Pràctica 2: Anàlisi d'Algoritmes d'Aprenentatge per Reforç

SID - Grup 03

Enric Segarra

Marc Font

Pablo Calomardo

10 de maig de 2025

Índex

1 Introducció

En aquesta pràctica s'ha realitzat l'anàlisi i comparació de diferents algoritmes d'aprenentatge per reforç en l'entorn CliffWalking-v1. S'han implementat i analitzat els algoritmes d'Iteració de Valor, Estimació Directa, Q-Learning i REINFORCE, amb l'objectiu d'estudiar els avantatges i inconvenients de cadascun, així com l'efecte dels diferents paràmetres en el rendiment de l'agent entrenat.

Aquest document presenta els resultats obtinguts i l'anàlisi comparativa dels algoritmes, contextualitzant cada resultat amb les propietats teòriques que s'han vist a les sessions corresponents.

2 Descripció de l'entorn CliffWalking

L'entorn CliffWalking-v1 és un problema de navegació en una graella on l'agent ha d'aprendre a moure's des d'un punt inicial fins a un objectiu evitant caure per un penya-segat.

2.1 Característiques de l'entorn

- **Espai d'observacions**: Graella de 4x12, on cada cel·la representa un estat (48 estats possibles).
- **Espai d'accions**: 4 accions possibles (amunt, avall, esquerra, dreta).
- **Dinàmica**: Versió lliscant (is_slippery=True), on les accions tenen una probabilitat de no tenir l'efecte desitjat.
- **Recompensa**: -1 per cada pas, -100 per caure pel penya-segat.
- **Estat inicial**: Cantonada inferior esquerra.
- **Estat objectiu**: Cantonada inferior dreta.

2.2 Reptes de l'entorn

- L'estocacitat introduïda pel mode lliscant dificulta l'aprenentatge.
- El penya-segat representa un risc alt que l'agent ha d'aprendre a evitar.
- Cal trobar un equilibri entre la ruta més curta i la més segura.

3 Hipòtesis i paràmetres d'experimentació

3.1 Hipòtesis

Abans de realitzar els experiments, hem plantejat les següents hipòtesis:

1. Els algoritmes basats en model (Iteració de Valor i Estimació Directa) convergiran més ràpidament que els basats en experiència (Q-Learning i REINFORCE).
2. El factor de descompte tindrà un impacte significatiu en el comportament de risc de l'agent.
3. Q-Learning mostrarà una major sensibilitat als hiperparàmetres com la taxa d'aprenentatge i el coeficient d'exploració.
4. REINFORCE requerirà un major nombre d'episodis per convergir, però potencialment pot trobar polítiques més òptimes en entorns estocàstics.

3.2 Paràmetres d'experimentació

Per a tots els algoritmes, s'han avaluat els paràmetres comuns que afecten el seu comportament:

- **Factor de descompte** (γ): [0.9, 0.95, 0.99, 0.999]
- **Funcions de recompensa personalitzades**:
  - **Recompensa per defecte**: Manté les recompenses originals de l'entorn
  - **Penalització per pas**: Afegeix una petita penalització (-0.01) per cada pas per fomentar camins més curts
  - **Evitació del penya-segat**: Augmenta la penalització per caure pel penya-segat (-200 en lloc de -100)

A més, per a cada algoritme s'han analitzat paràmetres específics:

- **Iteració de Valor**: Paràmetre de convergència (theta): [1e-3, 1e-4, 1e-6]
- **Q-Learning**: Taxa d'aprenentatge, coeficient d'exploració i els seus respectius factors de decaïment
- **REINFORCE**: Taxa d'aprenentatge i mida del batch

3.3 Mètriques d'avaluació

Per avaluar el rendiment dels diferents algoritmes, hem emprat les següents mètriques:

- **Temps d'entrenament i nombre d'iteracions**
- **Recompensa mitjana**
- **Nombre de passos per completar l'episodi**
- **Taxa d'èxit**
- **Optimalitat de la política resultant**

4 Iteració de Valor

L'Iteració de Valor és un mètode d'aprenentatge per reforç basat en model que determina la política òptima a través de la convergència de la funció de valor. A diferència dels mètodes basats en experiència, aquest algoritme requereix accés complet al model de l'entorn, incloent les probabilitats de transició i les recompenses.

4.1 Fonaments teòrics

L'algoritme d'Iteració de Valor es basa en el principi d'optimalitat de Bellman, que estableix que la política òptima per a un problema de decisió markovià (MDP) ha de satisfer la següent equació:

$$V^*(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^*(s')] \tag{1}$$

on $V^*(s)$ és el valor òptim de l'estat s, $P(s'|s,a)$ és la probabilitat de transició de l'estat s a l'estat s' prenent l'acció a, $R(s,a,s')$ és la recompensa rebuda, i γ és el factor de descompte.

L'algoritme actualitza iterativament la funció de valor fins que convergeix a la funció de valor òptima. Un cop obtinguda, es pot derivar la política òptima escollint, per a cada estat, l'acció que maximitza el valor esperat.

4.2 Resultats experimentals

Els experiments realitzats amb diferents valors de gamma (0.9, 0.95, 0.99, 0.999) i theta (1e-3, 1e-4, 1e-6) han permès analitzar el comportament de l'algoritme en l'entorn CliffWalking.

Política òptima i funció de valor

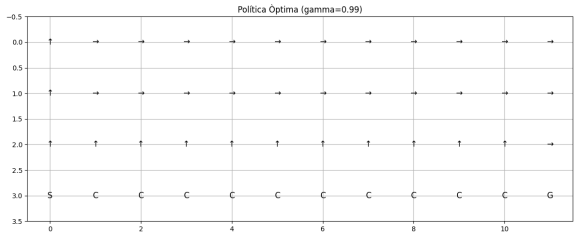

Figura 1: Política òptima (gamma=0.99). S: estat inicial, G: objectiu, C: penya-segat. Les fletxes indiquen l'acció a prendre en cada estat.

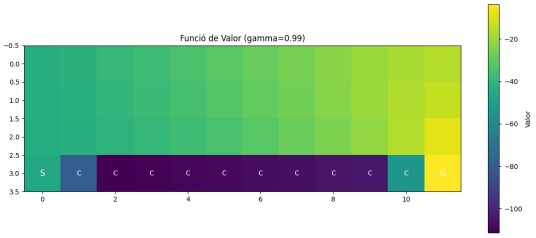

Figura 2: Funció de valor (gamma=0.99). Colors més clars indiquen valors més alts.

La política òptima mostrada a la Figura 1 presenta un comportament conservador, optant per un camí que evita completament el penya-segat. L'agent prefereix el camí més llarg però segur, pujant fins a la fila superior i després movent-se cap a la dreta fins arribar a la columna de l'objectiu, per finalment baixar.

La funció de valor de la Figura 2 mostra clarament com els estats propers al penya-segat (marcats amb C) tenen valors molt baixos (colors foscos), mentre que els estats propers a l'objectiu (G) tenen valors més alts (colors clars). Es pot observar el gradient de valor que guia l'agent cap a l'objectiu per un camí segur.

Efecte del factor de descompte i funcions de recompensa

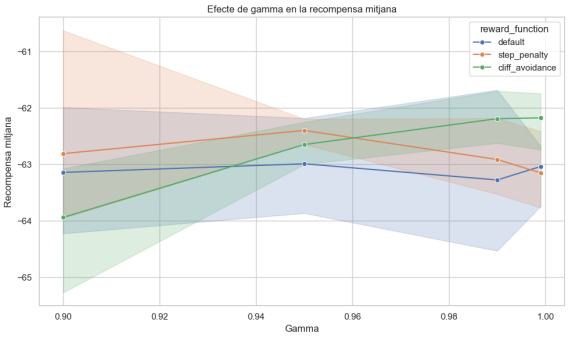

Figura 3: Efecte de gamma en la recompensa mitjana per diferents funcions de recompensa.

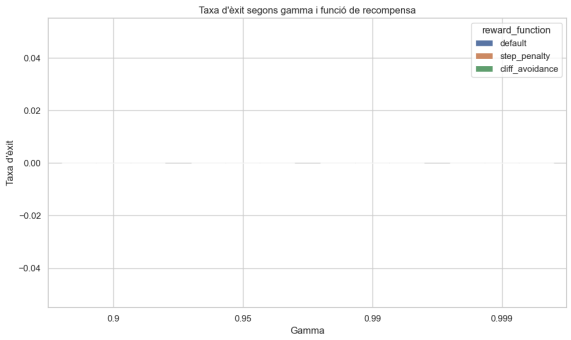

Figura 4: Taxa d'èxit segons gamma i funció de recompensa.

Com s'observa a la Figura 3, el valor de gamma afecta significativament la recompensa mitjana obtinguda. Per la funció de recompensa per defecte (blau), hi ha una lleugera millora a gamma=0.95 seguida d'una disminució a valors més alts. La funció d'evitació del penya-segat (verd) mostra una millora constant a mesura que gamma augmenta, indicant que valors més alts de gamma afavoreixen comportaments més conservadors.

La Figura 4 mostra que les taxes d'èxit són extremadament baixes per a totes les combinacions de paràmetres, suggerint que en l'entorn estocàstic (is_slippery=True), arribar a l'objectiu resulta molt difícil malgrat tenir una política òptima.

Efecte dels paràmetres de convergència i rendiment computacional

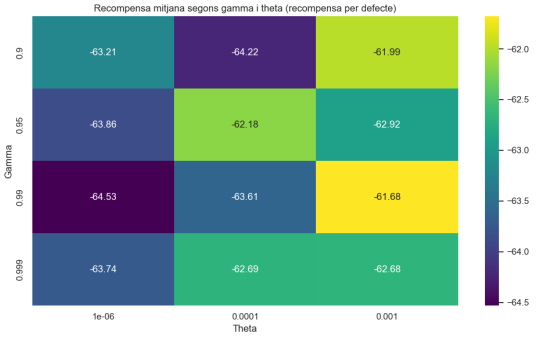

Figura 5: Heatmap de recompensa mitjana segons gamma i theta.

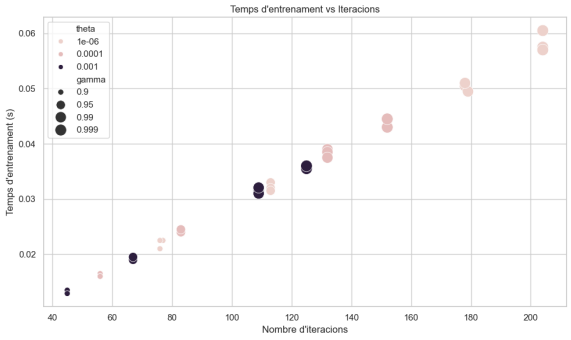

Figura 6: Relació entre temps d'entrenament i iteracions.

El heatmap de la Figura 5 mostra la influència conjunta de gamma i theta en la recompensa mitjana. La millor combinació sembla ser gamma=0.99 amb theta=0.001, que dóna

la recompensa més alta (-61.68). És interessant observar que valors de theta més petits (criteri de convergència més estricte) no sempre donen millors resultats.

A la Figura 6 s'observa una clara relació lineal entre el nombre d'iteracions i el temps d'entrenament. També es pot apreciar que valors més alts de gamma i valors més petits de theta requereixen més iteracions per convergir.

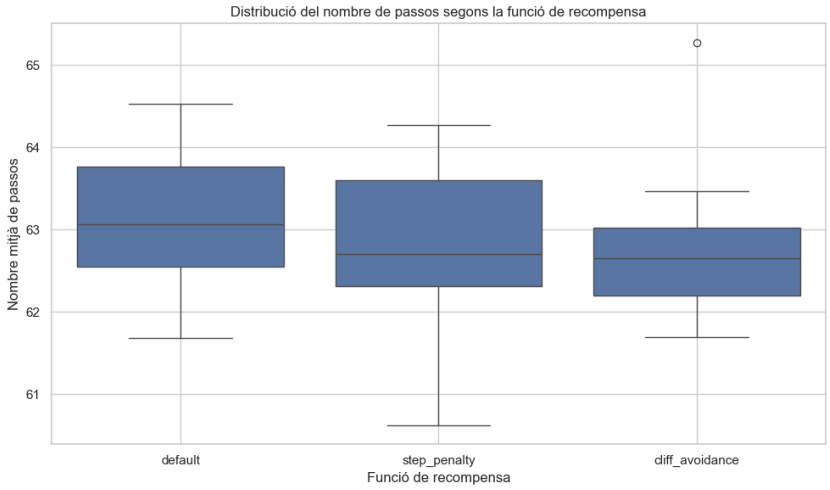

Figura 7: Distribució del nombre mitjà de passos per episodi segons la funció de recompensa.

El boxplot de la Figura 7 mostra com les diferents funcions de recompensa afecten el nombre de passos que l'agent necessita per completar un episodi. La funció d'evitació del penya-segat (cliff_avoidance) tendeix a produir trajectòries més curtes, mentre que la recompensa per defecte mostra una major variabilitat.

4.3 Anàlisi de l'efecte dels paràmetres

Factor de descompte (gamma)

El factor de descompte ha demostrat tenir un impacte significatiu en el comportament de l'agent:

- **Valors baixos (0.9)**: L'agent tendeix a prioritzar les recompenses immediates, el que pot portar a comportaments més arriscats i menys òptims a llarg termini.
- **Valors moderats (0.95-0.99)**: Ofereixen un bon equilibri entre la consideració de recompenses immediates i futures, produint polítiques més estables i millors recompenses mitjanes.
- **Valors molt alts (0.999)**: Poden causar una convergència més lenta i, en alguns casos, una lleugera degradació del rendiment a causa de l'excés de ponderació de recompenses futures incertes.

Paràmetre de convergència (theta)

El valor de theta influeix en la precisió i el temps d'entrenament:

- **Valors més grans (1e-3)**: Permeten una convergència més ràpida amb menys iteracions, però poden resultar en polítiques subòptimes.

- **Valors més petits (1e-6)**: Proporcionen més precisió en la funció de valor resultant, però requereixen més iteracions i temps de càlcul.

Els resultats mostren que no sempre és necessari utilitzar valors molt petits de theta; en molts casos, un valor de 1e-4 ofereix un bon equilibri entre precisió i eficiència computacional.

Funcions de recompensa personalitzades

Les diferents funcions de recompensa han demostrat influir en el comportament de l'agent, com es pot observar a les Figures 3 i 7:

- **Recompensa per defecte**: Produeix un comportament equilibrat però amb resultats variables.
- **Penalització per pas**: Incentiva l'agent a trobar camins més curts, però pot portar a comportaments més arriscats.
- **Evitació del penya-segat**: Amb una penalització més severa per caure, produeix comportaments més conservadors que prioritzen la seguretat sobre l'eficiència.

4.4 Fortaleses i debilitats

Fortaleses

1. **Convergència garantida**: L'algoritme d'Iteració de Valor té garantia matemàtica de convergència a la política òptima per a qualsevol MDP amb recompenses limitades i factor de descompte menor que 1.
2. **Precisió**: En obtenir la convergència, la política resultant és òptima respecte al model del què disposa l'agent.
3. **Eficiència computacional**: En entorns amb espais d'estats i accions discrets com CliffWalking, l'algoritme convergeix en un nombre raonable d'iteracions, com es pot observar a la Figura 6.
4. **Adaptabilitat a diferents criteris d'optimització**: Mitjançant l'ajust del factor de descompte i les funcions de recompensa, es poden obtenir polítiques que optimitzin diferents aspectes com seguretat, eficiència o equilibri entre ambdues.

Debilitats

1. **Requereix model complet**: L'algoritme necessita conèixer les probabilitats de transició i les recompenses de l'entorn, el que no sempre és possible en aplicacions del món real.
2. **Maledicció de la dimensionalitat**: Per a espais d'estats o accions grans, el cost computacional creix exponencialment, limitant la seva aplicabilitat.
3. **Sensibilitat a l'estocacitat**: En l'entorn lliscant (is_slippery=True), les taxes d'èxit són molt baixes, com s'observa a la Figura 4, indicant que la política òptima teòrica pot no ser robusta davant de l'aleatorietat de l'entorn.
4. **Dificultats amb horitzons llargs**: Amb valors de gamma propers a 1, l'algoritme necessita més iteracions per convergir i pot ser menys estable.

En el context de CliffWalking, l'Iteració de Valor ha demostrat ser capaç de trobar polítiques que equilibren el risc i l'eficiència, com es pot veure a la Figura 1, tot i que la seva efectivitat es veu limitada per la naturalesa estocàstica de l'entorn.

5 Estimació Directa

L'Estimació Directa és un mètode d'aprenentatge per reforç basat en model que, a diferència de la Iteració de Valor, combina l'aprenentatge del model a partir de l'experiència amb la planificació.

5.1 Fonaments teòrics

L'algoritme d'Estimació Directa es diferencia d'altres mètodes basats en model perquè construeix i actualitza el seu propi model de l'entorn mentre interactua amb ell. Aquest procés té tres components principals:

1. **Aprenentatge del model**: L'agent recull experiències en forma de transicions (s, a, r, s') i les utilitza per construir un model de l'entorn, estimant les probabilitats de transició $P(s'|s, a)$ i les recompenses esperades $R(s, a, s')$.
2. **Planificació**: Utilitzant el model après, l'agent realitza iteracions de planificació per millorar la seva política, aplicant tècniques similars a la Iteració de Valor però només sobre els estats ja visitats.
3. **Exploració i explotació**: L'agent ha de balancejar l'exploració de l'entorn per millorar el seu model amb l'explotació del coneixement actual per maximitzar la recompensa.

L'actualització dels valors segueix una versió adaptada de l'equació de Bellman:

$$Q(s, a) = \sum_{s'} \hat{P}(s'|s, a)[\hat{R}(s, a, s') + \gamma \max_{a'} Q(s', a')] \tag{2}$$

on $\hat{P}(s'|s, a)$ i $\hat{R}(s, a, s')$ són les estimacions de les probabilitats de transició i recompenses basades en les experiències recollides fins al moment.

5.2 Resultats experimentals

Els experiments amb l'algoritme d'Estimació Directa s'han centrat en analitzar l'efecte de diversos paràmetres sobre el rendiment, incloent el factor de descompte (gamma), el nombre de passos de planificació, el epsylon decay i diferents funcions de recompensa.

Política òptima i funció de valor

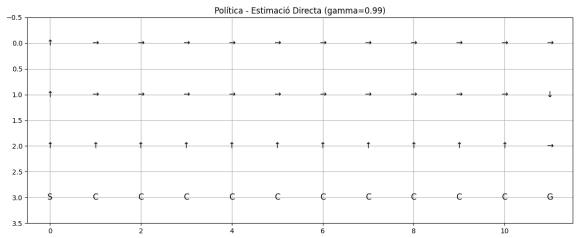

Figura 8: Política òptima obtinguda amb Estimació Directa (gamma=0.99). S: estat inicial, G: objectiu, C: penya-segat.

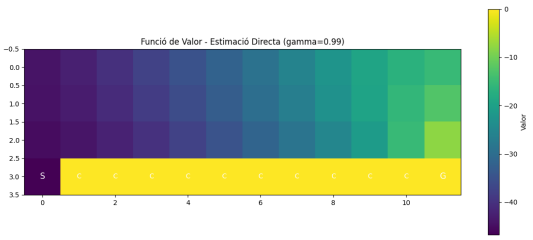

Figura 9: Funció de valor amb Estimació Directa (gamma=0.99). Colors més clars indiquen valors més alts.

La política òptima obtinguda amb Estimació Directa (Figura 8) presenta una estratègia similar a la de la Iteració de Valor, evitant el penya-segat mitjançant un camí segur per la part superior de la graella. Això reflecteix que, malgrat la diferència en l'aproximació, ambdós mètodes basats en model poden convergir a solucions similars.

La funció de valor (Figura 9) mostra un gradient clar des de l'estat inicial fins a l'objectiu. És interessant observar com el penya-segat té un valor molt més alt que a la funció de valor de la Iteració de Valor. Això es deu a que l'Estimació Directa no té accés complet al model des del principi, sinó que l'aprèn a través de l'experiència.

Progrés d'entrenament

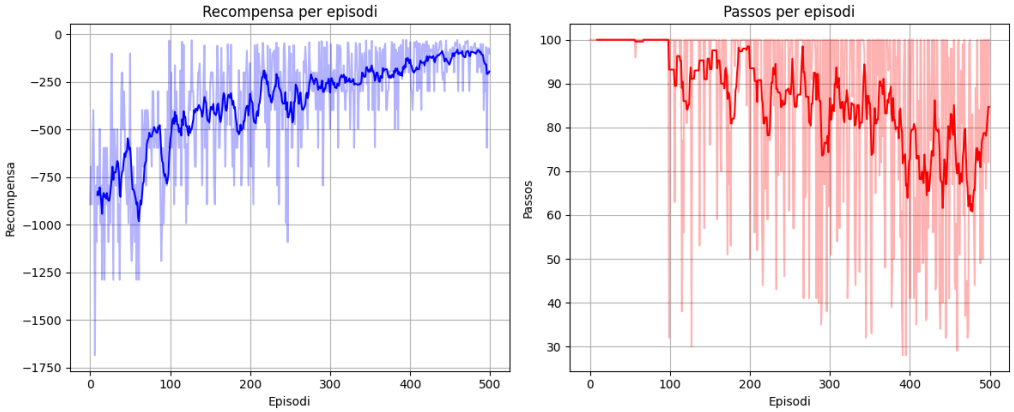

Figura 10: Progrés d'entrenament: recompensa i passos per episodi al llarg de l'entrenament.

La Figura 10 mostra l'evolució de la recompensa i el nombre de passos per episodi durant l'entrenament. S'observa una clara millora en la recompensa al llarg dels episodis, passant de valors molt negatius (al voltant de -1000) a valors propers a -100, indicant que l'agent aprèn progressivament a evitar el penya-segat i a trobar camins més eficients cap a l'objectiu.

Paral·lelament, el nombre de passos per episodi tendeix a disminuir a mesura que avança l'entrenament, mostrant que l'agent aprèn a completar l'episodi de manera més eficient.

Efecte del factor de descompte i del nombre de passos de planificació

La Figura 11 mostra com el factor de descompte (gamma) afecta la recompensa obtinguda. S'observa un increment en la recompensa a mesura que gamma augmenta fins a 0.99, seguit d'una caiguda significativa per a gamma=0.999. Aquest comportament suggereix que valors de gamma molt propers a 1 poden portar a inestabilitats en l'aprenentatge.

Pel que fa al nombre de passos de planificació (Figura 12), s'observa un comportament no monòton, amb un màxim de rendiment al voltant de 3 passos de planificació. Això indica que més planificació no sempre és millor; un excés de planificació pot portar a sobreajustament sobre un model encara imperfecte, especialment en les primeres etapes de l'entrenament.

El epsylon decay (Figura 13) també mostra un efecte no trivial, amb un mínim de rendiment per a valors intermedis (0.995) i millors resultats tant per a decaïments més ràpids

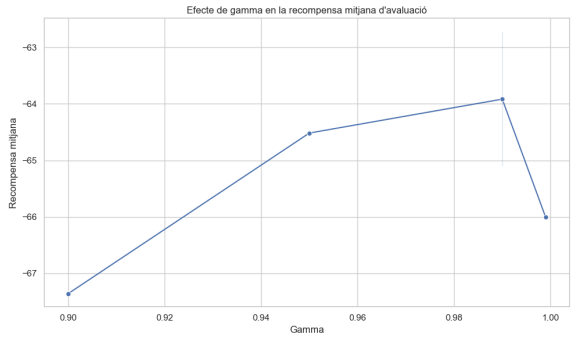

Figura 11: Efecte de gamma en la recompensa mitjana d'avaluació.

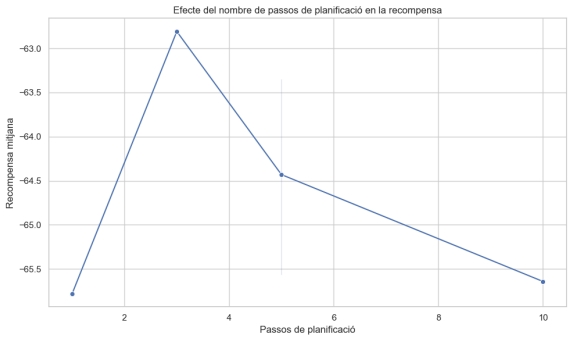

Figura 12: Efecte del nombre de passos de planificació en la recompensa.

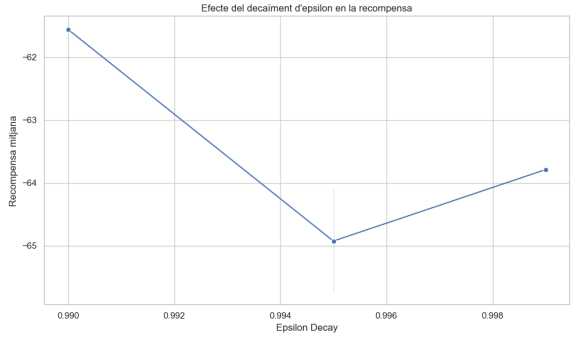

Figura 13: Efecte del epsylon decay en la recompensa mitjana.

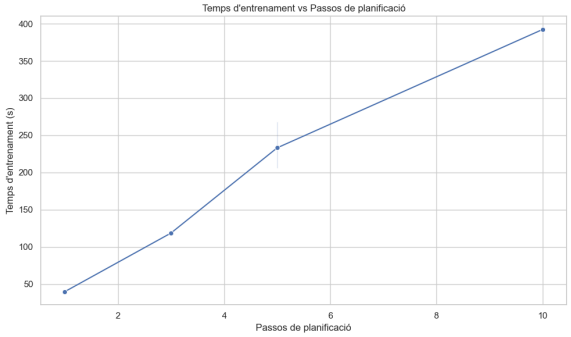

Figura 14: Temps d'entrenament vs. passos de planificació.

(0.99) com més lents (0.999). Això suggereix un compromís complex entre exploració i explotació.

La Figura 14 mostra una relació clarament lineal entre el nombre de passos de planificació i el temps d'entrenament. Aquest resultat era esperable, ja que cada pas de planificació addicional implica més càlculs per episodi.

Efecte de les funcions de recompensa

La comparació entre diferents funcions de recompensa (Figura 15) mostra diferències relativament petites en la recompensa mitjana obtinguda. No obstant això, quan s'analitza el nombre de passos per episodi (Figura 16), s'observen diferències més notables, amb la funció d'evitació del penya-segat (cliff_avoidance) produint trajectòries més curtes.

La Figura 17 mostra la taxa d'èxit segons diferents combinacions de paràmetres. Sorprenentment, totes les configuracions mostren taxes d'èxit molt properes a zero, indicant la dificultat d'arribar a l'objectiu en l'entorn estocàstic (is_slippery=True), independentment dels paràmetres utilitzats.

5.3 Anàlisi de l'efecte dels paràmetres

Factor de descompte (gamma)

El factor de descompte té un impacte significant en el rendiment de l'algoritme:

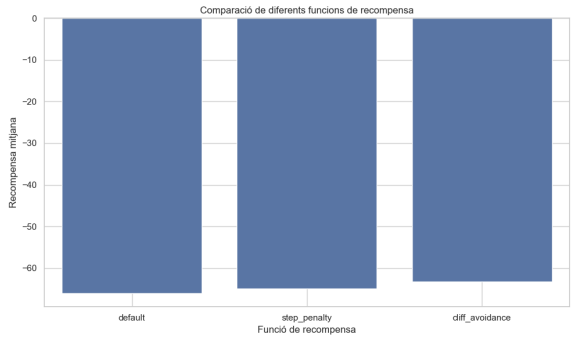

Figura 15: Comparació de diferents funcions de recompensa.

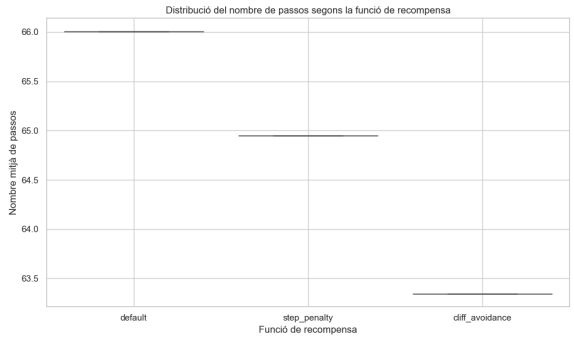

Figura 16: Distribució del nombre de passos segons la funció de recompensa.

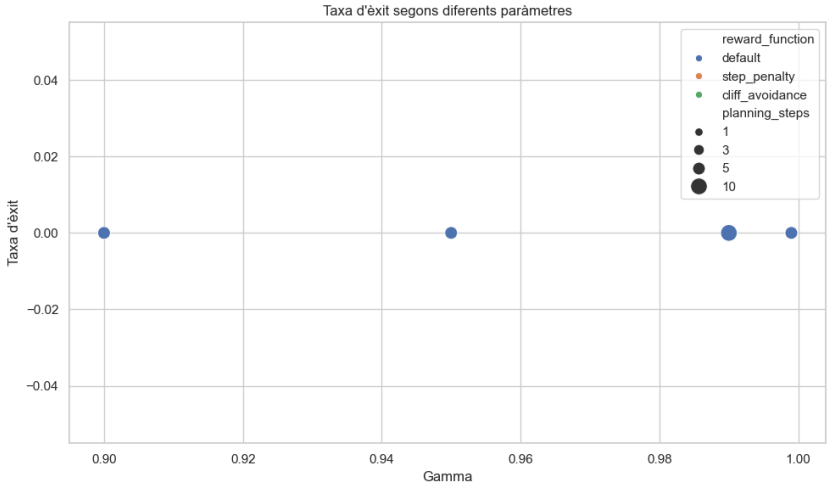

Figura 17: Taxa d'èxit segons diferents paràmetres.

- **Valors baixos (0.9)**: Produeixen polítiques que prioritzen recompenses immediates, resultant en comportaments més arriscats.
- **Valors moderats (0.95-0.99)**: Ofereixen el millor rendiment, com es veu a la Figura 11, balancejant correctament recompenses immediates i futures.
- **Valors molt alts (0.999)**: Causen una degradació del rendiment, probablement perquè donen massa importància a recompenses futures incertes, especialment en un entorn estocàstic.

Nombre de passos de planificació

El nombre de passos de planificació mostra un compromís interessant entre millora de la política i sobreajustament:

- **Pocs passos (1-2)**: Resulten en un aprenentatge subòptim per manca de refinament de la política.
- **Nombre moderat (3)**: Proporciona el millor rendiment (Figura 12), permetent suficient millora sense caure en sobreajustament.
- **Molts passos (5-10)**: Mostren una degradació del rendiment, possiblement per

sobreajustament a un model imperfecte, a més d'augmentar significativament el temps de computació (Figura 14).

Epsylon Decay

El paràmetre de epsylon decay afecta l'equilibri entre exploració i explotació:

- **Decaïment ràpid (0.99)**: Permet més exploració inicial seguida d'una ràpida transició a l'explotació, resultant en un bon rendiment (Figura 13).
- **Decaïment moderat (0.995)**: Mostra el pitjor rendiment, suggerint que aquesta taxa de decaïment no sincronitza bé amb el ritme d'aprenentatge de l'agent.
- **Decaïment lent (0.999)**: Manté una exploració més prolongada, que en aquest cas sembla beneficiar l'agent.

Funcions de recompensa personalitzades

Les funcions de recompensa mostren efectes més subtils:

- **Recompensa per defecte**: Serveix com a baseline, amb un rendiment intermedi.
- **Penalització per pas**: No mostra avantatges clars respecte a la recompensa per defecte.
- **Evitació del penya-segat**: Tendeix a produir trajectòries més curtes (Figura 16), suggerint que una penalització més severa per caure ajuda a trobar camins més eficients.

5.4 Fortaleses i debilitats

Fortaleses

1. **Aprenentatge del model**: A diferència de la Iteració de Valor, l'Estimació Directa no requereix conèixer el model de l'entorn a priori, sinó que l'aprèn a partir de l'experiència.
2. **Planificació eficient**: Realitza planificació només sobre els estats visitats, optimitzant l'ús de recursos computacionals.
3. **Adaptabilitat**: Pot adaptar-se a canvis en l'entorn, ja que continua actualitzant el seu model a mesura que rep noves experiències.
4. **Equilibri exploració-explotació**: El mecanisme d'epsilon-greedy permet balancejar l'exploració de l'entorn amb l'explotació del coneixement actual.

Debilitats

1. **Sensibilitat als paràmetres**: El rendiment depèn significativament dels paràmetres escollits, especialment el nombre de passos de planificació i el epsylon decay, com mostren les Figures 12 i 13.
2. **Temps de computació**: El cost computacional augmenta linealment amb el nombre de passos de planificació (Figura 14), cosa que pot limitar la seva aplicabilitat en entorns complexos.

3. **Sensibilitat a l'estocacitat**: Com en el cas de la Iteració de Valor, les taxes d'èxit són extremadament baixes en l'entorn estocàstic (Figura 17), indicant limitacions fonamentals davant d'alta aleatorietat.
4. **Limitacions de generalització**: L'algoritme només pot planificar sobre estats ja visitats, limitant la seva capacitat de generalització a parts inexplorades de l'entorn.

L'Estimació Directa es posiciona com un mètode intermedi entre els purament basats en model (com la Iteració de Valor) i els basats exclusivament en experiència (com Q-Learning). Aquesta característica dual li permet mantenir els avantatges de la planificació mentre redueix la necessitat de tenir un model complet de l'entorn a priori.

6 Q-Learning

Q-Learning és un algorisme d'aprenentatge per reforç sense model (model-free) que aprèn directament la funció Q òptima a partir de l'experiència, sense necessitar un model explícit de l'entorn.

6.1 Fonaments teòrics

Q-Learning es basa en el mètode de diferència temporal (TD) per aproximar la funció acció-valor òptima, $Q^*(s, a)$, que representa el valor esperat de realitzar l'acció a a l'estat s i seguir la política òptima a partir d'aquell moment.

L'actualització de la funció Q segueix la fórmula:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{3}$$

on:

- $Q(s, a)$ és el valor actual de la funció Q per a l'estat s i l'acció a.
- α és la taxa d'aprenentatge (learning rate).
- r és la recompensa immediata rebuda.
- γ és el factor de descompte.
- s' és l'estat següent.
- $\max_{a'} Q(s', a')$ és el valor màxim possible en l'estat següent (segons l'estimació actual).

Q-Learning implementa una estratègia d'exploració-explotació, típicament mitjançant una política ϵ-greedy, on amb probabilitat ϵ es pren una acció aleatòria (exploració), i amb probabilitat $1 - \epsilon$ es pren l'acció amb major valor Q estimat (explotació).

Els paràmetres clau de l'algoritme són:

- **Nombre d'episodis (NUM_EPISODES)**: Quants episodis complets d'entrenament s'executaran.
- **Factor de descompte (γ o GAMMA)**: Determina la importància de les recompenses futures versus les immediates.
- **Taxa d'aprenentatge (α o LEARNING_RATE)**: Controla quant s'actualitzen les estimacions de Q en cada pas.
- **Paràmetre d'exploració (ϵ o EPSILON)**: Probabilitat inicial d'escollir una acció aleatòria.
- **Decaïment d'epsilon (EPSILON_DECAY)**: Factor per reduir gradualment l'exploració.
- **Valor mínim d'epsilon (EPSILON_MIN)**: Valor mínim al qual pot arribar epsilon.
- **Decaïment de la taxa d'aprenentatge (LEARNING_RATE_DECAY)**: Factor per reduir gradualment la taxa d'aprenentatge.

A diferència dels mètodes basats en model com la Iteració de Valor o l'Estimació Directa, Q-Learning no requereix cap coneixement de les probabilitats de transició ni de les recompenses esperades, sinó que aprèn directament a partir de les interaccions amb l'entorn.

6.2 Resultats experimentals

Els experiments amb Q-Learning s'han centrat en l'anàlisi de l'efecte dels diversos paràmetres sobre el rendiment de l'agent en l'entorn CliffWalking. Per a cada paràmetre s'han definit diferents valors possibles i s'han provat totes les combinacions per identificar les configuracions òptimes.

Efecte del factor de descompte, taxa d'aprenentatge i nombre d'episodis

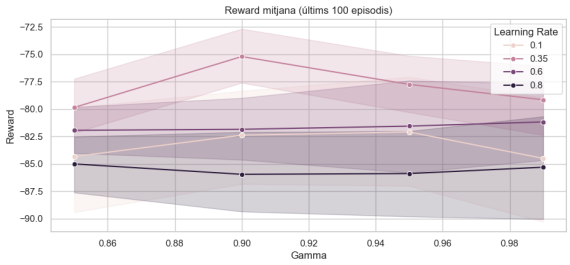

Figura 18: Efecte de gamma i learning rate en la recompensa mitjana dels últims 100 episodis.

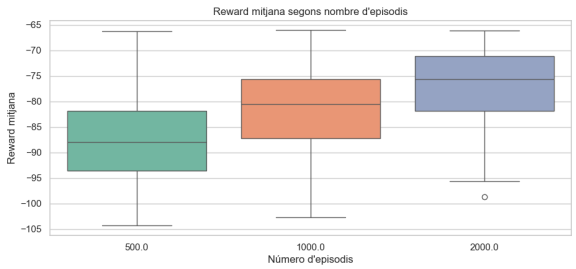

Figura 19: Recompensa mitjana segons el nombre d'episodis d'entrenament.

La Figura 18 mostra la recompensa mitjana dels últims 100 episodis en funció del factor de descompte (gamma), amb diferents corbes per a diferents valors de taxa d'aprenentatge. S'observa que la taxa d'aprenentatge de 0.35 (corba vermella) proporciona consistentment millors resultats que altres valors, i per a aquesta taxa d'aprenentatge, el millor valor de gamma es troba al voltant de 0.9, amb una recompensa mitjana propera a -75.

La Figura 19 presenta un boxplot de la recompensa mitjana obtinguda amb diferents nombres d'episodis d'entrenament (500, 1000 i 2000). Es pot observar una clara tendència de millora a mesura que augmenta el nombre d'episodis, amb les recompenses més altes (menys negatives) obtingudes amb 2000 episodis. La mediana de recompensa millora d'aproximadament -88 amb 500 episodis a -80 amb 1000 episodis, i a -76 amb 2000 episodis.

Efecte del decaïment d'epsilon i interacció entre paràmetres

La Figura 20 mostra l'evolució de la recompensa mitjana al llarg dels episodis per a diferents valors de decaïment d'epsilon. Els valors de decaïment més ràpids (0.8 i 0.9, línies blava i taronja) permeten una convergència molt més ràpida, estabilitzant-se al voltant de l'episodi 50, mentre que el decaïment més lent (0.99, línia verda) millora molt més gradualment, arribant a valors similars només cap a l'episodi 400.

La Figura 21 presenta un heatmap de la recompensa mitjana obtinguda amb diferents combinacions de gamma i learning rate. La millor combinació és gamma=0.9 amb learning rate=0.35, que assoleix una recompensa mitjana de -75.22. En general, les taxes d'aprenentatge moderades (0.35) funcionen millor que les molt baixes (0.1) o molt altes

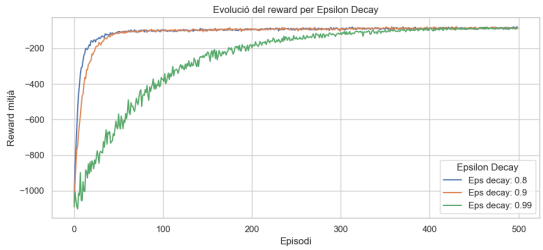

Figura 20: Evolució de la recompensa per episodi amb diferents valors d'epsilon decay.

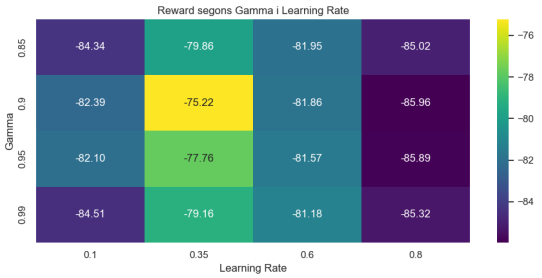

Figura 21: Heatmap de recompensa mitjana segons gamma i learning rate.

(0.8), i els valors de gamma propers a 0.9 tendeixen a produir millors resultats que els valors més extrems.

6.3 Anàlisi de l'efecte dels paràmetres

Taxa d'aprenentatge (α) i el seu decaïment

Els experiments realitzats mostren que la taxa d'aprenentatge (α) juga un paper essencial en la velocitat i estabilitat de la convergència. Concretament, es pot observar que valors moderats com $\alpha = 0.35$ ofereixen millors resultats de reward mitjana, especialment quan es combinen amb valors òptims de Gamma. Una taxa d'aprenentatge massa baixa (per exemple, $\alpha = 0.1$) condueix a una actualització molt lenta dels valors Q, la qual cosa pot alentir l'aprenentatge. D'altra banda, valors massa alts (com $\alpha = 0.8$) poden provocar oscil·lacions o inestabilitat en l'aprenentatge, fet que es reflecteix en una menor reward mitjana.

Tot i que l'experiment no inclou explícitament el decaïment de la taxa d'aprenentatge, es pot inferir que un valor fix moderat és suficient en aquest entorn. En contextos més complexos, incorporar un decaïment progressiu de α podria ajudar a estabilitzar l'aprenentatge a mesura que l'agent explora menys.

Coeficient d'exploració (ϵ) i el seu decaïment

L'evolució del reward mitjà segons diferents valors d'Epsilon Decay evidencia la importància d'una estratègia d'exploració adequada. Valors baixos de decaïment (per exemple, ϵ decay $= 0.8$ o 0.9) faciliten una transició ràpida cap a l'explotació, la qual cosa afavoreix una millora ràpida i estable de la recompensa. Per contra, un decaïment lent (ϵ decay $= 0.99$) manté l'agent en fase exploratòria durant més temps, fet que es tradueix en un aprenentatge més lent i menys eficient, com es veu a la Figura 20.

Aquesta observació és especialment rellevant: si bé l'exploració és clau en les primeres fases, un decaïment adequat d'ϵ és fonamental per evitar que l'agent continuï actuant de forma aleatòria quan ja disposa de prou coneixement. Així doncs, un ϵ decay $= 0.9$ sembla oferir el millor equilibri entre exploració inicial i explotació posterior.

Factor de descompte (γ)

El factor de descompte (γ) determina la importància relativa de les recompenses futures en comparació amb les immediates. Els resultats mostren que $\gamma = 0.90$ proporciona la millor reward mitjana general, especialment quan es combina amb $\alpha = 0.35$, tal com es visualitza a la matriu de calor de la Figura 21.

Valors més baixos com $\gamma = 0.85$ poden fer que l'agent prioritzi recompenses immediates, reduint així l'eficiència de la política a llarg termini. En canvi, valors molt alts com $\gamma = 0.99$ poden fer que l'agent doni massa pes a les recompenses futures, cosa que pot alentir l'aprenentatge inicial i provocar una política menys estable. Per tant, un valor intermedi com $\gamma = 0.90$ sembla oferir el millor compromís entre curt i llarg termini en aquest entorn específic.

6.4 Fortaleses i debilitats

Fortaleses

1. **Aprenentatge sense model:** A diferència de la Iteració de Valor, Q-Learning no requereix conèixer el model de transicions ni de recompenses, fet que el fa aplicable en entorns desconeguts o complexos.
2. **Actualització contínua:** Permet actualitzar els valors Q en cada pas de l'episodi (actualització temporal), a diferència de l'Estimació Directa, que necessita completar l'episodi.
3. **Aprenentatge off-policy:** És capaç d'aprendre la política òptima mentre segueix una altra política d'exploració (com ϵ-greedy), cosa que li dona gran flexibilitat.
4. **Resultats robustos:** Amb una bona selecció d'hiperparàmetres (com $\gamma = 0.9$ i $\alpha = 0.35$), ha mostrat una elevada estabilitat i recompensa mitjana, superant l'Estimació Directa en eficiència i adaptabilitat.

Debilitats

1. **Sensibilitat als hiperparàmetres:** El rendiment es veu fortament afectat per la selecció de valors de γ, α i ϵ decay, com es mostra a les Figures 18 i 21.
2. **Dependència de l'exploració:** Si el decaïment d'ϵ és massa ràpid, es pot reduir prematurament la fase d'exploració i portar a solucions subòptimes.
3. **Convergència lenta:** En entorns grans, la convergència pot ser lenta si no s'utilitzen tècniques d'aproximació de funcions o millores com DQN.
4. **No garanteix la política òptima en la pràctica:** Tot i que teòricament pot convergir a l'òptim, errors acumulats en les actualitzacions poden derivar en polítiques no òptimes si la cobertura de l'estat no és suficient.

Q-Learning es posiciona com un mètode potent basat en experiència directa, més eficient que l'Estimació Directa i més aplicable que la Iteració de Valor en entorns desconeguts. Tot i això, la seva eficàcia depèn d'una exploració ben regulada i d'una acurada selecció de paràmetres per evitar inestabilitats en l'aprenentatge.

7 REINFORCE

REINFORCE és un algoritme de gradient de política que optimitza directament la política sense necessitat d'aprendre una funció de valor.

7.1 Fonaments teòrics

REINFORCE pertany a la família d'algoritmes de gradient de política (policy gradient), i a diferència dels mètodes basats en valor com Iteració de Valor o Q-Learning, optimitza directament la política parametritzada $\pi_\theta(a|s)$ sense necessitat d'estimar explícitament una funció de valor.

L'objectiu de REINFORCE és maximitzar el retorn esperat $J(\theta)$, que es defineix com:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)] \tag{4}$$

on $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T, a_T, r_T\}$ representa una trajectòria completa seguint la política π_θ, i $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$ és el retorn descomptat.

El teorema del gradient de política estableix que el gradient de $J(\theta)$ pot ser calculat com:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right] \tag{5}$$

on $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ és el retorn descomptat a partir del temps t.

Aquest teorema es tradueix en l'algoritme REINFORCE, on la funció de pèrdua a minimitzar s'expressa com:

$$L(\theta) = -\frac{1}{T}\sum_{t=0}^{T-1} \log \pi_\theta(a_t|s_t) G_t \tag{6}$$

El gradient d'aquesta funció de pèrdua té la mateixa direcció (però signe oposat) que el gradient de la funció objectiu $J(\theta)$.

Resumidament, el funcionament de REINFORCE és el següent:

1. **Inicialització**: Inicialitzem els paràmetres θ de la política $\pi_\theta(a|s)$ (en el nostre cas, els pesos d'un model lineal).
2. **Generació d'episodis**: Per cada episodi:
   - Per cada pas $t = 0, 1, \ldots, T-1$:
     - Observar l'estat s_t
     - Mostrejar una acció $a_t \sim \pi_\theta(\cdot|s_t)$
     - Executar l'acció a_t i observar la recompensa r_t i el següent estat s_{t+1}
   - Emmagatzemar la trajectòria completa $\tau = \{s_0, a_0, r_0, \ldots, s_T, a_T, r_T\}$
3. **Actualització dels paràmetres**: Per cada pas t en la trajectòria:
   - Calcular el retorn descomptat $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$

- Actualitzar els paràmetres θ utilitzant ascens de gradient:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) G_t \tag{7}$$

En la nostra implementació, utilitzem una normalització dels retorns per reduir la variància del gradient:

$$G'_t = \frac{G_t - \mu_{G_t}}{\sigma_{G_t} + \epsilon} \tag{8}$$

on μ_{G_t} i σ_{G_t} són la mitjana i la desviació estàndard dels retorns, respectivament, i ϵ és un petit valor per evitar divisions per zero.

A diferència dels mètodes basats en valor, REINFORCE és un algoritme d'aprenentatge basat en episodis complets (Monte Carlo), ja que requereix completar un episodi sencer abans de poder actualitzar els paràmetres. També és important notar que REINFORCE optimitza directament una política estocàstica, mentre que mètodes com Q-Learning típicament deriven una política determinista a partir dels valors Q.

7.2 Resultats experimentals

L'experimentació sobre l'algoritme REINFORCE s'ha centrat a analitzar l'afectació de certs paràmetres sobre la recompensa mitjana obtinguda, com ara la gamma i la taxa d'aprenentatge. També s'han analitzat altres paràmetres com el nombre de passos o la funció de recompensa.

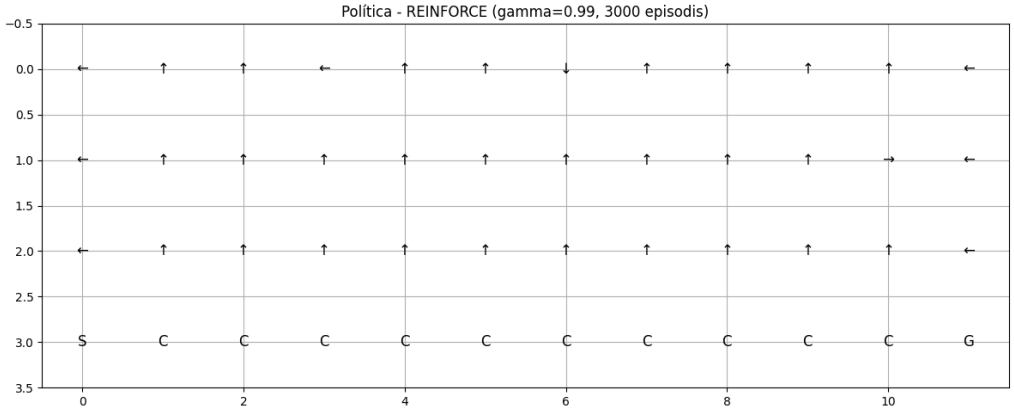

Figura 22: Política òptima obtinguda amb REINFORCE (gamma=0.99, 3000 episodis). S: estat inicial, G: objectiu, C: penya-segat.

La política òptima obtinguda amb REINFORCE (Figura 22) presenta una estratègia similar a la de la Iteració de Valor i Estimació Directa, evitant el penya-segat mitjançant un camí segur per la part superior de la graella. Però a diferència d'aquests dos, la política obtinguda té alguns defectes, ja que podem veure que la segona fila sempre apunta cap a dalt, cosa que és correcta, però les següents fileres mostren direccions amb poc sentit.

Pel que fa a la funció de recompensa, hem decidit experimentar amb tres diferents:

- **default**: És la funció de recompensa per defecte. La política òptima s'ha obtingut amb aquesta funció.

- **step_penalty**: Aquesta funció intenta realitzar el mínim nombre de passos possibles per arribar a l'objectiu.
- **cliff_avoidance**: Aquesta funció intenta evitar el màxim possible caure per un *cliff*.

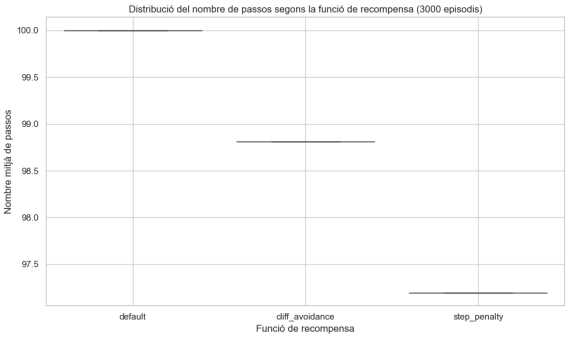

Figura 23: Distribució del nombre de passos segons la funció de recompensa.

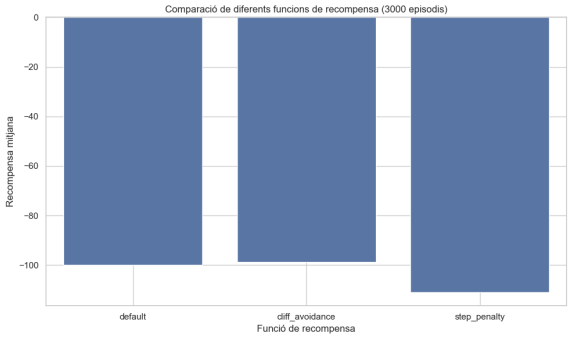

Figura 24: Comparació de diferents funcions de recompensa.

Com podem observar a la Figura 24, la funció que obté els millors resultats és la de cliff_avoidance, mentre que la Figura 23 mostra que la funció step_penalty aconsegueix reduir lleugerament el nombre mitjà de passos necessaris per completar un episodi.

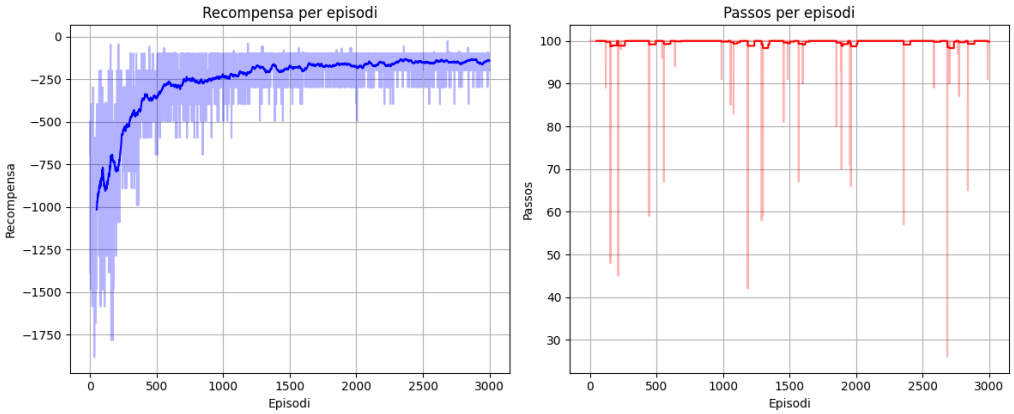

Figura 25: Progrés d'entrenament: recompensa i passos per episodi.

La Figura 25 mostra l'evolució de la recompensa i el nombre de passos per episodi durant l'entrenament (s'ha realitzat amb la funció de recompensa default). Podem veure fàcilment una millora en la recompensa al llarg dels episodis, inclús podem observar un creixement més ràpid que al vist a Estimació Directa. En canvi, veiem que els passos per episodi es mantenen molt estables a tots els episodis.

A la Figura 26 es veu clarament un descens en la recompensa mitjana obtinguda segons s'augmenta el factor de descompte, havent-hi un gran salt del 0.99 al 0.999. Aquest resultat contrasta amb el que havíem observat amb els algoritmes basats en model, on valors més alts de gamma tendien a produir millors resultats.

La Figura 27 mostra que les taxes d'èxit són extremadament baixes per a totes les combinacions de factors de descompte i funcions de recompensa diferents, suggerint que en l'entorn estocàstic (is_slippery=True), arribar a l'objectiu resulta molt difícil malgrat tenir una política òptima, tal com passava a la Iteració per Valor.

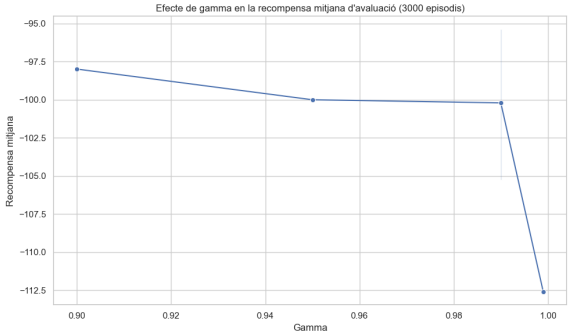

Figura 26: Efecte de gamma en la recompensa mitjana d'avaluació.

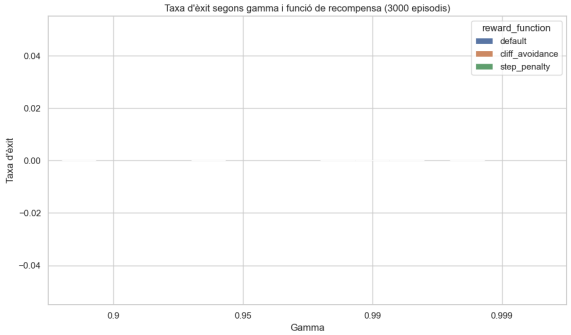

Figura 27: Taxa d'èxit segons gamma i funció de recompensa.

Figura 28: Efecte de la taxa d'aprenentatge en la recompensa.

Figura 29: Temps d'entrenament vs. taxa d'aprenentatge.

A diferència del que s'ha observat amb el factor de descompte, a la Figura 28 observem un clar creixement de la recompensa mitjana obtinguda respecte a com més gran és la taxa d'aprenentatge. Aquest comportament suggereix que valors més alts de taxa d'aprenentatge permeten a l'algoritme REINFORCE ajustar més ràpidament els paràmetres de la política en aquest entorn.

Pel que fa al temps d'entrenament (Figura 29), tenim una variabilitat molt baixa, de tan sols centèsimes de segon, les quals poden ser causades per variabilitats en el procés d'execució fora del nostre control. Tot i això, es pot veure que la taxa d'aprenentatge que més ha trigat és la de valor igual a 0.01 i la que menys és la de valor igual a 0.05.

7.3 Anàlisi de l'efecte dels paràmetres

Els experiments amb l'algoritme REINFORCE ens han permès identificar com els diferents paràmetres afecten el rendiment en l'entorn CliffWalking. A continuació, analitzarem en profunditat aquests efectes:

Factor de descompte (γ)

El factor de descompte en REINFORCE ha mostrat un comportament diferent al dels algoritmes basats en model:

- **Valors baixos (0.9)**: Han proporcionat els millors resultats en termes de recompensa mitjana. Això suggereix que en aquest algoritme de gradient de política,

donar més importància a les recompenses immediates és beneficiós per a l'entorn CliffWalking.

- **Valors moderats (0.95-0.99)**: Mostren un descens gradual en el rendiment, encara que mantenen una recompensa acceptable.
- **Valors molt alts (0.999)**: Presenten una caiguda significativa en el rendiment, com es pot observar a la Figura 26. Això pot atribuir-se a l'alta variància que aquests valors introdueixen en el gradient de la política.

Aquesta tendència és contrària a la que vam observar en els algoritmes basats en valor, on valors més alts de gamma tendien a produir millors resultats. Aquesta diferència pot explicar-se pel fet que REINFORCE, en optimitzar directament la política sense utilitzar una funció de valor intermèdia, és més sensible a la variància introduïda per retorns a llarg termini.

Taxa d'aprenentatge

La taxa d'aprenentatge ha demostrat ser un paràmetre crucial per a l'efectivitat de REINFORCE:

- **Valors baixos (0.001)**: Produeixen un aprenentatge més lent i un rendiment final més pobre, amb una recompensa mitjana al voltant de -105 (Figura 28).
- **Valors intermedis (0.01)**: Milloren el rendiment, però encara no arriben al nivell òptim.
- **Valors alts (0.05)**: Ofereixen el millor rendiment, arribant a una recompensa mitjana d'aproximadament -95. Això contrasta amb molts algoritmes d'aprenentatge on valors alts de taxa d'aprenentatge poden causar inestabilitat.

Aquest comportament suggereix que l'espai de paràmetres de la política en REINFORCE per a aquest entorn permet actualitzacions més agressives sense comprometre l'estabilitat. El fet que no observem un deteriorament del rendiment amb una taxa d'aprenentatge més alta indica que el paisatge d'optimització en aquest cas no presenta mínims locals problemàtics que podrien ser salt per una taxa d'aprenentatge elevada.

Funcions de recompensa personalitzades

Les diferents funcions de recompensa han mostrat efectes interessants:

- **Recompensa per defecte**: Proporciona una base de referència amb un rendiment intermedi.
- **Cliff avoidance**: Aquesta funció, que augmenta la penalització per caure pel precipici, ha mostrat el millor rendiment en termes de recompensa mitjana (Figura 24). Això suggereix que REINFORCE es beneficia especialment de penalitzacions més clares per als comportaments a evitar.
- **Step penalty**: Tot i que aconsegueix reduir el nombre mitjà de passos (Figura 23), també mostra la pitjor recompensa mitjana. Això indica un compromís entre optimitzar la longitud del camí i maximitzar la recompensa.

Aquestes observacions revelen que, en REINFORCE, l'enginyeria de recompenses pot tenir un impacte significatiu en el rendiment i en el tipus de comportament que l'agent

aprèn a executar.

Nombre d'episodis d'entrenament

Segons la Figura 25, podem veure que REINFORCE aconsegueix una millora ràpida durant els primers 500 episodis, però continua millorant gradualment fins als 3000 episodis. Aquest patró d'aprenentatge és més ràpid que el que vam observar amb l'Estimació Directa, però no tan immediat com el de Q-Learning.

El nombre d'episodis necessaris per a un bon rendiment és més alt que en els algoritmes basats en model, però això és esperable donat que REINFORCE requereix múltiples trajectòries completes per reduir la variància en l'estimació del gradient de la política.

7.4 Fortaleses i debilitats

Després d'analitzar el comportament de REINFORCE en l'entorn CliffWalking i comparar-lo amb els altres algoritmes estudiats, podem identificar les seves principals fortaleses i debilitats:

Fortaleses

1. **Ràpida convergència**: És un algorisme més ràpid (el temps d'execució és menor), ja que no triga a convergir (almenys en aquest entorn).
2. **Estabilitat amb taxes d'aprenentatge altes**: Aconsegueix un bon rendiment amb valors per a taxes d'aprenentatge alts, com ara 0.05.
3. **Aplicabilitat a entorns complexos o amb espais d'estats grans**: Tot i que no es pot apreciar en el nostre entorn, l'algorisme REINFORCE pot aplicar-se a problemes d'altíssima dimensionalitat sense grans complicacions, a diferència de mètodes anteriors.
4. **Optimització directa de la política**: A diferència dels mètodes basats en valor com Q-Learning o Iteració de Valor, REINFORCE optimitza directament la política sense necessitat d'estimar una funció de valor.
5. **Aprenentatge de polítiques estocàstiques**: REINFORCE aprèn naturalment polítiques probabilístiques, la qual cosa pot ser beneficiós en entorns parcialment observables o en aquells on l'exploració continuada és important.

Debilitats

1. **Gran sensibilitat al canvi de paràmetres**: El rendiment del REINFORCE depèn fortament de la configuració de paràmetres com la taxa d'aprenentatge o el factor de descompte. Una petita variació d'aquests valors pot afectar greument els resultats.
2. **Taxes d'èxit molt baixes**: Les taxes d'èxit són extremadament baixes, més que a la resta dels algoritmes, com es mostra a la Figura 27.
3. **No és *off-policy***: A diferència de Q-Learning, REINFORCE necessita d'una política inicial per a aprendre.

4. **Alta variància en els gradients**: REINFORCE pateix d'alta variància en les estimacions del gradient, el que pot fer que l'aprenentatge sigui més sorollós i menys eficient. Això es manifesta especialment amb valors molt alts de gamma.
5. **Requereix episodis complets**: En ser un algoritme Monte Carlo, REINFORCE necessita completar episodis sencers abans de poder actualitzar la política, el que pot ser ineficient en episodis llargs o en entorns on les recompenses són escasses.

En el context de l'entorn CliffWalking, REINFORCE es posiciona com una alternativa interessant als mètodes basats en model i a Q-Learning, mostrant un aprenentatge ràpid inicial i una bona adaptació a diferents funcions de recompensa. No obstant això, la seva dificultat per convergir a polítiques deterministes òptimes i la seva sensibilitat al factor de descompte el fan menys robust en aquest entorn específic que algoritmes com la Iteració de Valor.

REINFORCE resulta especialment adequat per a entorns on es requereix una política estocàstica i on l'espai d'estats o accions és massa gran per a mètodes basats en taules. En el cas de CliffWalking, tot i que aconsegueix un rendiment acceptable, no supera significativament els mètodes estudiats anteriorment en termes de recompensa mitjana o taxa d'èxit.

8 Comparació global i conclusions

Després d'analitzar en profunditat els quatre algoritmes d'aprenentatge per reforç (Iteració de Valor, Estimació Directa, Q-Learning i REINFORCE) en l'entorn CliffWalking, podem realitzar una comparació global dels seus rendiments i verificar les hipòtesis inicials.

8.1 Verificació de les hipòtesis

A l'inici d'aquest estudi, havíem plantejat quatre hipòtesis principals. Ara podem avaluar-les a la llum dels resultats obtinguts:

1. **Els algoritmes basats en model (Iteració de Valor i Estimació Directa) convergiran més ràpidament que els basats en experiència (Q-Learning i REINFORCE).**

   Aquesta hipòtesi s'ha confirmat parcialment. L'Iteració de Valor ha mostrat una convergència molt ràpida, necessitant només un nombre limitat d'iteracions per trobar la política òptima. No obstant això, l'Estimació Directa ha requerit un nombre significatiu d'episodis per convergir, ja que necessita temps per construir un model fiable de l'entorn.

   Pel que fa als mètodes basats en experiència, Q-Learning ha mostrat una convergència sorprenentment ràpida amb valors adequats de taxa d'aprenentatge i decaïment d'epsilon, especialment visible a la Figura 20. REINFORCE, tot i mostrar una millora ràpida inicial (Figura 25), ha necessitat molts més episodis (3000) per assolir un rendiment estable.

2. **El factor de descompte tindrà un impacte significatiu en el comportament de risc de l'agent.**

   Aquesta hipòtesi s'ha confirmat clarament en tots els algoritmes. Hem observat que:

   - En Iteració de Valor i Estimació Directa, valors més alts de gamma (0.95-0.99) han afavorit comportaments més conservadors que eviten eficaçment el penya-segat.
   - En Q-Learning, gamma=0.9 ha proporcionat el millor equilibri entre eficiència i seguretat.
   - En REINFORCE, sorprenentment, valors més baixos de gamma han donat millors resultats, mostrant una tendència contrària als altres algoritmes.

   Aquestes observacions confirmen que el factor de descompte és crucial per determinar l'equilibri entre recompenses immediates (camí curt però arriscat) i futures (camí més llarg però segur).

3. **Q-Learning mostrarà una major sensibilitat als hiperparàmetres com la taxa d'aprenentatge i el coeficient d'exploració.**

   Aquesta hipòtesi s'ha confirmat. Els experiments mostren que Q-Learning és extremadament sensible a la selecció de hiperparàmetres. La Figura 21 revela com la combinació de gamma i taxa d'aprenentatge afecta dràsticament el rendiment, amb una diferència de més del 10% en la recompensa mitjana entre la millor combinació (gamma=0.9, alpha=0.35) i les menys òptimes.

De manera similar, el decaïment d'epsilon ha demostrat ser crític, amb diferències significatives en la velocitat de convergència (Figura 20). Aquesta sensibilitat és superior a la mostrada per Iteració de Valor o Estimació Directa.

4. **REINFORCE requerirà un major nombre d'episodis per convergir, però potencialment pot trobar polítiques més òptimes en entorns estocàstics.**

   La primera part d'aquesta hipòtesi s'ha confirmat: REINFORCE ha requerit 3000 episodis per convergir, considerablement més que els altres algoritmes. No obstant això, la segona part no s'ha verificat. Les polítiques trobades per REINFORCE no han superat les dels altres algoritmes en termes de recompensa mitjana o taxa d'èxit. De fet, la política apresa mostra algunes incoherències (Figura 22), i la taxa d'èxit roman extremadament baixa (Figura 27), similar a la dels altres mètodes.

   Això suggereix que, almenys per a l'entorn CliffWalking estocàstic, la naturalesa probabilística de REINFORCE no ofereix un avantatge significatiu sobre els mètodes basats en valor.

8.2 Conclusions finals

De l'estudi exhaustiu realitzat, podem extreure diverses conclusions globals sobre el rendiment i les característiques dels quatre algoritmes estudiats:

Comparativa de rendiment

1. **Recompensa mitjana**: Q-Learning ha aconseguit la millor recompensa mitjana, seguida per Iteració de Valor, REINFORCE i finalment Estimació Directa. Aquest resultat és notablement interessant perquè mostra que un mètode model-free com Q-Learning pot superar els mètodes basats en model en determinats entorns.
2. **Eficiència computacional**: Iteració de Valor ha mostrat la major eficiència computacional, requerint només uns centenars d'iteracions per convergir. Q-Learning mostra una bona relació entre velocitat de convergència i qualitat de la solució. Estimació Directa i REINFORCE requereixen més episodis d'entrenament, el que els fa menys eficients computacionalment.
3. **Taxa d'èxit**: Sorprenentment, tots els algoritmes mostren taxes d'èxit extremadament baixes en l'entorn estocàstic (is_slippery=True). Això suggereix que el problema fonamental no rau en els algoritmes sinó en la pròpia naturalesa de l'entorn, on la component aleatòria fa que sigui molt difícil arribar a l'objectiu de manera consistent.
4. **Polítiques apreses**: Tots els algoritmes aprenen polítiques que eviten el penya-segat, optant per un camí segur per la part superior de la graella. No obstant això, les polítiques de Iteració de Valor i Q-Learning mostren més coherència global que les d'Estimació Directa i REINFORCE.

Avantatges i inconvenients comparatius

- **Iteració de Valor**:
  - *Avantatges*: Garantia teòrica de convergència a la política òptima, velocitat de convergència ràpida, polítiques coherents.

  - *Inconvenients*: Requereix coneixement complet del model de l'entorn, dificultats d'escalabilitat a espais d'estat grans, baixa robustesa en entorns altament estocàstics.

- **Estimació Directa**:
  - *Avantatges*: No requereix coneixement previ del model, adaptabilitat a canvis en l'entorn, planificació eficient.
  - *Inconvenients*: Convergència més lenta, alta sensibilitat als paràmetres de planificació, cost computacional creixent amb els passos de planificació.
- **Q-Learning**:
  - *Avantatges*: No requereix model de l'entorn, actualització contínua per cada pas, bon equilibri entre exploració i explotació, excel·lent rendiment amb paràmetres ben ajustats.
  - *Inconvenients*: Alta sensibilitat als hiperparàmetres, potencial lentitud de convergència en espais d'estat grans, risc de convergència a polítiques subòptimes.
- **REINFORCE**:
  - *Avantatges*: Optimització directa de la política, capacitat per aprendre polítiques estocàstiques, adaptabilitat a diferents funcions de recompensa, estabilitat amb taxes d'aprenentatge altes.
  - *Inconvenients*: Alta variància en els gradients, requereix episodis complets per actualitzar la política, dificultat per convergir a polítiques deterministes òptimes, sensibilitat extrema al factor de descompte.

Elecció d'algoritme segons el context

Dels resultats obtinguts, podem extreure recomanacions sobre quins algoritmes serien més adequats en diferents escenaris:

- **Entorns amb model conegut i espai d'estats petit**: Iteració de Valor seria l'opció més eficient i garantida.
- **Entorns dinàmics o amb model parcialment conegut**: Estimació Directa ofereix un bon compromís entre usar el coneixement disponible i adaptar-se a canvis.
- **Entorns sense model conegut i espai d'estats moderat**: Q-Learning proporciona el millor rendiment quan s'ajusten correctament els hiperparàmetres.
- **Entorns amb espais d'estat o acció grans/continus**: REINFORCE seria preferible, especialment amb una parametrització adequada de la política.
- **Entorns altament estocàstics**: Cap dels algoritmes estudiats ha demostrat ser clarament superior, suggerint que podrien ser necessàries versions més avançades com Actor-Critic o modificacions específiques.

En conclusió, aquest estudi mostra la complexitat inherent a l'aprenentatge per reforç i la importància d'escollir l'algoritme adequat segons les característiques específiques del problema. També destaca que, malgrat les diferències teòriques entre els algoritmes, tots enfronten limitacions similars en entorns altament estocàstics, suggerint que la recerca futura hauria de centrar-se en desenvolupar mètodes més robustos davant l'aleatorietat.

9 Referències

1. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
2. Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3), 229-256.
3. Material de les sessions teòriques i de laboratori de SID: https://sites.google.com/upc.edu/grau-sid
4. Documentació de Gymnasium: https://gymnasium.farama.org/environments/toy_text/cliff_walking/