en el que es trobava l'entorn en aquell moment, amb les columnes de *big*, *right* i *first* es pot saber quina és l'acció que va prendre el subjecte en aquell moment i amb la columna *stimuli* podem saber la recompensa que es va donar. Per últim també tenim la columna de *performance*, que no ens dona informació d'un intent en concret, sinó que ens dona el rendiment del subjecte en l'episodi en general, el que amaga informació dels intents individuals, però dona un bon resum de quan bé que ho ha fet en aquell episodi.

3.3 Caracterització dels subjectes

La caracterització dels subjectes és essencial per tal de poder recrear el seu comportament de forma fiable, per això en aquest apartat mirarem a les dades per tal

de veure com podem simplificar el comportament del subjecte en un descriptor del qual puguem determinar com de similars són dos comportaments.

El primer que es pot fer és mirar a les dades tal i com estan, per tal de buscar possibles patrons en el comportament dels subjectes que ens puguin servir per fer la seva caracterització. En la figura 5 es grafiquen les tres mètriques que descriuen les accions preses pels subjectes.

El primer que es pot veure és que tots els subjectes tendeixen a no considerar ni el costat ni l'ordre d'aparició de l'estímul a l'hora d'escollir, ja que la proporció en els dos casos és sempre propera al 50%. En canvi, es pot veure que la proporció d'estímuls grans escollits no es manté 50% al llarg dels experiments com fan les dues altres mètriques, el que ens fa pensar que els subjectes han escollit quasi totes les vegades en base a la mida dels estímuls i no en la seva posició o ordre en el que apareixen. Amb això podem fer una simplificació a l'hora de caracteritzar ja que, pel que es veu en les dades, sembla que els subjectes no consideraven escollir l'estímul per un altre criteri que no fos la seva mida.

Sabent l'estructura de l'experiment, podem pensar com estaria representada l'estratègia òptima en forma d'histogrames. Tant la proporció que representa el costat com l'ordre d'aparició de l'estímul escollit oscil·laria al voltant del 50%, el que faria que les columnes de l'histograma tinguessin una altura al voltant de 25. En canvi, en el cas de la mida de l'estímul escollit hauria d'anar variant al llarg del temps, a mesura que va canviant l'horitzó dels episodis:

1. Per la primera etapa de l'experiment, amb horitzó de 0 i composta de 100 intents, s'hauria de veure que les barres de l'histograma arriben a la màxima alçada, indicant que sempre s'escull l'estímul més gran.

2. Per la segona fase, amb horitzó de 1 i composada per 200 intents hauríem de veure que s'escull la meitat de les vegades l'estímul més gran, degut a que l'estratègia òptima en aquest cas és escollir una vegada l'estímul més petit per, en el següent intent escollir el més gran.

3. Per la última fase, amb horitzó de 2 i composada per 300 intents, la proporció hauria de quedar en que un terç dels estímuls escollits eren els més grans degut a que per arribar a la major recompensa amb horitzó igual a 2 s'han d'escollir els estímuls més petits en els dos primers intents de l'episodi, per després escollir el més gran en l'últim.

Una cosa que cal recalcar a l'hora de mirar aquestes mètriques és que un subjecte que segueixi aquestes proporcions no vol dir que estigui seguint l'estratègia òptima,

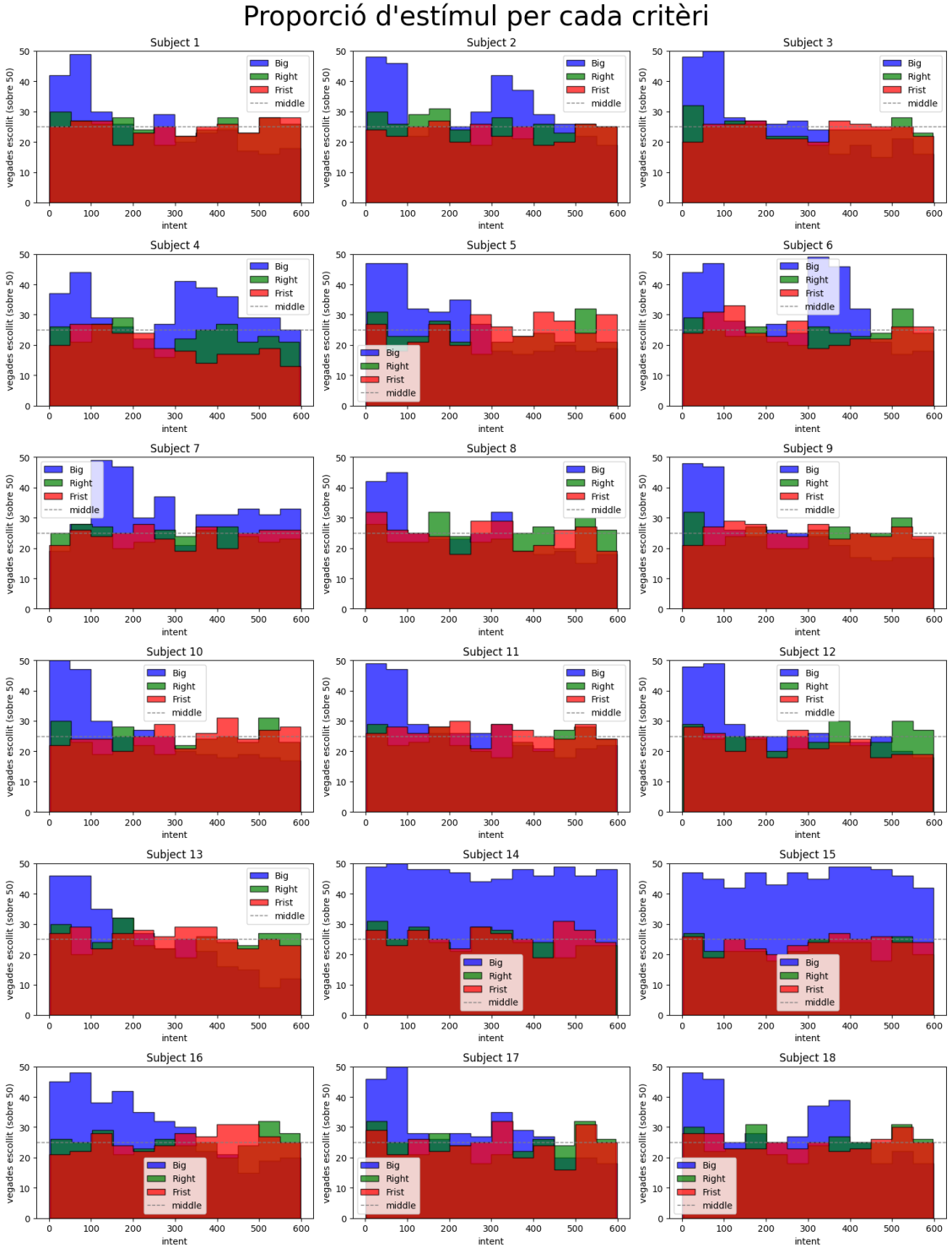

Figura 5: Histogrames del nombre d'estímuls grans escollits (blau), número d'estímuls escollits que estaven a la dreta de la pantalla (verd) i número d'estímuls escollits que van aparèixer primer (vermell). Cada columna de l'histograma conté la mostra de 50 intents, de manera que si la columna té una alçada de 50, vol dir que tots els estímuls que ha escollit aquell subjecte són d'aquell tipus.

ja que podria tenir un proporció del 33% d'estímuls grans escollits en la tercera etapa, però que estigui escollint l'estímul gran en el primer intent de l'episodi i després que esculli en els altres dos intents els estímuls més petits, el que no donaria la millor recompensa possible encara donar la mateixa proporció que si ho estigués fent. L'única cosa que es pot discernir amb dades és que, si el subjecte es surt d'aquestes proporcions, vol dir que hi ha hagut alguna acció que ha pres que no ha contribuït a l'estratègia òptima.

Els exemples que ressalten més són el subjecte 14 i 15, els quals sembla que en qualsevol dels casos escollien la major part de les vegades l'estímul més gran, deixant clar que en la majoria dels episodis de les etapes 2 i 3 no hauran aconseguit la major recompensa. També es veuen subjectes com el 2, el 4 o el 6 que, entre la segona i tercera etapa, comencen a escollir més estímuls grans que petits, el que fa pensar que en la part inicial de la tercera etapa hauran tingut pitjors recompenses.

Encara que aquestes dades ens han donat informació rellevant sobre el comportament dels subjectes, podem veure que no podem basar la descripció del comportament completament en aquestes, ja que fer servir el conjunt de dades de sobre si ha escollit l'estímul gran o petit en cada cas pot ser massa poc permissiu, ja que dos comportaments poden ser semblants encara que no segueixin les mateixes accions exactament. Per altra banda, si generalitzem les dades a la proporció d'estímuls escollits en un grup d'intents, tenim el problema que hem explicat abans.

En canvi, si mirem a les dades de la columna de *performance*, podem veure com de bé ho està fent el subjecte en relació a la recompensa que ha aconseguit. Una mètrica que no és tant exacta en el comportament del subjecte, però que dona una informació molt més rellevant a l'hora de determinar el seu rendiment en aquell moment. En la figura 6 podem mirar com evoluciona el seu rendiment en cada episodi al llarg de l'exercici per cada subjecte.

Mirant la figura 6, podem veure que, encara que no és tan exacta en descriure el comportament dels subjectes, encara podem veure el que havíem vist en la figura 5. Podem veure que els subjectes 14 i 15 tenen una baixada en el rendiment a partir de l'episodi 100 i que, quan veiem que el subjecte 2, 4 o 6 se surten de la proporció esperada, també veiem una baixada ocasional del rendiment en aquells episodis.

Ara que hem vist que el conjunt de dades que dona una millor caracterització és la *performance*, ja que dades com la proporció d'estímuls grans escollits ens pot donar amb seguretat quins subjectes no han seguit l'estratègia òptima però no ens pot dir quins subjectes l'han seguit. Podem concloure que fer servir el rendiment com a descriptor del subjecte ens donarà el millor resultat, tenint un vector de 300 elements del qual es pot calcular la distància amb altres per tal de poder mesurar

Figura 6: Gràfics per cada subjecte de la seva *performance* o rendiment a través dels 300 episodis de l'experiment. Cada punt indica el valor de la *performance* en un dels episodis.

la similitud entre les dues estratègies.

Una forma de calcular la distància entre dos vectors pot ser la distància euclidiana, la qual suma les diferències de cadascun dels eixos (elements) del vector de la següent manera:

- Per dos vectors $v_1 = [i_0, i_1, i_2, \cdots, i_n], v_2 = [j_0, j_1, j_2, \cdots, j_n]$
- La distància euclidiana es calcularia de la següent manera: $d = \sqrt{\sum_{k=0}^{n} (i_k - j_k)}$

El problema que pot comportar utilitzar la distància euclidiana en el vector de *performance* és que es calcula element per element, fent que es calculi la diferència de rendiments del mateix episodi. Això encara pot comportar un problema a l'hora de comparar subjectes, ja que pot ser que dos subjectes tinguin una estratègia semblant encara que no facin el mateix en cada episodi. La diferència que hi ha en aquest cas amb el que ens passava amb les dades de proporció d'estímuls grans escollits, és que en aquest cas podem fer servir un mètode de suavitzat per fer que la mètrica de similitud sigui més permissiva.

El mètode que hem escollit és l'aplicació d'un filtre gaussià al vector, una tècnica molt usada en processament d'imatges per treure soroll i per a la suavització de senyals digitals. Aquesta tècnica consisteix en fer una convolució del vector fent servir un nucli gaussià per la ponderació dels valors.

Una convolució en un vector unidimensional consisteix en aplicar, en aquest cas un nucli gaussià, a cada posició del vector, de manera que es va lliscant el centre del nucli a la posició que es vol calcular i després es fa la suma ponderada seguint els pesos del nucli. El nucli gaussià és creat amb la funció gaussiana, la qual té una forma de campana, com es pot veure en la figura 7, de manera que els valors més propers a la posició del vector que estem calculant tindran un pes més alt, mentre que com més lluny de la posició, menor és el pes que tindrà.

El grau de suavització del resultat del procés es pot controlar fent servir el paràmetre σ, de manera que com més alt és aquest valor, més suavitzades queden les dades. Per el nostre cas, hem vist que $\sigma = 5$ ens dona els millors resultats. En la figura 8 podem veure el resultat de l'aplicació del filtre.

D'aquesta manera, no només es veu millor el progrés de la *performance*, sinó que la mesura de la distància euclidiana és més significativa a l'hora de calcular la similitud degut a aquesta pèrdua de possibles diferències de *performance* puntuals. Així es poden veure les diferències entre calcular la distància euclidiana a la *performance* sense passar per el filtre gaussià i passant-lo. Si mirem el *heatmap* de les

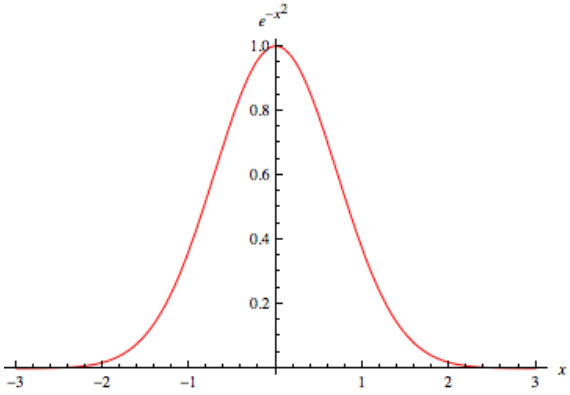

Figura 7: Exemple de fórmula gaussiana

distàncies entre subjectes a la figura 9, podrem veure aquesta diferència.

Com es pot veure en la figura 9, el patró de diferències no sembla canviar entre fer servir un i l'altre, però es pot veure que quan s'utilitza la *performance* sense suavitzar es genera una mena de soroll, de manera que les distàncies no baixen del valor de 6, m'entres que fent servir les performances suavitzades, el rang de distàncies es molt més ampli i fa que els valors baixos siguin més fiables.

3.4 Disseny i implementació de l'agent de Q-learning

A l'hora d'implementar l'agent de Q-learning, tenia clar que la prioritat d'aquest agent sería l'adaptabilitat i versatilitat davant a modificacions de l'entorn, ja que esperava poder fer canvis en aspectes de l'entorn, com la descripció dels estats, sense haver de tornar a modificar l'agent per aquell cas en concret.

Per això, l'agent no fa servir una matriu com a Q-table, sinó que fa servir un diccionari de python, el qual fa que l'agent no posi cap mena de restriccions als estats possibles ni a les accions que es poden pendre, ja que cada Q-valor és una entrada diferent en el diccionari.D'aquesta manera, la única restricció que té a l'hora de identificar els estats i les accions és que han de ser valors de tipus immutables, és a dir, qualsevol tipus de número, *strings* o tuples que estiguin composades per valors de tipus immutable, aquesta definició es deu a que es fa servir una tupla del parell estat-acció com a *key* del corresponent Q-value. Aquesta implementació també va molt bé per entorns que tinguin un gran nombre d'estats, la majoria dels quals l'agent de Q-learning no visitarà, ja que estalvia memòria creant espai per només per els parells estat acció que l'agent hagi fet servir.

La gran adaptabilitat d'aquesta implementació de la Q-table també pot comportar problemes ja que l'agent no té cap mena de control d'errors, de manera que

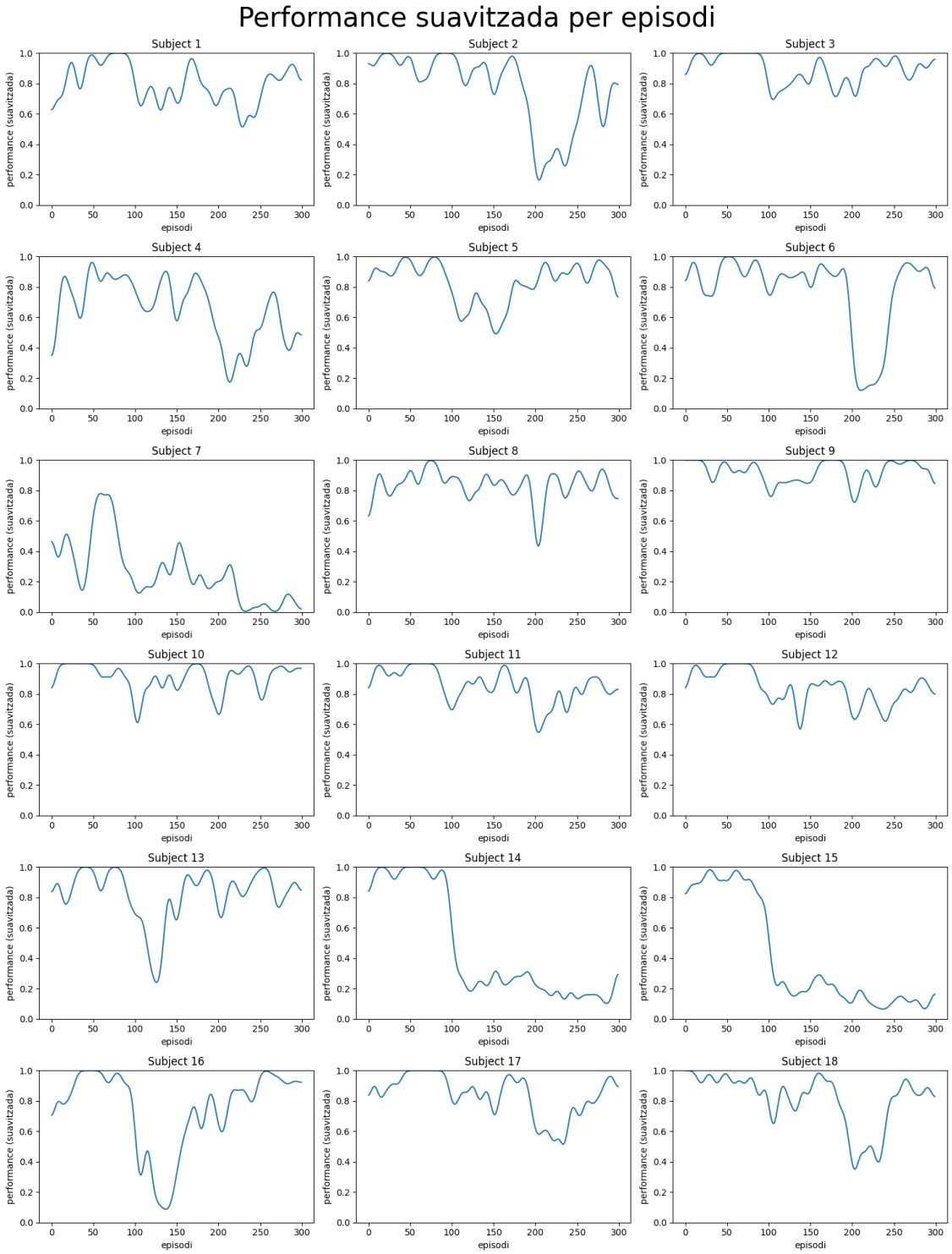

Figura 8: Gràfic que indica la *performance* suavitzada en cada subjecte, indicat per la línia blava, la *performance* està representada en l'eix Y i l'episodi en l'eix X.

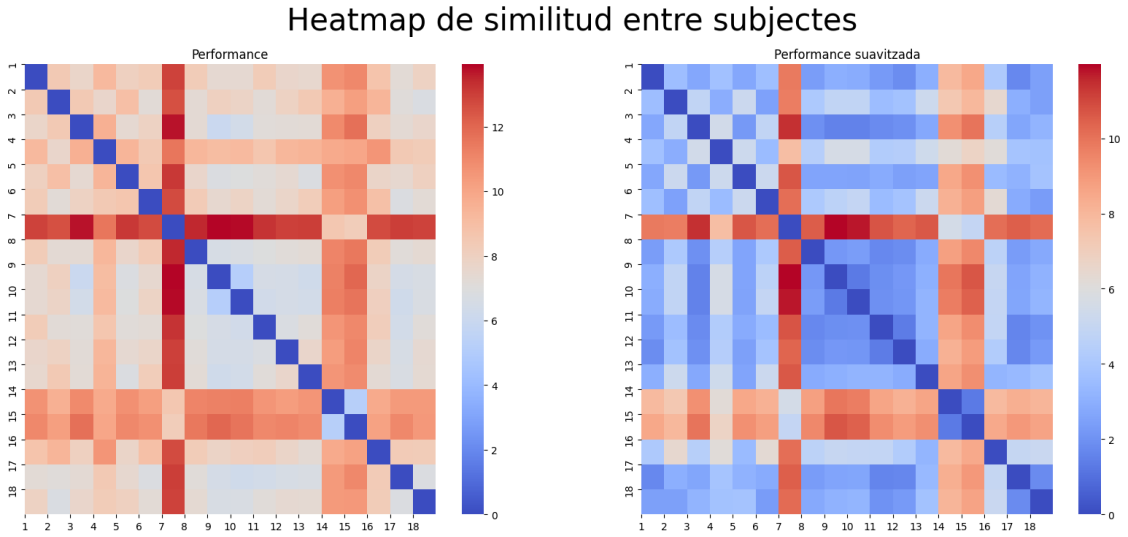

Figura 9: *Heatmap* o mapa de calor que ens dona la distància entre un parell de subjectes fent servir la *performance* sense suavitzar (esquerra) i fent servir la *performance* suavitzada (dreta), un color més càlid o vermellós indica una major diferència i un color més fred o blavós indica una menor diferència. La diagonal que es genera de caselles de color blau és perquè està calculant la distància amb el mateix subjecte, donant una distància de 0.

si se li passa una acció que no existeixi o un estat incorrecte l'algorisme seguirà l'execució. Per això s'ha d'assegurar que l'entorn on es faci aprendre l'agent no pugui donar cap dada incorrecta.

Per mantenir un equilibri entre l'exploració i l'explotació, l'agent fa servir la política ϵ-greedy, ja que l'experiment en si és bastant senzill i no considero que una política més complexa tingués una repercussió significativa en els resultats. Una modificació que hem fet en aquest aspecte és l'afegiment d'un paràmetre que serà un coeficient per el que es multiplicarà el factor d'exploració ϵ al final de cada episodi, de manera que es pot fer que el paràmetre ϵ varii amb el pas del temps, fent que l'agent pugui variar la proporció d'accions preses per explotació i exploració al llarg de l'exercici. Si el coeficient es deixa a 1, no té cap efecte, però si es posa un valor diferent de 1, fa que el factor d'exploració vagi variant al llarg de l'entrenament. Si es fa que sigui un valor inferior a 1, el factor de exploració disminuirà, fent que l'agent cada vegada explori menys. Els valors del coeficient de ϵ són útils en problemes estàtics, en els que la estratègia òptima és la mateixa al llarg del temps, fent que quan ja porti molts episodis i els Q-valors ja siguin bones aproximacions dels valors reals, deixarà quasi per complert d'explorar.

Per últim, també vam afegir el paràmetre biases o biaixos, el qual no és un

valor, sinó que consisteix en un valor associat a cada acció, sent aquest valor el Q-value inicial de tots els parells estat-acció de l'acció corresponent. Si poses un valor negatiu a una acció en concret, l'agent al principi no escollirà aquestà acció tant sovint, ja que tindrà un Q-value més baix que la resta i només l'escollirà per exploració. El contrari passarà si es posa un valor positiu alt. Això té com a objectiu replicar els biaixos que poden tenir els subjectes a causa de preconcepcions que hagin pogut fer per experiències fora de l'experiment. Per exemple, un subjecte pot tenir un biaix en escollir sempre l'estímul més gran degut a la preconcepció de que els intents dins de l'episodi són independents.

3.5 Transfer learning en Q-learning

Un altre aspecte que cal replicar del comportament dels subjectes és que un subjecte pot fer servir la informació aconseguida en l'etapa anterior per saber quina estratègia òptima en la actual, encara que hagi canviat el número d'intents per episodi entre una i l'altra, per això farem servir la tècnica de *transfer learning*.

El *transfer learning* que es pot fer en un agent de Q-learning és principalment la transferència de les dades de la Q-table d'un experiment a un altre, de manera que els Q-valors de la taula generada en l'anterior exercici siguin útils en el següent. En el cas d'aquest treball, la transferència es farà entre les tres etapes de l'experiment, la primera amb un horitzó igual a 0, la segona amb un horitzó igual a 1 i la última amb horitzó igual a 2[Figura 10].

El primer que podem veure en la figura 10 és com l'arbre de decisions d'una etapa és igual a la branca esquerra de l'arbre de decisions de l'etapa següent, fins i tot els Q-valors serien proporcionals, ja que l'estratègia òptima en els dos casos en aquells estats és la mateixa. Per això la primera forma de *transfer learning* que podem aplicar és una que, en el canvi d'etapa, es facin servir els Q-valors de l'anterior etapa com a Q-valors dels estats després d'escollir l'estímul petit.

Per implementar-lo, no ha calgut fer cap mètode que modifiqui la Q-table de l'agent per fer la transferència de la informació, sinó que simplement amb el fet de la Q-table del nostre agent no es limita a uns estats en concret, podem identificar els estats de cada etapa d'una manera que quan es canvii d'etapa, els Q-valors de l'anterior passin a ser automàticament els de la part que ens interessa.

Per fer això hem dissenyat la següent forma d'identificar els estats:

Tenint l'horitzó de l'episodi h i la profunditat actual de l'intent d:

- L'estat inicial serà el valor $s_0 = 2^{h+1}$

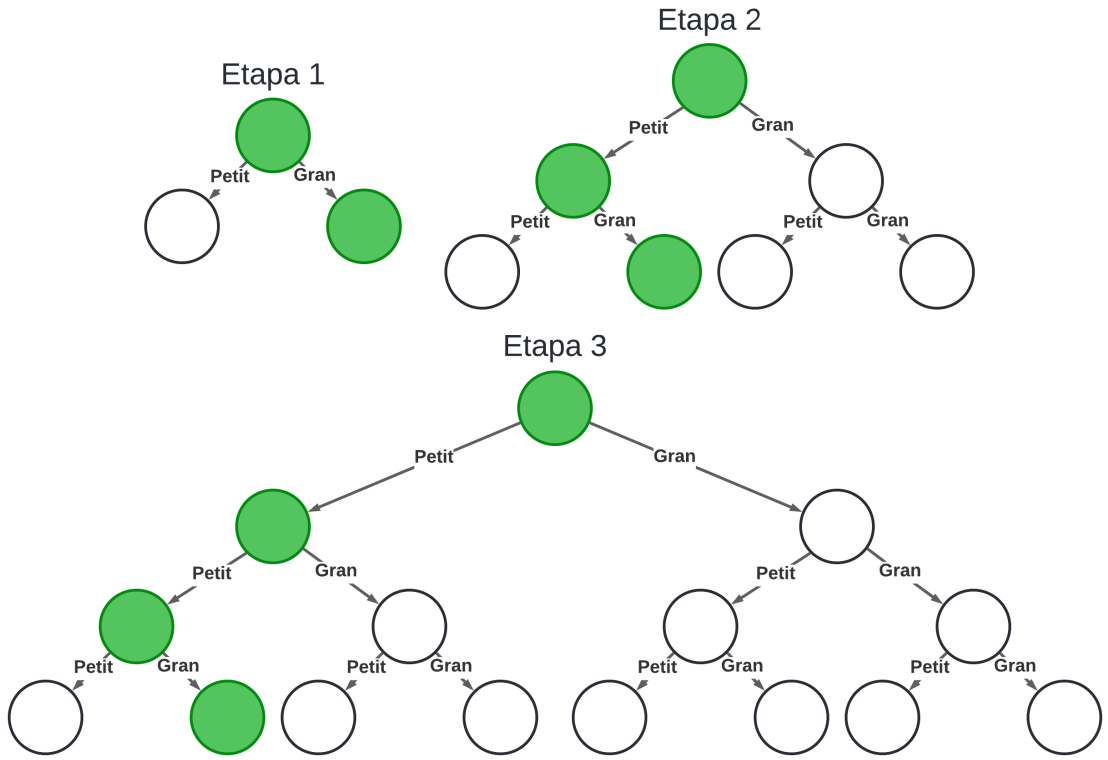

Figura 10: Arbre de decisions d'un episodi en cada etapa de l'experiment on els nodes representen els estats i les línies la transició entre estats al prendre una decisió (les etiquetades com *Gran* indiquen les transicions en cas de que l'agent esculli l'estímul més gran, i les etiquetades com *Petit* la transició en el cas de que s'esculli l'estímul petit). Els nodes verds indiquen quin és el camí de decisions que l'agent ha de seguir per aconseguir la màxima recompensa en cada cas.

- L'estat següent, en el cas que s'esculli l'estímul petit, es calcularà $s_{d+1} = s_d - 2^{h-d}$, en canvi si s'escull el estímul gran, es calcularà com $s_{d+1} = s_d + 2^{h-d}$

D'aquesta manera, la identificació dels estats en els tres casos quedaria com a la figura 11.

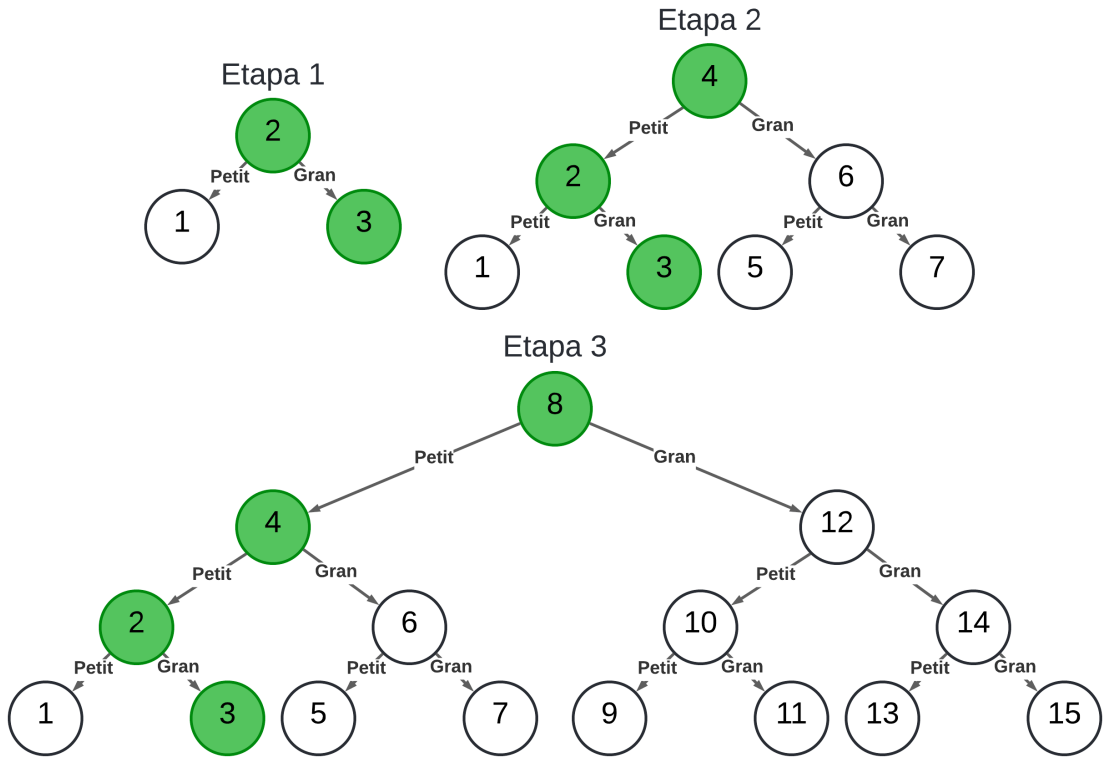

Figura 11: L'estructura és igual a la de la figura 10, l'única diferència és que aquí es mostren com cada estat estarà identificat a la Q-table de l'agent de manera que pugui aprofitar els Q-valors d'una etapa en les següents amb un major horitzó.

D'aquesta manera, quan es passa d'horitzó 0 a 1, els Q-valors sobre l'estat 1, 2 i 3 estimats en l'etapa anterior, es podràn aprofitar en el cas amb horitzó 1, ja que en aquells tres estats, la millor decisió seguirà sent triar l'estímul gran. Aquest canvi li dona a l'agent una millor adaptabilitat a canvis en l'horitzó, ja que no té perque tornar a començar amb una Q-table buida.

Encara així, no és difícil veure a la figura 11 que igual que estem aprofitant els Q-valors de l'etapa anterior en els de la branca de l'esquerra, podríem fer-los servir per els de la branca de la dreta també, ja que les decisions òptimes són les mateixes. Per exemple, l'estratègia òptima en els estats 1, 2 i 3 és la mateixa que en els estats 5, 6 i 7, de manera que es poden aprofitar els mateixos Q-valors i copiar-los a la branca de la dreta, fent que, en el moment que es canvia d'etapa, els únics Q-valors

que no aprofitaran dades de etapes anteriors seran les dues accions possibles en l'estat inicial.

Amb aquests mètodes de *transfer learning* veurem en l'apartat de resultats com donen millores considerables sobre com els agents aprenen i com es suavitza la baixada de *performance* en els canvis d'etapa.

Capítol 4

Resultats

4.1 Agrupació de subjectes amb K-means

En aquest apartat farem servir l'algorisme K-means per tal de generar un sistema de grups per els subjectes basant-se en el seu comportament, representat per la *performance* suavitzada que hem explicat anteriorment.

A l'hora de buscar quin valor de k ens anirà millor per el nostre problema, fem servir l'*elbow method* per trobar-lo en un rang entre 0 i 10. El resultat es mostra en la figura 12.

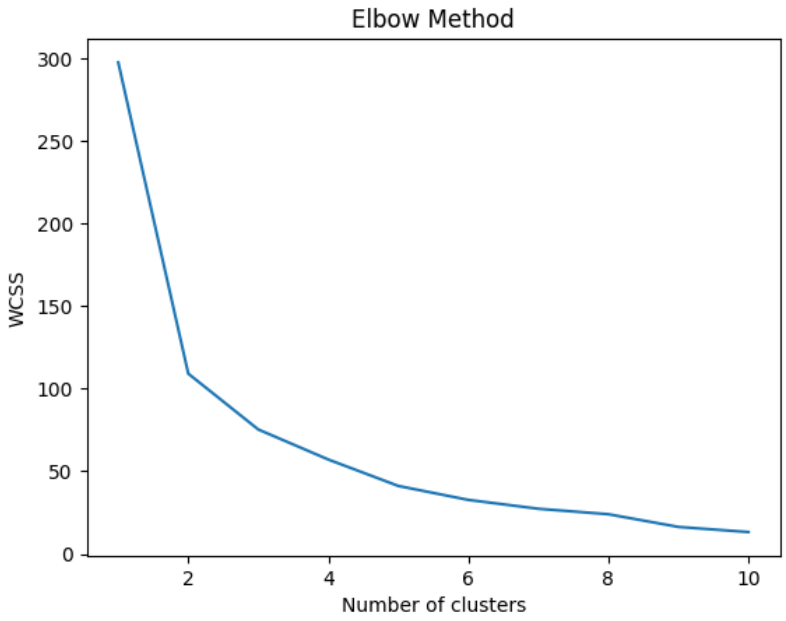

Figura 12: Gràfic resultant d'aplicar l'*elbow method*. En l'eix X està representat el número de clústers k i en l'eix Y està representat el WCSS o inèrcia del resultat de la clústerització.

Com podem veure, el WCSS entre elements i el seu centroide comença a deixar de disminuir entre el número de clústers 2 i 3, generant aquesta forma de colze que li dona nom al mètode. Per fer l'agrupació hem decidit fer servir $K = 3$. El resultat que dona són els tres grups següents:

- Grup 0: Composat per els subjectes 7, 14 i 15.

- Grup 1: Composat per els subjectes 1, 3, 5, 8, 9, 10, 11, 12, 13, 16, 17 i 18.
- Grup 2: Composat per els subjectes 2, 4 i 6.

A la figura 13 mostrem els rendiments suavitzats de cada subjecte en cada un dels grups.

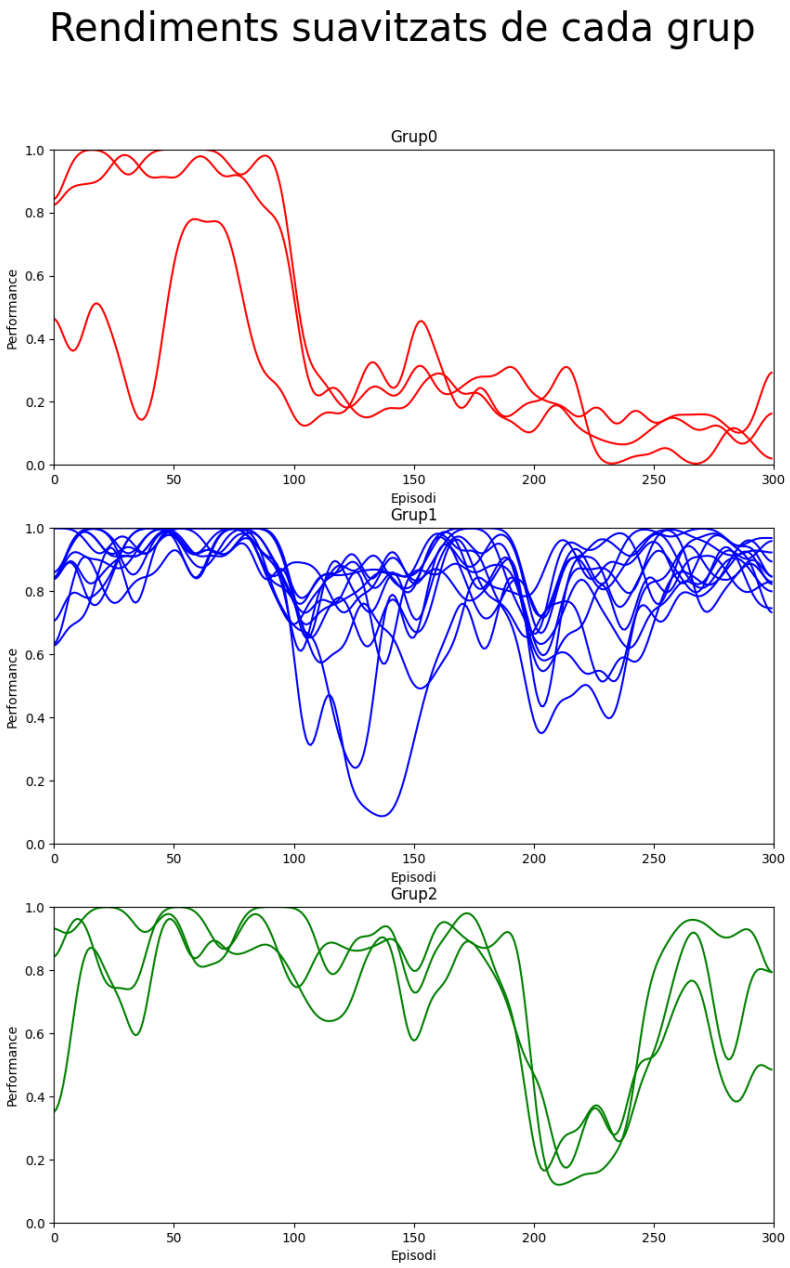

Figura 13: Gràfic mostrant totes les *performances* suvaitzades de cada subjecte colorejades depenent del grup assignat per l'algorisme K-means. Les línies de color vermell corresponen als subjectes del grup 0, les blaves al grup 1 i les verdes al grup 2.

En el primer gràfic de la figura 13, podem veure les *performances* dels subjectes del grup 0. Els rendiments dels subjectes d'aquest grup són les més baixes de tots, tots tres comencen tenint una *performance* relativament alta, però quan es canvia d'etapa baixa significativament fins quasi ser 0. Aquests subjectes ja els hem fet servir com exemple en altres parts del treball, ja que tenen un perfil de rendiment bastant peculiar degut a la seva estratègia d'escollir sempre l'estímul més gran.

Al segon gràfic de la figura 13 podem veure els subjectes del grup 1, els quals tenen un comportament bastant normatiu, sent en la majoria dels casos que no baixen de la *performance* suavitzada del 0.4 en cap moment de l'exercici. Encara així hi ha alguns casos en els que els subjectes han tingut dificultats a l'hora d'adaptar l'estratègia en un canvi d'etapa, però en tots els casos acaben aconseguint trobar l'estratègia òptima.

Per últim, tenim el tercer gràfic de la figura 13, en el que veiem les *performances* suavitzades dels subjectes del grup 2. Aquest grup, encara semblar molt similar al comportament dels subjectes del grup 1, té una particularitat, els tres subjectes tenen una baixada bastant significativa en el canvi de la segona etapa a la tercera de l'exercici, en alguns casos, sense arribar ni a tornar a tenir una *performance* propera a 1. Del comportament d'aquests subjectes també n'hem parlat quan comentavem la figura 5, ja que ressaltava la seva desviació en la proporció d'estímuls grans triats.

Aquestes agrupacions ens han descrit principalment quina és la norma en el comportament dels subjectes, sent aquesta el comportament dels subjectes del grup 1, i quins subjectes surten d'aquesta norma, sent els del grup 0 i 2. Aquesta informació ens servirà en l'estudi dels següents resultats, donant-nos una idea de quins comportaments comuns entre subjectes hauríem de veure.

4.2 Diferències entre tipus de *transfer learning*

Tenim un total de tres formes d'afrontar la transferència del coneixement previ de l'agent entre fases. Podem restablir la taula, fer servir els valors de l'anterior etapa en la part esquerra de la taula, a la que ens referirem a partir d'ara com adaptació parcial, o podem fer servir les dades de l'etapa anterior en les dues parts de l'arbre de decisions que encaixen amb l'estratègia, a la que ens referirem com adaptació complerta. Per provar les diferències que tenen en el comportament farem que un agent genèric faci un experiment complert i mirarem la seva *performance* en els tres casos. Els paràmetres que tindrà aquest agent seran:

- $\epsilon = 0.3$

- $\epsilon - decay = 1$
- $\gamma = 0.9$
- $\alpha = 0.4$

L'agent no tindrà cap mena de biaixos i les accions que podrà fer seran escollir l'estímul gran o el petit, ja que hem vist que els subjectes no feien cas ni a la posició ni a l'ordre d'aparició.

Començant per l'agent que no fa servir cap mena de *transfer learning*, podem veure la seva *performance* en la figura 14 com els canvis d'etapa causa una baixada considerable de la *performance*. També podem veure com en cada fase té una *performance* pitjor que en l'anterior a causa de l'increment de la complexitat del problema, fins al punt en que a la tercera etapa, es pot veure que quasi no aconsegueix fer cap episodi en el que aconsegueixi la millor recompensa possible, Aquesta prova ens demostra que intentar fer un model de tots els subjectes sense fer servir *transfer learning* no donarà molts bons resultats, ja que hem vist a la figura 8 hi ha molts subjectes que no tenen tants problemes a l'hora de passar de etapa.

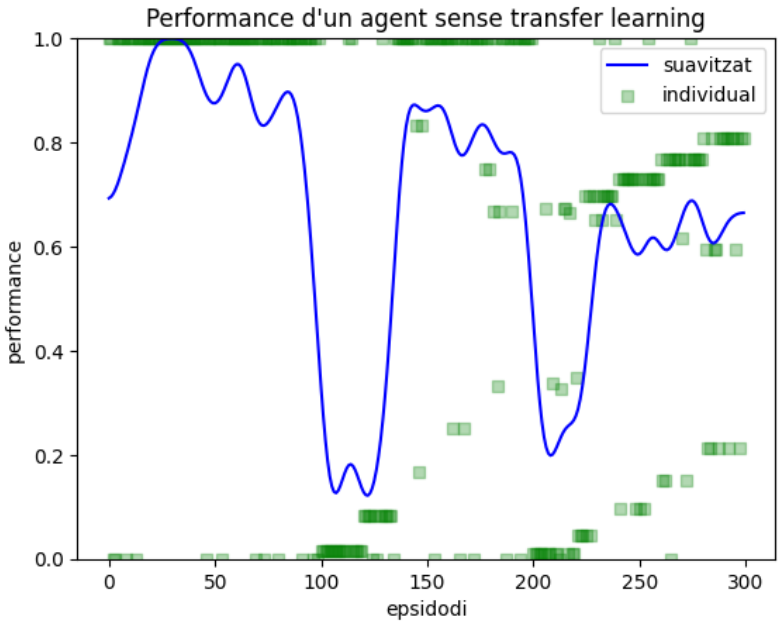

Figura 14: Gràfic de la *performance* suavitzada (línia blava) i la no suavitzada (punts verds) en un agent que no fa servir cap mena de *transfer learning* entre les etapes de l'exercici.

Vist que el comportament de l'agent sense transferència de dades, a continuació provarem el mateix amb un agent que farà una adaptació parcial. Podem veure a

la figura 15 com el canvi entre no fer servir *transfer learning* i fer servir l'adaptació parcial és significatiu. Aquest nou agent aconsegueix trobar l'estratègia òptima en les tres etapes, ja que podem veure una gran quantitat de punts a dalt de tot, indicant que l'agent no només ha aconseguit seguir la seqüència de decisions òptima per casualitat, sinó que l'està explotant de forma activa. També podem veure que les baixades de *performance* es suavitzen molt, sent el canvi de la primera etapa i la segona etapa quasi indistingible.

Un dels problemes que es pot veure en utilitzar aquesta tècnica de transferència parcial, és casos com els del canvi de la 2a etapa a la 3a, on hi ha una baixada molt pronunciada. Això es deu a que, si l'agent en el canvi de fase comença escollint l'estímul gran, que és el que té un Q-valor més alt al principi, ja que la seva recompensa immediata és major, fent que entri al que consideraríem com la branca dreta de l'arbre de decisions del que parlavem en la figura 11, la qual no té els Q-valors de l'etapa anterior, fent que segueixi escollint l'estímul gran fins que per exploració provi d'escollir l'estímul petit al principi d'un episodi, fent que llavors trobi l'estratègia òptima quasi immediatament.

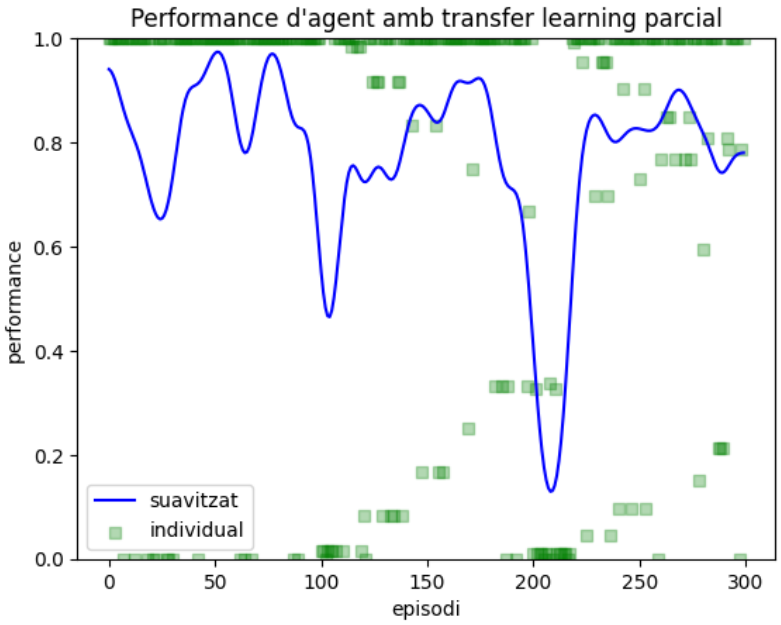

Figura 15: Gràfic de la *performance* suavitzada (línia blava) i la no suavitzada (punts verds) en un agent que fa servir una adaptació parcial entre les etapes de l'exercici.

Veient que l'adaptació parcial dona bons resultats però amb algunes baixades ocasionals en els canvis de etapa, ara provarem com es comporta un agent que fa una adaptació complerta en cada canvi d'etapa. Podem veure a la figura 16 com el que abans ens ha passat es suavitza, deixant una baixada de la *performance*

entre l'etapa 2 i 3 molt més suau, ja que, encara que l'agent esculli primer l'estímul més gran, la resta de decisions ja tenen associada un Q-valor el qual farà que la *performance* en els primers episodis de la tercera etapa no sigui zero, sinó que, com podem veure amb els punts, es queda al voltant del 0.38.

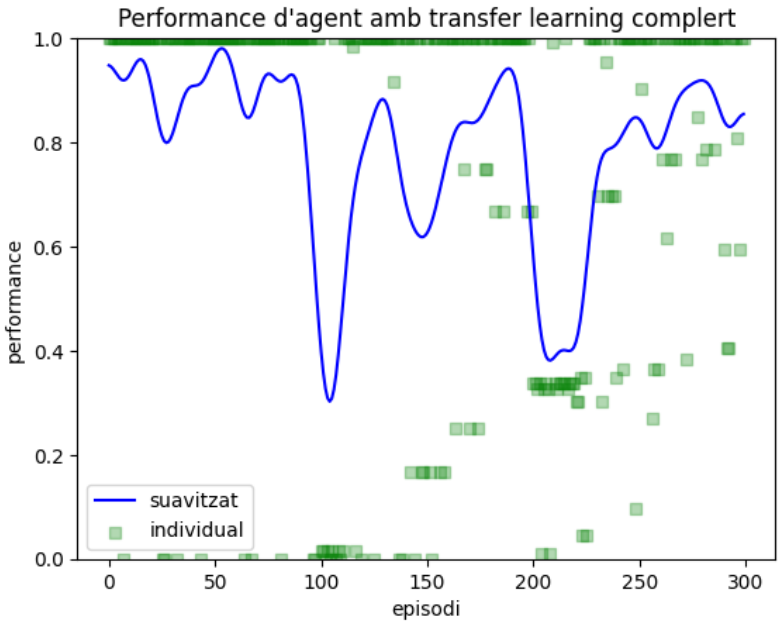

Figura 16: Gràfic de la *performance* suavitzada (línia blava) i la no suavitzada (punts verds) en un agent que fa servir una adaptació complerta entre les etapes de l'exercici.

Ara que ja tenim una bona idea de l'efecte que tenen aquestes diferents tècniques en el procés d'aprenentatge de l'agent podem comparar el comportament d'aquests agents amb els subjectes. Una forma ràpida sería executar tres conjunt de 100 agents, cada conjunt fent servir una de les tècniques, i veure en quin grup el algorisme K-means ja entrenat els posiciona, donant-nos una idea aproximada de a quin grup pertany el comportament causat per cada una de les tècniques.

Per als agents que no aprofiten les dades entre fases de l'experiment, el resultat que ens dona és que un 67% de les vegades l'algorisme categoritza a un agent d'aquest tipus com del grup 2, un 22% com del grup 1 i un 11% com del grup 2. En canvi si s'executen 100 agents que s'aprofiten de les dades de la manera en la que hem explicat, el 78% dels agents executats es categoritzen com del grup 1 i un 22% és categoritzat en el grup 2. Finalment, si fem el mateix amb agents que adapten totalment la Q-table en el canvi de fase, el K-means categoritza al 75% dels agents al grup 1, el 23% al grup 2 i el 2% al grup 0.

D'aquesta manera podem veure com fer servir el *transfer learning* pot tenir efec-

tes significatius en el comportament de l'agent i que no tots els subjectes s'assimilen en comportament a la mateixa tècnica d'adaptació de les dades.

4.3 Ajustament d'hiperparametres

Ara que ja hem estudiat tots els factors que poden afectar al comportament de l'agent de manera que s'assimili a com els subjectes han après, ara podem fer servir l'ajustament d'hiperparametres per tal de buscar quin conjunt de paràmetres s'ajusta més al comportament de cadascun dels subjectes. També estudiarem els valors que ens doni l'algorisme d'ajustament d'hiperparàmetres fent servir l'agrupació que hem fet anteriorment fent ús de l'algorisme K-means per buscar possibles similituds entre elements del mateix grup o diferències entre elements de grups diferents.

Per fer l'ajustament d'hiperparàmetres hem fet servir l'algorisme TPE, ja que l'espai d'hiperparàmetres és massa gran per fer servir una cerca en graella i perquè sabem que el TPE és capaç de gestionar funcions d'objectiu complexes. L'espai de paràmetres que considerem que dona els millors resultats ha estat el següent:

- La taxa d'aprenentatge es prova des de 0.001 fins a 1.
- El factor de descompte es prova des de 0.1 fins a 1.
- El factor d'exploració es prova des de 0.01 fins a 1.
- El progrés del factor d'exploració es proven els valors 0.99, 0.995, 0.999 i 1.
- El biaix cap a l'acció d'escollir l'estímul més gran es proven els valors de 0, 2 i 5.
- Es provarà a activar o desactivar l'adaptació complerta de la Q-table en els canvis d'etapa.
- Es provarà a activar o desactivar l'adaptació parcial de la Q-table en els canvis d'etapa.

Juntament amb aquest espai de paràmetres i fent servir la distància euclidiana entre rendiments suavitzats que expliquem en l'apartat de *caracterització del comportament*, podem generar un agent amb aquests paràmetres perquè entreni fent l'experiment per després, amb el vector de rendiment suavitzat, calcular la distància euclidiana entre el generat per l'agent i el del subjecte. Per tenir una millor mesura d'aquesta similitud, aquest procés es fa un total de 4 vegades i es fa servir com a similitud definitiva la mitjana entre aquestes quatre. Això evita que

la similitud es senti afectada tant significativament per possibles factors aleatoris en el procés d'aprenentatge de l'agent generat. Els resultats que ha donat aplicar aquest procés per cada subjecte es mostren en la taula 1.

subjecte	big_bias	clear_table	γ	eps_prog	ϵ	full_adaptation	α
1	2	True	0.785	1.000	0.464	False	0.989
2	2	True	0.410	0.995	0.280	False	0.679
3	2	False	0.410	0.995	0.303	True	0.991
4	2	True	0.899	1.000	0.361	False	0.374
5	5	True	0.494	1.000	0.253	False	0.710
6	2	False	0.381	0.990	0.616	True	0.998
7	2	False	0.469	0.995	0.993	False	0.002
8	2	False	0.651	1.000	0.302	True	0.993
9	5	True	0.496	0.999	0.217	False	0.9941
10	5	True	0.438	1.000	0.249	False	0.7191
11	5	True	0.581	0.999	0.250	False	0.735
12	0	False	0.831	1.000	0.359	False	0.739
13	0	False	0.799	0.999	0.237	True	0.705
14	5	False	0.103	0.999	0.198	False	0.003
15	5	False	0.806	0.995	0.087	True	0.002
16	2	False	0.970	1.000	0.333	False	0.251
17	5	True	0.726	0.995	0.285	False	0.940
18	2	True	0.922	0.999	0.469	False	0.852

Taula 1: Taula de paràmetres òptims per cada subjecte segons l'ajustament d'-hiperpàrametres, cada columna ens indica quin valor ha considerat l'algorisme de TPE que dona un major resultat en la recreació del subjecte corresponent. La primera columna ens indica el subjecte al que vol replicar, la segona ens indica biaix a escollir els estímuls grans, la tercera si no fa servir cap mena de *transfer learning*, la quarta el factor de descompte, la cinquena el coeficient de el factor d'exploració, la sisena el factor d'exploració, la setena si fa servir una adaptació complerta de les dades i la vuitena la taxa d'aprenentatge.

Aquests paràmetres mostrats a la taula 1 ens poden donar una descripció bastant complerta del comportament de cada subjecte. Per exemple, si creem un agent de Q-learning amb els paràmetres que ens ha donat l'ajustament d'hiperparàmetres i ho comparem amb el subjecte corresponent podrem veure quines diferències en el comportament tenen (Figura 17). Podem veure com els diferents agents repliquen les diferents estratègies dels subjectes.

Si calculem la mitjana dels paràmetres de la taula 1 ens dona els següents valors:

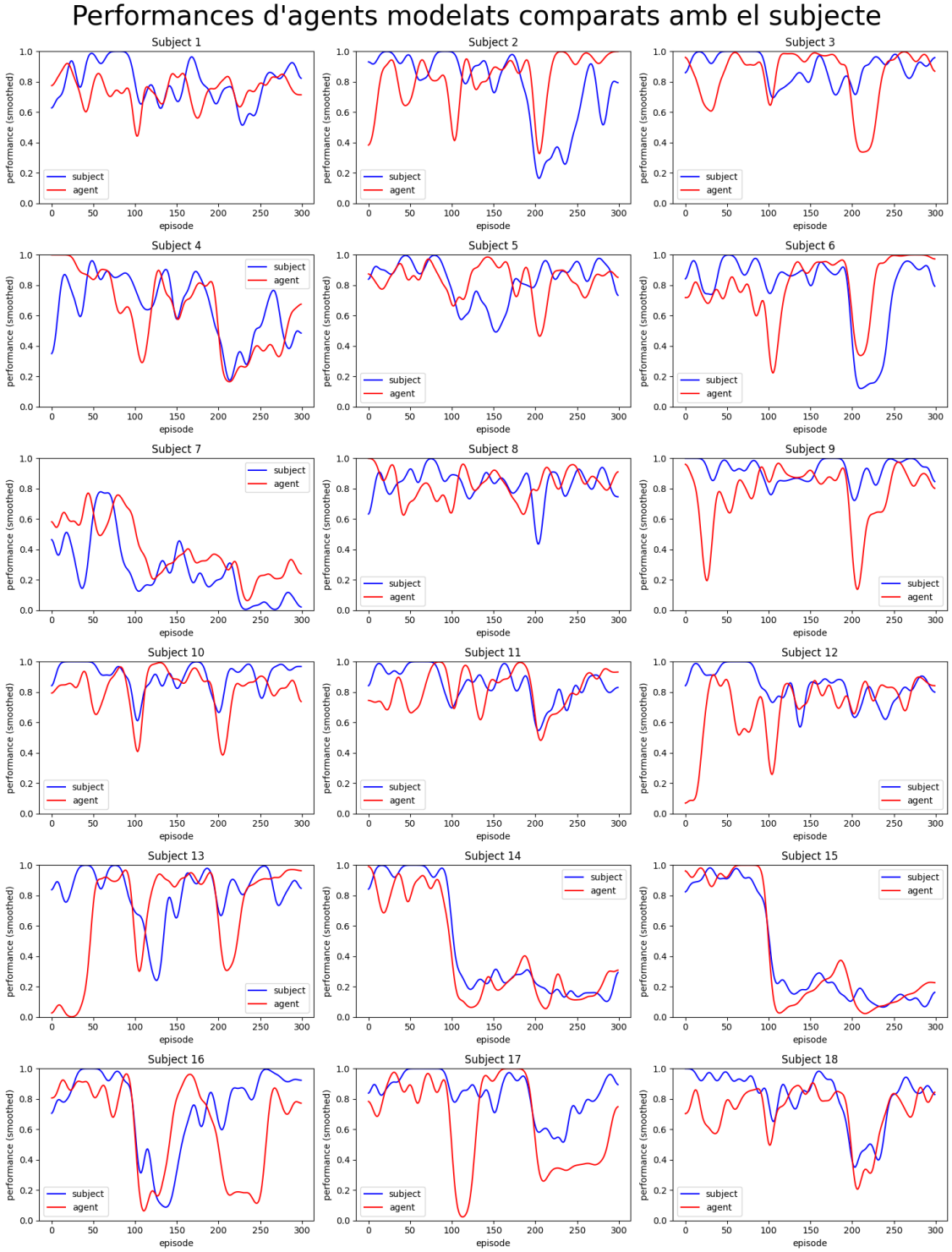

Figura 17: Visualització comparativa de l'evolució de la *performance* entre cada subjecte (línia blava) i l'agent amb els paràmetres determinats per l'ajustament de paràmetres (línia vermella) per replicar el seu comportament.

- $\gamma = 0.649$ amb variància de 0.131: Podem veure que el valor del factor de descompte és relativament baixa, amb la qual cosa ens dona l'idea que els subjectes li donaven molta importància a les recompenses futures, cosa que és normal degut a poca duració dels episodis. També podem veure una variància relativament alta, indicant que hi ha tant subjectes amb una gamma més alta, com alguns amb una encara més baixa.

- $\alpha = 0.621$ amb variància de 0.0554: Aquí podem veure que la taxa d'aprenentatge, en canvi, és un valor relativament alt i amb una variància més baixa que la del factor de descompte, el que fa pensar que els subjectes no esperaven a provar una acció en concret varies vegades per determinar si valia la pena prendre-la o no. Aquest valor també és esperat ja que, encara que els valors de cada episodi tenen un factor aleatori, el comportament era determinista, tenint en tots els episodis el mateix. Aquest paràmetre acostuma a ser més baix en entorns estocàstics o aleatoris, en els quals no es busca l'estratègia que doni la millor recompensa, sinó l'estratègia que dona la millor recompensa més vegades.

- $\epsilon = 0.348$ amb variància de 0.0399: El factor d'exploració té un valor que podria ser considerat moderat en un agent de Q-learning i amb una variància relativament petita, donant a entendre que la majoria dels subjectes han sabut equilibrar la exploració i l'explotació de forma eficient.

- ϵ-progression = 0.998 amb variància de 0.000008: Aquest valor, com era d'esperar degut a que el rang de valors era molt limitat, la variància és molt baixa. El valor de la mitjana, en canvi, és bastant interessant, ja que és el valor intermèdi entre els tres valors possibles, volent dir que més o menys hi ha els mateixos subjectes fent servir cada valor.

També podem mirar les modes dels valors que no són numèrics:

- big_bias = 2: Podem veure que en general, els subjectes tenien una predisposició a escollir l'estímul més gran.

- clear_table: En aquest cas hi ha tants que la fan servir com que no, fent-la l'opció més usada per sobre dels dos tipus d'adaptació.

- full_adaptation = $False$: Podem veure que la adaptació complerta no s'ha fet servir en la majoria dels casos.

Ja tenint tots els valors mitjans o més comuns de cada paràmetre, podem mirar com es comportaria un agent que seguís aquests paràmetres. A la figura 18 podem veure el seu rendiment suavitzat comparat al de tots els altres subjectes.

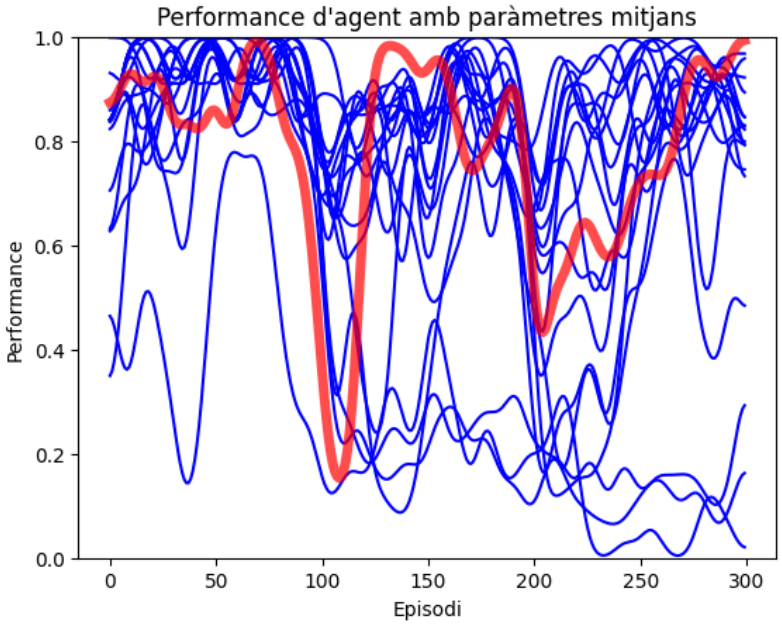

Figura 18: Visualització comparativa de l'evolució de la *performance* entre els subjectes (línia blava) i l'agent amb els paràmetres mitjans de tots els subjectes (línia vermella).

En la figura 18 podem veure com l'agent que hem creat, encara no fer servir cap mètode de *transfer learning*, degut a una taxa d'aprenentatge i un factor de descompte més alt, troba l'estratègia òptima en 100 episodis fins i tot en la tercera etapa de l'exercici. Encara així, es poden veure les baixades pronunciades del rendiment en les dues transicions de etapa, que molts subjectes també van tenir quan feien l'exercici.

El que també podem fer és provar de crear, fent servir l'ajustament d'hiperparàmetres, un agent que s'assembli el màxim possible a tots els subjectes a la vegada, calculant la distància total com la mitjana entre cada subjecte. El resultat d'aquest procés té els següents valors i la visualització del rendiment (figura 19) és la següent:

- $\gamma = 0.705$
- $\alpha = 0.988$
- $\epsilon = 0.2986$
- ϵ-progression$= 1$
- clear_table $= True$

- full_adaptation = $False$
- big_bias = 5

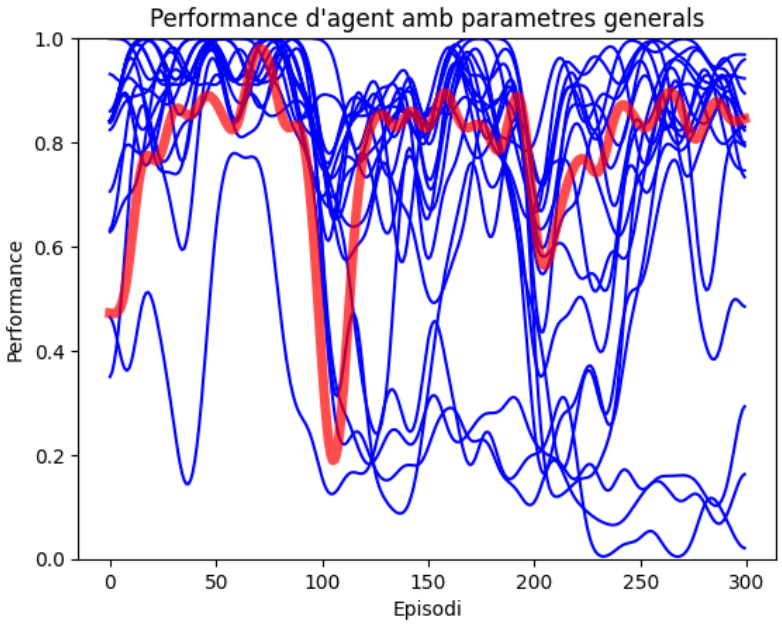

Figura 19: Visualització comparativa de l'evolució de la *performance* entre els subjectes (linea blava) i l'agent amb els paràmetres determinats per l'ajustament de paràmetres (linea vermella) fent servir tots els subjectes com a objectiu.

Podem veure que tant el factor d'exploració com el d'aprenentatge tenen valors molt alts fent que l'agent aprengui molt ràpid l'estratègia òptima malgrat no fer servir les dades de les etapes anteriors. En la figura 19 podem veure que, encara mostrar baixades en els canvis d'etapes, són més suaus que en el cas de la figura 18 i també cal dir que si s'executa vàries vegades, el comportament de la *performance* varia bastant en cada cas. També podem veure que la *performance* quan no acaba de canviar es manté en una posició que com es pot veure és l'alçada en la que quasi sempre estan la majoria dels subjectes.

Ara que hem vist el comportament dels diferents paràmetres de forma global, podem mirar també com es comporten en cadascun dels grups que ens ha donat l'algorisme K-means abans. En la següent taula podem veure els valors mitjans en cada grup (taula 2).

En el grup 0, podem veure a la taula 2 com la taxa d'aprenentatge pren un valor molt baix, quasi 0. Això es deu a que els subjectes classificats en aquest grup són els que no van aconseguir gaire bons rendiments en l'exercici, i l'algorisme ha trobat que si posa la taxa d'aprenentatge a quasi 0, pot fer que l'agent no aprengui

group	learning_rate	discount_factor	epsilon	eps_prog	big_bias
0	0.002441	0.459210	0.425856	0.996333	4.000000
1	0.801526	0.675272	0.310114	0.998833	2.916667
2	0.683892	0.563590	0.418947	0.995000	2.000000

Taula 2: Taula de la mitjana de cada paràmetre numèric en cada grup.

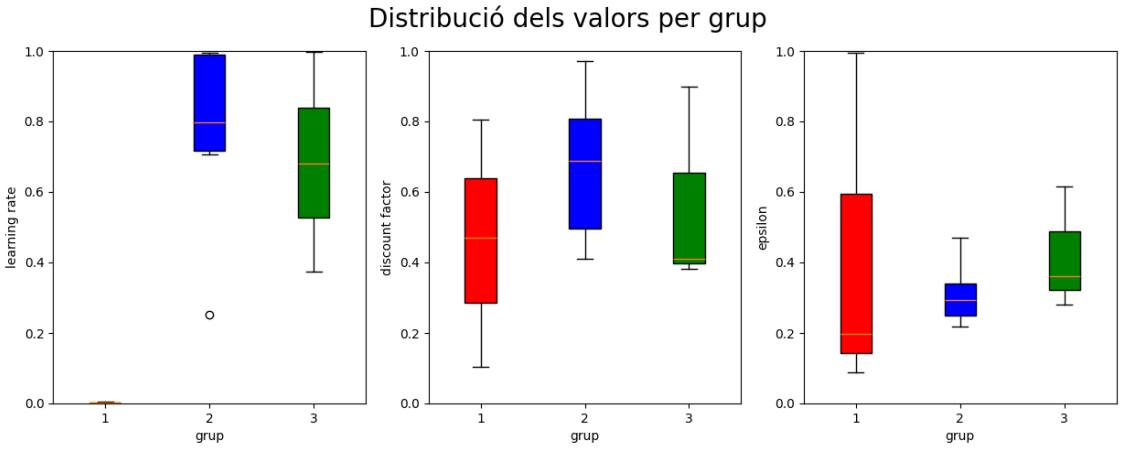

Figura 20: Gràfic de caixes de la distribució de cada paràmetre numèric continu de l'espai d'hiperparàmetres separats per grups.

i es quedi tot l'exercici amb un mala *performance*. Tant el factor de descompte, el factor d'exploració i el coeficient tenen valors molt normals, però com la taxa d'aprenentatge és tan baix, no tenen cap mena d'efecte en el comportament de l'agent. Finalment podem mirar el biaix inicial a escollir els estímuls més grans, que com podem esperar, és el més alt dels tres grups, tenint el subjecte 14 i 15 que tenen un biaix de 5 i el subjecte 7 de 2. Si mirem la proporció de les tres diferents tècniques de *transfer learning* ens dona lo mostrat a la figura 21, encara que passa aproximadament el mateix que en el cas dels paràmetres, que com l'agent quasi no aprèn, no tenen un efecte massa significatiu en el seu comportament.

En el grup 1, podem veure un conjunt de paràmetres més comú i amb una variància relativament baixa pel que podem veure en la figura 20. La taxa d'aprenentatge és relativament alta, degut al que hem explicat de que l'entorn és determinista, el que fa que no calguin varies mostres per veure el valor aproximat de prendre aquella acció. El factor de descompte és el més alt dels tres grups també i el factor d'exploració també està dins dels valors normals en un agent de Q-learning. Si mirem el biaix, encara no ser tan alt com el del grup 0, també té un valor relativament alt.

En aquest grup, l'ús de les diferents tècniques de *transfer learning* sorprenen,

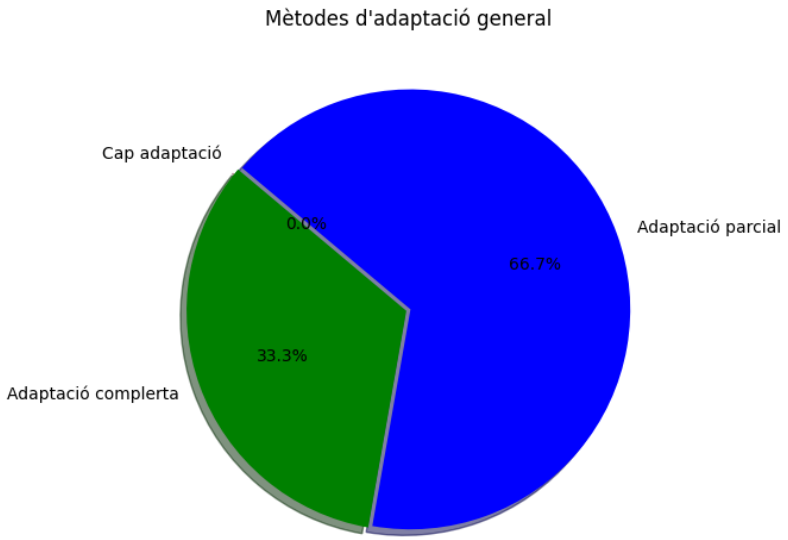

Figura 21: Percentatges d'us de les diferents tècniques de *transfer learning* del grup 0. Un subjecte fa servir l'adaptació complerta i els altres 2 fan servir l'adaptació parcial.

ja que si mirem la figura 22, podem veure que la majoria dels subjectes se'ls hi ha assignat no fer servir cap *transfer learning*, mentre que menys de la meitat fan servir alguna mena d'adaptació entre etapes. Això es deu a que, amb un learning rate tan alt, l'algorisme és capaç de trobar l'estratègia òptima per qualsevol etapa sense necessitat d'informació prèvia.

Per últim, en el grup 2, tenim un taxa d'aprenentatge no tant alt com el del grup 1 i un factor de descompte relativament baix. Tant el factor d'exploració com el seu coeficient i el biaix tenen també valors que no estan fora de lo normal. Si mirem quines tècniques han fet servir d'adaptació a la figura 23, podem veure que només a un dels subjectes se li ha assignat una tècnica de *transfer learning*. Si mirem les performances dels tres subjectes d'aquest grup, podem veure que només el número 6, després de la baixada de *performance* al principi de la tercera etapa de l'exercici, acaba amb una *performance* alta. Justament aquest subjecte és el que se li ha assignat l'adaptació complerta, això sembla que es deu a aquesta pujada de la *performance* en pocs episodis, comportament que hem vist bastant comú en el subapartat de *transfer learning* quan s'aplica l'adaptació complerta de la Q-table.

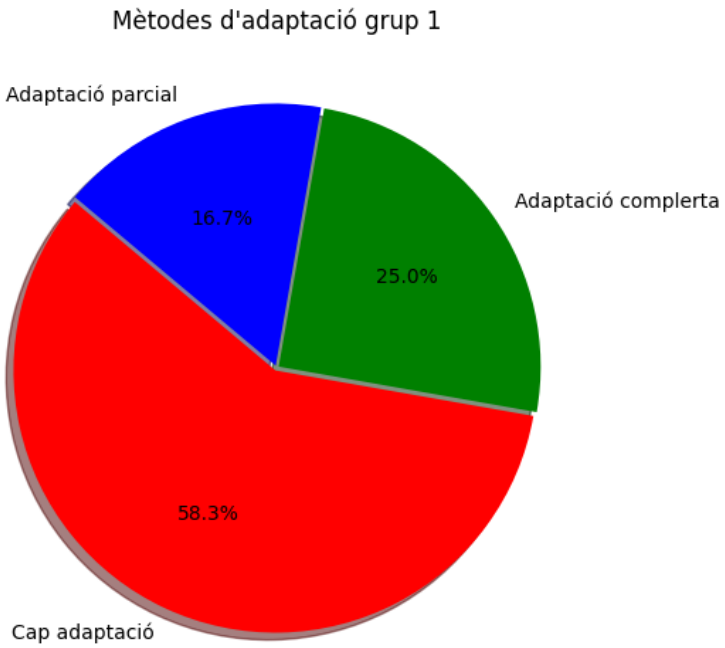

Figura 22: Percentatges d'us de les diferents tècniques de *transfer learning* del grup 1. La majoria no fan servir *transfer learning*, seguit pels que fan servir l'adaptació complerta, i, per últim, els que fan servir l'adaptació parcial.

Capítol 5

Conclusions

Després d'analitzar les dades sobre com els subjectes aprenen i comparar-les amb el comportament d'un agent d'aprenentatge per reforç, hem pogut veure com els subjectes, sense ser-ne conscients trobaven estratègies tal i com ho faria un algorisme com el Q-learning.

D'aquest treball també hem pogut concloure com d'importants són les experiències passades i els coneixements previs quan aprenem i en com això ens afecta encara que les dues situacions no tinguin relació aparent.

5.1 Possibles ampliacions

Aquest tema té moltes branques que es podrien prendre per ampliar i aconseguir més informació interessant sobre la relació entre l'aprenentatge per reforç i l'apre-

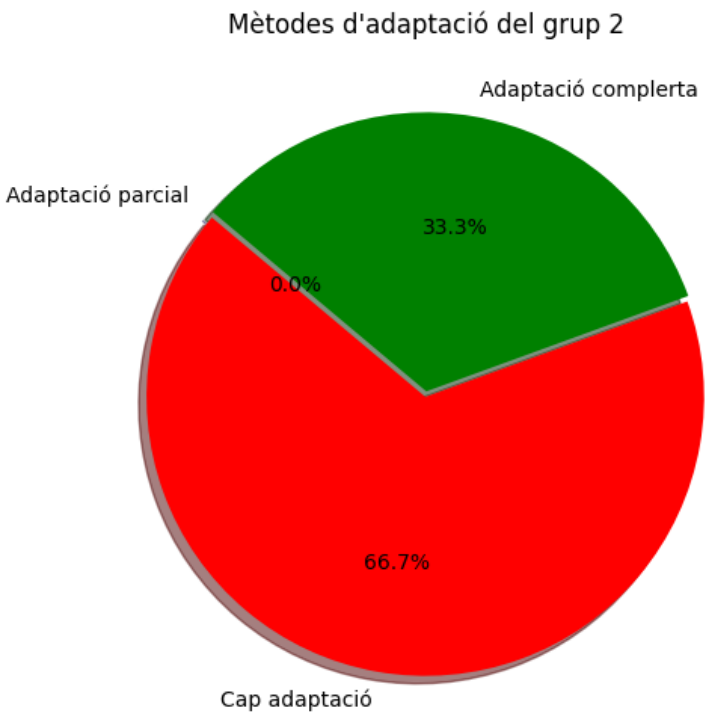

Figura 23: Percentatges d'ús de les diferents tècniques de *transfer learning* del grup 2. Dos dels subjectes en el grup no fan servir cap mena d'adaptació de les dades i un fa servir l'adaptació complerta.

nentatge animal. Per exemple, es podria provar de fer un descriptor que no només tingui en compte el rendiment general del subjecte, sinó que estigui descrit com es comporta a un nivell més baix. També es podrien provar de fer servir altres algorismes d'aprenentatge per reforç com podria ser el *Deep Q-Networks*, el qual és una ampliació del Q-learning que fa servir xarxes neuronals.

També considero que en aquest treball no s'ha vist un efecte significatiu de l'aplicació de *transfer learning*, però que es podrà intentar utilitzar el mateix però amb un experiment més complex de resoldre, per exemple fer episodis amb un horitzó superior a 2. El rol crucial del *transfer learning* ressaltaria més, ja que, amb aquest exercici, el Q-learning aconseguia igualar al comportament del subjecte simplement tenint una taxa d'aprenentatge més alta.

Referències

[1] Richard S. Sutton; Andrew G. Barto: Reinforcement Learning: An Introduction (2014, 2015)
https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

[2] Medium (2023) Colin Horgan: Building a Tree-structured Parzen Estimator from scratch (Kind Of).
https://towardsdatascience.com/building-a-tree-structured-parzen-estimator-from

[3] builtin (2022) Nikolas Donges: What Is *transfer learning*? Exploring the Popular Deep Learning Approach.
https://builtin.com/data-science/transfer-learning

[4] Medium (2020) Dhanoop Karunakaran: Q-learning: a value-based reinforcement learning algorithm.
https://builtin.com/data-science/transfer-learning

[5] dendroid (2011) K-means clústering algorithm explained
https://dendroid.sk/2011/05/09/k-means-clústering/