TP 2 : Equation de Bellman et Programmation Dynamique.

(a rendre à l'adresse : santucci@univ-corse.fr)

Exercices à partir de l'exemple Frozen Lake

1. Value itération

1.1 Fonction Value_iteration

Ecrire une fonction appelée Value_iteration qui permet de calculer la fonction de valeur optimale de manière itérative en prenant le maximum sur la fonction Q c'est à dire :

$$V^*(s) = \max_a Q^*(s,a)$$

Conformément à l'algorithme vu en cours il vous faudra initialiser :

- le nombre d'itérations (1000 par exemple)
- un nombre seuil (threshold) qui permet de vérifier la convergence de la fonction Value_iteration (à 1e-20)
- le discount facteur gamma à 1 (comme en cours)
- initialiser la table des valeurs à 0 pour tousses états (du style : value_table = np.zeros(env.observation_space.n)

A chaque itération:
-Vous mettrez tout d'abord à jour la value_table (car on a vu en cours que on utilise à chaque itération les valeurs définies lors de l'itération précédente)
-Ensuite vous devez calculer

$$V^*(s) = \max_a Q^*(s,a)$$

Avec

$$Q(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$$

- Mettre à jour la value_table avec la nouvelle valeur calculée.

Apres avoir calculé la table des valeurs, (donc la valeur de tous les états), on vérifie si la différence entre la table des valeurs obtenues à l'itération courante et l'itération précédente est inférieure ou égale à la valeur seuil (threshold). Si la valeur est inférieure au seuil il faut arrêter la boucle et renvoyer la table des valeurs (fonction code valeur optimale trouvée).

Donc ont calculé la fonction de valeur optimale en prenant le maximum des Q values. Il faut maintenant extraire la politique optimale.

1.2. Extraction de la politique optimale à partir de la la fonction de valeur optimale calculée en 1.1

On va tout d'abord définir une fonction extract_policy qui prend en entre une table des valeurs (value_table)

Initialiser le discount factor gamma à 1.
Initialiser la politique avec des zéros (on initialise les actions pour tous les états à zero.
Vous devez ensuite calculer la Q fonction en utilisant la fonction de valeur optimale obtenue à l'étape précédente.
Rappel : la fonction Q peut être calculée avec

$$Q(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$$

Apres avoir calculé la Q fonction, on peut extraite la politique en sélectionnant l'action qui a la valeur Q maximale. Comme on calcule la Q fonction en utilisant la fonction de valeur optimale, la politique extraite de la fonction Q sera la politique optimale donc

$$\pi^* = \arg\max_a Q(s,a)$$

Donc vous allez calculer pour chaque état les valeurs Q pour toutes les actions de l'état puis extraire la politique en sélectionnant l'action qui a la valeur Q maximale.
On pourra alors retourner la politique optimale.

1.3. Résolution du Frozen Lake en intégration les deux fonctions précédentes

Nous avons vu que dans l'environnement Frozen Lake, l'objectif est de trouver la politique optimale qui sélectionne l'action correcte dans chaque état afin que nous puissions atteindre l'état G à partir de l'état A sans visiter les états trou.
Donc on calcule tout d'abord la fonction de valeur optimale à l'aide de la fonction value_iteration de 1.1 en passant en paramètre l'environnement Frozen Lake.
Ensuite vous allez extraire la politique optimale de la fonction de valeur optimale à l'aide la fonction extract_policy du 1.2.
Vous aurez le résultat en affichant cette politique optimale.

2. Policy iteration

Nous avons vu en cours que dans la méthode « policy itération » on calcule la Value fonction en utilisant la politique d manière itérative. Une fois que la la value fonction optimale est trouvée , la politique utilisée pour calculer la fonction de leur optimale sera la politique optimale.
Donc 1ere étape calcul de la valeur fonction à l'aide de la politique.

2.1. Calculer la value fonction à l'aide de la politique.

Cette étape est exactement la même que la façon dont vous avez calculé la valeur fonction dans la méthode précédente. Mais il y a une petite différence : on calcule la value fonction en utilisant la politique alors que précédemment on calculait la valeur fonction en prenant le maximum des Q values.

Donc vous devez écrire une fonction appelée compute_valeur_function qui prend en paramètre une politique.
ON aura besoin de définir:
- le nombre d'itérations
- le seuil (threshold)
- le discount factor (gamma) fixé à 1 dans notre exemple

On doit aussi initialiser la value_table à zéro.

Pour chaque itération :
-il faut mettre à jour la value_table à l'aide des valeurs calculées à l'itération précédente.
-Il faut ensuite calculé la value fonction en utilisant la politique donnée en paramètre. le calcul d'une value fonction peut être calculée selon une politique par :

$$V^{\pi}(s) = \sum_{s'} P^{\pi}_{ss'}[R^{a}_{ss'} + \gamma V^{\pi}(s')]$$

Donc pour chaque état, il. faut sélectionner l'action en fonction de la politique , puis mettre à jour la valeur de l'état en utilisant l'action sélectionnée. Vous avez donc calculée la value_table (la valeur de tous les états).
-on doit ensuite vérifier si la différence entre la table des valeurs (value_table) obtenue à l'itération courante et l'itération précédente est inférieure ou égale à la valeur seuil (threshold). Si il est inférieur, nous arrêtons la boucle et on renvoie la table des valeurs (value_table) que nous avons calculée qui est donc le bon résultat.

Une fois que l'on a calculé la Value fonction (table des valeurs) associée à la politique donnée en paramètre grâce à compute_value_function(policy), il faut voir comment extraire la politique de la value fonction.

2.2. Extraction de la politique à partir de la value fonction.

Cette étape est exactement la même que celle que vous avez définie dans la méthode de value itération précédente.
Vous écrirez donc la fonction extract_policy (value_function).

2.3. Intégration des deux étapes précédentes

Vous allez définir une fonction policy_iteration qui prend un environnement en paramètre .

Pour chaque itération, vous utiliserez compute_value_function (policy) pour calculer la value_function.
Puis la nouvelle politique grâce à extract_policy(value_function).
Vous versifiez si la nouvelle politique est identique à la politique de l'itération precedenete . Si oui c'est fini la politique a été trouvée.

Valider le résultat obtenu sur l'environnement du FrozenLake