Multi-armed bandits

Eden BELOUADAH

1 Introduction

Le but de ce travail pratique est d'implémenter plusieurs bandits à bras multiples et voir la différence entre eux.

2 Bandits à plusieurs bras

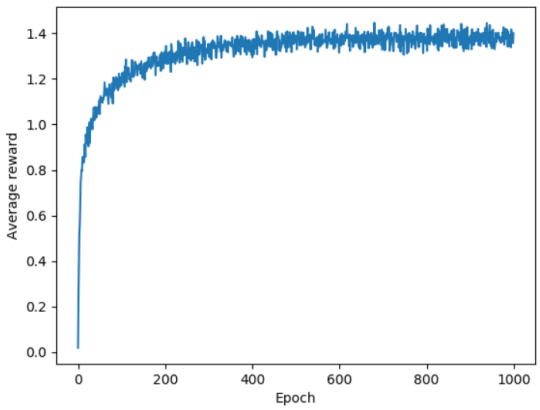
Un bandit à plusieurs bras est un agent qui a K bras et à chaque pas de temps, il essaye de choisir un bras de telle manière à faire un compromis entre l'exploration et l'exploitation. Le but étant de minimiser le regret total après un certain horizon de temps T.

2.1 Agent ϵ-greedy

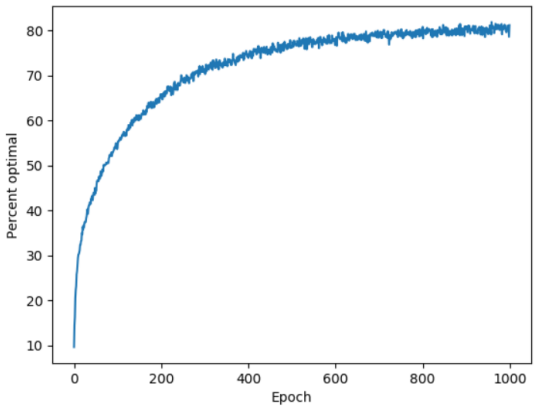
Cet agent choisit à chaque fois avec une probabilité d'exploitation $1 - \epsilon$ le bras ayant la moyenne de récompenses la plus grande depuis le début de la partie, et avec une probabilité d'exploration ϵ, un bras aléatoire uniformément choisi.

$$i_t = argmax\{\hat{\mu}_{1,t}, \hat{\mu}_{2,t}, ..., \hat{\mu}_{k,t}\}$$

Les résultats obtenus sont les suivants :



(a) Évolution des récompenses

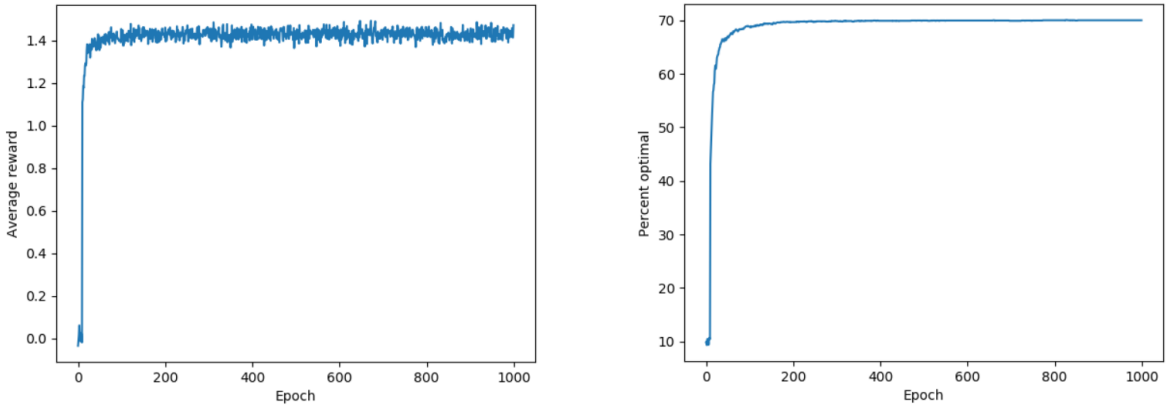


(b) Évolution de l'optimum

Figure 1. Résultats obtenus pour l'agent ϵ-greedy

2.2 Agent Optimistic ϵ-greedy

Cet agent est le même que le précédent, sauf qu'il initialise les Q_valeurs de chaque bras non pas à 0 mais à une valeur appelée *optimism*. Les résultats obtenus sont les suivats :



(a) Évolution des récompenses *(b) Évolution de l'optimum*

Figure 2. Résultats obtenus pour l'agent Optimistic ϵ-greedy

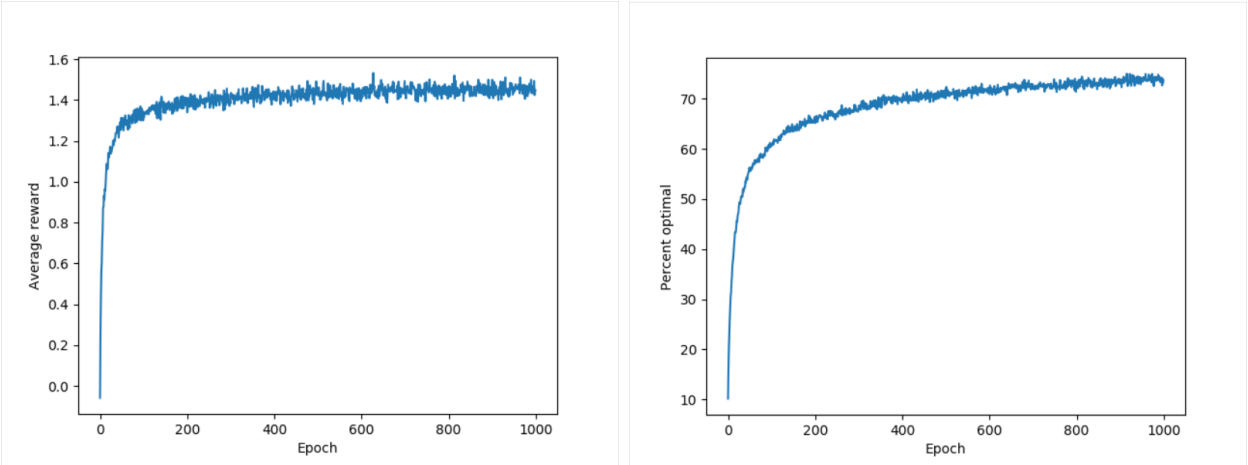
Nous remarquons que l'initialisation des Q_valeurs des bras a un grand impact sur la qualité de l'agent. En effet, la récompense fait un saut énorme vers des valeurs optimales peut perturbées au début de l'exécution.

2.3 Agent SoftMax

Cet agent choisit à chaque itération un bras avec une probabilité (τ est la température) :

$$p_k = \frac{e^{\frac{\hat{\mu}_k}{\tau}}}{\sum_{i=1}^{k} e^{\frac{\hat{\mu}_i}{\tau}}}$$

Les résultats obtenus sont les suivants :



(a) Évolution des récompenses *(b) Évolution de l'optimum*

Figure 3. Résultats obtenus pour l'agent SoftMax

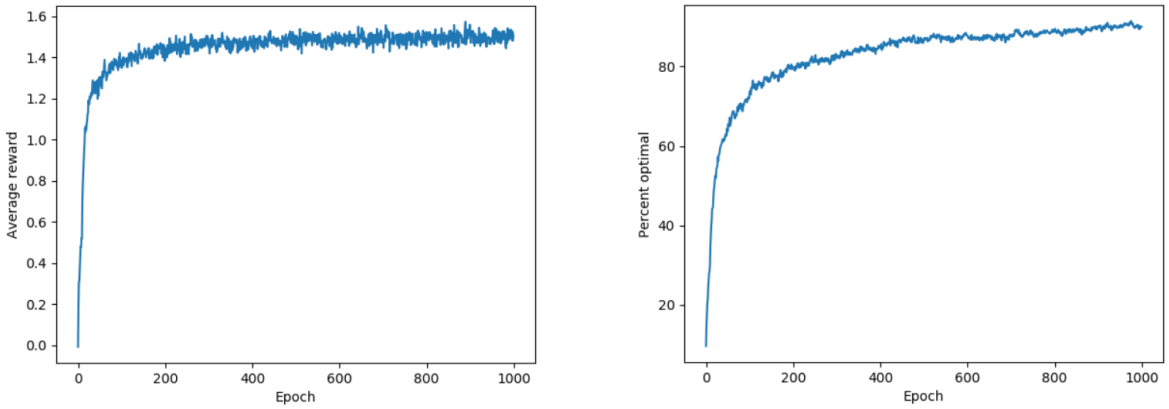
Nous remarquons que la récompense de l'agent augmente rapidement et ensuite elle est moins perturbée que celle de l'agent ϵ-greedy.

2.4 Agent UCB (Upper Confidence Bound)

Cet agent choisit le bras maximisant la moyenne des récompenses plus l'intervalle de confiance ($n_{i,t}$ est le nombre de fois où le bras i a été choisi du début jusqu'à l'instant t) :

$$i_t = argmax\{\hat{\mu}_{i,t} + \sqrt{2\frac{Log(\sum n_{j,t})}{n_{i,t}}}\}$$

Les résultats obtenus sont les suivants :



(a) Évolution des récompenses *(b) Évolution de l'optimum*

Figure 4. Résultats obtenus pour l'agent Thompson

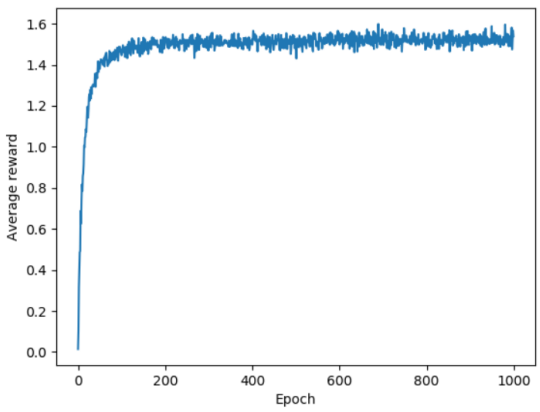
De même, nous remarquons que la récompense de l'agent augmente rapidement et presque de la même façon que celle de l'agent SoftMax. De plus, l'évolution dans l'optimum est meilleure et plus stable.

2.5 Agent Thompson

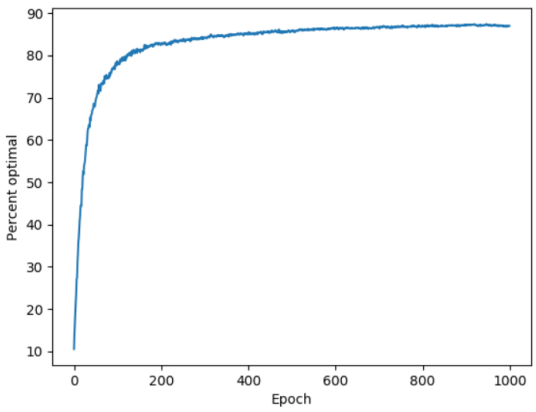
Cet agent échantillonne selon la loi normale et choisit le meilleur bras comme suit :

$$i_t = argmax\{\mathcal{N}(\hat{\mu}_{i,t}, \frac{1}{n_{i,t}+1})\}$$

Les résultats obtenus sont les suivants :



(a) Évolution des récompenses



(b) Évolution de l'optimum

Figure 5. Résultats obtenus pour l'agent Thompson

Les résultats pour cet agent sont les meilleurs, la récompense continue à augmenter jusqu'à arriver à de très grande valeurs non perturbées.

3 Conclusion

En comparant le temps d'exécution des agent, on trouve :

Agent	ϵ-greedy	Optimistic ϵ-greedy	SoftMax	UCB	Thompson
Temps d'exécution(s)	20.52	24.02	189.29	60.25	37.40

Tableau 1. Temps d'exécution des bandits

Nous remarquons que l'agent le plus rapide est ϵ-greedy (avec les deux versions simple et optimiste) vu qu'il est naïf et n'effectue aucun traitement particulier à part la comparaison avec la probabilité d'exploration ϵ.

En revanche, l'agent le plus lent est SoftMax, et cela parce qu'il calcule à chaque itération l'exponentiel de la fraction de la moyenne des récompenses par la température pour tous les bras afin de pouvoir calculer la probabilité de choisir chaque bras.

Les deux agents UCB et Thompson sont plus ou moins rapides.