DRL Blatt 1

Mathias Oehmen, Amin Tafla

October 2024

Aufgabe 1.1

(a)

Randomly generated Bernoulli bandit has reward probabilities:
$[0.3745401188473625, 0.9507143064099162, 0.7319939418114051, 0.5986584841970366]$
The best machine has index: 1; and probability: 0.9507143064099162
The highest probability resembles the optimal action.

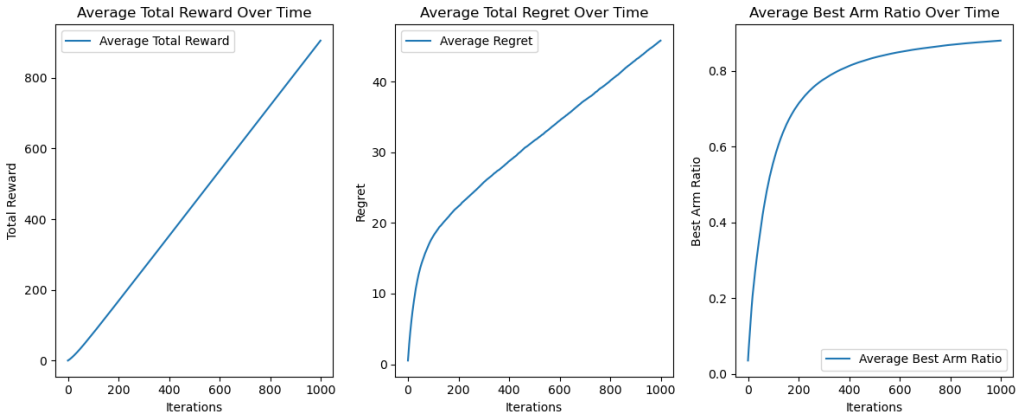

Abbildung 1: $\epsilon = 0.1$; 1000 Iterations; average of a total of 1000 runs

Zu Figur 1 1:
Man erkennt eine initiale abgeflachte Total Rewards Kurve für die ersten

Iterationen, bis der Agent die optimale Action gefunden hat. Danach erhöht sich der Total Reward fast linear mit den Iterations.

Außerdem sehen wir sehen eine Stauchung des Total Regrets over Time nach ungefähr 100 Iterations. Das vorherige exponentielle Wachstum ist auf suboptimale Entscheidungen bei der Wahl der aktuell optimalen Action des Agents zurückzuführen, bis sich die optimale Action verfestigt und somit die Total Regret Kurve abflacht.

Mit steigenden Iterations sieht man, dass der prozentuale Anteil der Wahl des besten Arms sich ϵ annähert. Dies ist damit zu begründen, dass sich eben mit steigender Iterationen die optimale Action herauskristallisiert und dementsprechend mit genügend Iterationen mit Wahrscheinlichkeit $1-\epsilon$ vom Agent gewählt wird.

(b)

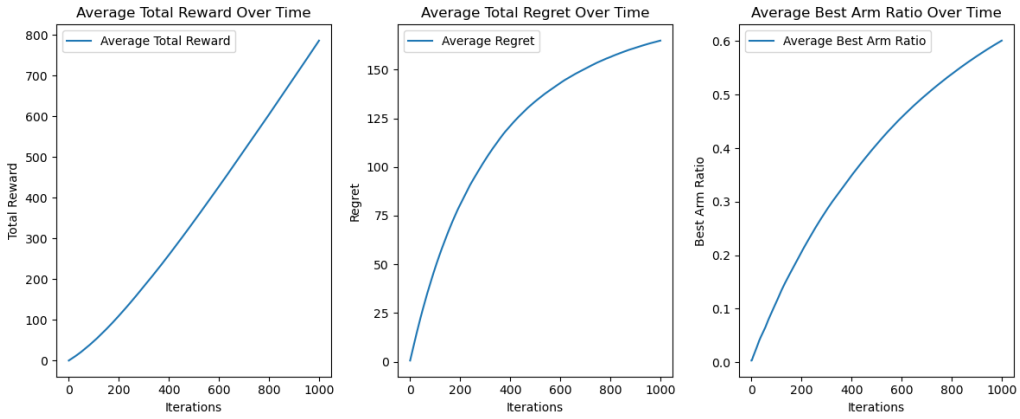

Abbildung 2: $\epsilon = 0.01$; 1000 Iterations; average of a total of 1000 runs

Mit ϵ wird der Anteil an Exploration zu den gesamten Steps gesetzt. Demnach ist eben der Anteil an Exploration von unseren gesamten Iterationen, entweder 0.1, 0.01 oder 0.2 und jeweils 0.9, 0.99 und 0.8 für Exploitation.

Im Vergleich zu 1 mit $\epsilon = 0.1$ sehen wir in 3 mit $\epsilon = 0.2$ eine frühzeitige Abflachung des exponentiellen Wachstums des Total Regrets, welche darauf zurückzuführen ist, dass nun in mehreren Iterationen erkundet wird, wodurch sich die optimale Action schneller herauskristallisieren kann. Gleichzeitig sehen wir jedoch auch einen insesamt höheren Total Regret, dadurch, dass

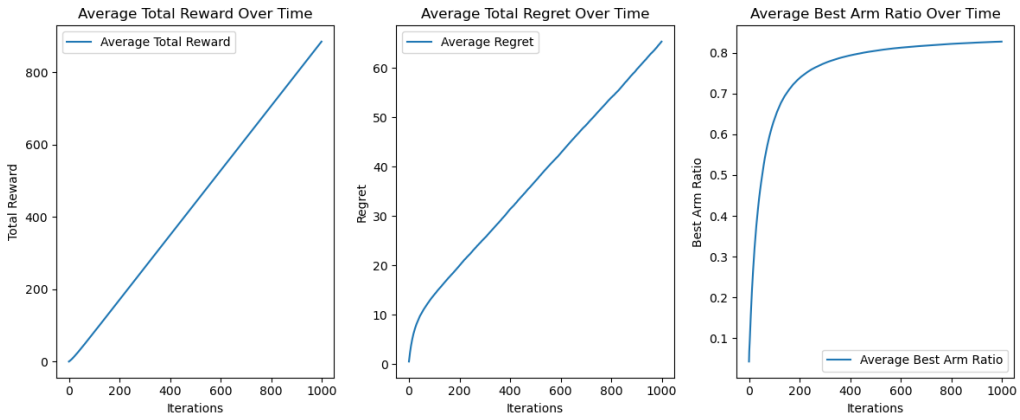

Abbildung 3: $\epsilon = 0.2$; 1000 Iterations; average of a total of 1000 runs

auch nachdem die optimale Action gefunden wurde, weiterhin zu 20% erkundet wird. Durch die höhere Erkundungsrate ϵ sehen wir auch eine schnellere Einstellung von linearem Wachstum des Total Rewards, vermutlich da durch mehr Erkunden analog zum Total Regret, die optimale Action schneller gefunden wird, während bei kleineren ϵ, wie in 1 und 2 der Agent längere Zeit mit suboptimalen Actions verbringt. Außerdem konvergiert hierdurch auch der prozentuelle Anteil des optimalen Arms schneller zu $1 - \epsilon$.

Ein $\epsilon = 0.01$ wie in 2 zeigt genau das gegenteilige Verhalten. Die Findung der optimalen Action dauert länger, wodurch sich die Kurve des Total Rewards abflacht und länger bis zu linearem Wachtum benötigt, ebenso steigt die Kurve des Total Regrets stark an und flacht erst nach deutlich mehr Iterations ab., was zu einem deutlich höheren Total Regret führt. Das Verhalten wird auch in dem Verlauf des prozentuellen Anteil des optimalen Arms klar, welcher deutlich später mit $1-\epsilon$ konvergiert, was die verspätete Verfestigung der optimalen Action nochmal verdeutlicht.

Abschließend lässt sich sagen, dass durch eine Erhöhung von ϵ eine schnellere Verfestigung der optimalen Action erzielt wird. Analog durch eine Reduktion eben eine langsamere Verfestigung. Gleichzeitig sollte jedoch ϵ nicht zu hoch angesetzt werden, da dadurch selbst nach einer klaren Verfestigung der optimalen Action weiterhin viele zufällige Actions ausgeführt werden, welche wiederum im späteren Verlauf das Lernen signifikant beeinträchtigen können.

(c)

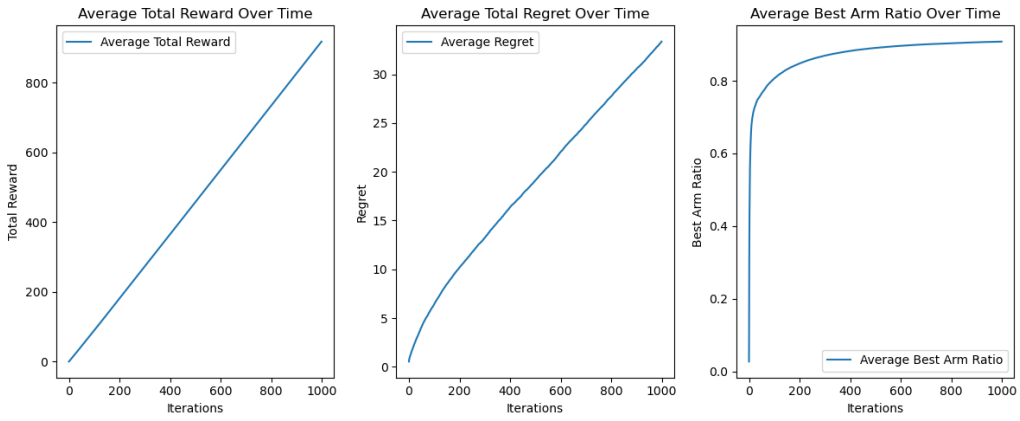

Abbildung 4: $\epsilon = 0.1$; 1000 Iterations; avgs of 1000 runs with initial $Q = 0.8$

Durch die Initialisierung der Schätzwerte mit einem hochangesetzten Wert von 0.8 sehen wir eine deutlich schnellere Konvergenz des prozentuellen Anteils des optimalen Werts. Dies wird dadurch ermöglicht, dass bereits zu Beginn der Wert der tatsächlichen Wahrscheinlichkeit eines Reward ähnelt und somit die anfängliche Varianz reduziert wird, z.B. in Fällen in denen in der ersten Ausführung der Action 0 als Reward erzielt wird, trotz hoher Wahrscheinlichkeit eines Rewards durch die optimale Action. In solchen Fällen wird die Erkundung zwischenzeitlich stark angeregt, da es in den anfänglichen Iterationen leichter dazu kommt, dass die zu dem Zeitpunkt optimale Action von der Tatsächlichen abweicht. Dies und die anfängliche kleine Differenz zwischen tatsächlicher Wahrscheinlichkeit und Schätzwert führt zu einer schnelleren Kristrallisierung der optimalen Action. Außerdem sehen wir eine anfängliche flachere Steigung des Total Regrets und frühzeitige Linerarität des Wachstums des Total Rewards, welche durch eine frühzeitge Kristallisierung der optimalen Action erzielt werden.

(d)

Wie man in 5 erkennt, führt ein Decay von ϵ, wie zu erwarten zu einem höheren prozentuellen Anteil der optimalen Action nach einer vollständigen Ausführung, da nun eben kein konstantes Limit durch ϵ mehr gesetzt ist.

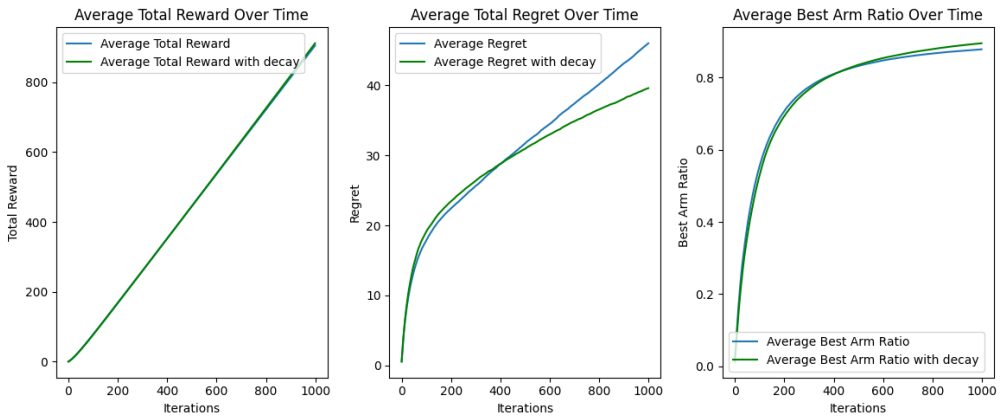

Abbildung 5: decaying $\epsilon = 0.1$; 1000 Iterations; averages of 1000 runs

Gleichzeitig sehen wir bei niedrigeren Wiederholungen, ungefähr im Bereich bis 400 Iterationen der durchschnittliche optimale Action Anteil sogar niedriger ist mit einem eingebauten Decay. Dies kommt dadurch zustande, dass ϵ kontinuierlich (jedoch nicht linear) reduziert wird, wodurch am Anfang weniger erkundet wird und dadurch der Algorithmus länger braucht um die optimalen Action zu verfestigen. Im Vergleich des Total Regrets sieht man einen deutlichen Unterschied. Eine Implementation eines Decays führt zu einem signifikant niedrigerem Total Regret, nach genügend Iterationen, während in den ersten hunderten Iterationen der akkumulierte Regret noch höher ist in dem Algorithmus mit Decay. Dies ist auf die vorher erwähnte längere Zeit für die Verfestigung des optimalen Arms zurückzuführen, während der insgesamt niedrigere Total Regret nach Abschluss aller Iterationen auf das schrumpfende ϵ zurückzuführen ist, wodurch nach Verfestigung des optimalen Arms, dieser eben prozentuell vermehrt ausgeführt wird. Wohingegend bei dem ϵ-greedy Algorithmus ohne decay dieses Konstant bleibt und demnach bei $1 - \epsilon$ stagniert. Bei dem Average Total Reward sieht man bei 1000 Iterationen kaum einen Unterschied, die Ausführung mit decay ist dabei nur minimal besser. (Dies geht aus der Grafik in 5 kaum heraus, man sieht aber hoffentlich ganz am Ende der 1000 Iterationen die blaue Linie leicht unter der grünen Linie.) Allerdings wird sich dieses Verhalten stärker wiederspiegeln mit mehr Iterationen, da bei der Ausführung ohne decay, das ϵ natürlich gleich bleibt, wodurch die optimale Action zu konstant $\epsilon\%$ und eine zufällige zu $1-\epsilon\%$ ausgeführt wird, während mit einem Decay, dieses Verhältnis immer

weiter zu Gunsten der optimalen Action sich abändert.

Somit führt ein Decay zu vorteilhafteren Werten im Gegensatz zu einer Ausführung ohne Decay, jedoch auf Kosten der Effizienz der Verfestigung auf den optimalen Arm, folglich werden mehr Iterationen benötigt.

Aufgabe 1.2

(a)

Bei dem k-armigen Bandit-Problem hat ein Agent die Wahl zwischen k verschiedenen Optionen (Armen) auszuwählen, wobei jede Option einen Reward mit unbekannter Wahrscheinlichkeit liefert. Das Ziel des Agents besteht darin, den kummulierten Reward über eine Reihe von Itterationen zu maximieren in dem er zwischen Exploration und Exploitation abwägt, also dem Erkunden neuer Arme und der Ausbeutung bereits als gut identifizierter Arme.

Bei einem nicht-stationären k-armigen Banditen ändern sich die Wahrscheinlichkeiten der Arme im Verlauf der Zeit. Nicht-Stationarität kann auf verschiedene Arten modelliert werden:

- Die Wahrscheinlichkeiten der Arme ändern sich nach einer bestimmten Anzahl an Itterationen.
- Die Wahrscheinlichkeiten ändern sich konstant über die Itterationen hinweg.
- Die Wahrscheinlichkeiten ändern sich mit einer bestimmten Wahrscheinlichkeitsverteilung (bspw. der Noramlverteilung mit Parameter μ und σ)

Um das nicht-stationäre Bandit Problem zu lösen muss ein Algorithmus genutzt werden der ebenfalls dynamisch auf die Veränderung der Wahrscheinlichkeiten reagiert. Ein Ansatz dafür wäre die gewichtete Durchschnittsbildung: Beispielsweise könnte der ϵ-greedy Algorithmus dahingehend angepasst werden, dass für die Schätzungen der erwarteten Belohnungen für jeden Arm ältere Beobachtungen weniger ins Gewicht fallen durch das Einführen einer Vergessensrate α.

Ein weiterer Ansatz besteht darin nur die letzten x Beobachtungen für die Schätzung der Rewards zu verwenden, da sich der Algorithmus so schneller and Veränderungen anpassen kann.

Im Fall des nicht-stationären k-armigen Banditen muss der Agent häufiger zu Erkundung tendieren, um veränderte Bedingungen zu erkennnen.

Zur Evaluierung eines Algorithmus für das k-armige Banditen-Problem können mehrere Metriken betrachtet werden. Ein mögliches Gütekriterium ist der "Total Reward Over Time", also die Gesamtbelohnung die der Agent im Laufe der Zeit gesammelt hat. Eine weitere Mertrik die betrachtet werden kann ist der "Total Regret Over Time"der den Unterschied zwischen der tatsächlichen Belohnung und der optimalen Belohnung über den Gesamtzeitraum errechnet. Im nicht-stationären Fall muss ein dynamischer Regret verwendet werden, der den Unterschied zu der in dem Zeitpunkt(t) besten Strategie misst. Ein drittes Gütekriterium, was für den nicht-stationären Fall verwendet werden könnte, wäre die Anpassungsgeschwindigkeit, die misst wie schnell der Algorithmus auf Änderugen der Reward-Wahrscheinlichkeiten reagiert.

(b)

Siehe Programmcode.

(c)

In Abbildung 6 kann bei dem "Average Total Reward Over Time" erkennen, dass die Linie annähernd linear ansteigt, was dafrauf hindeutet, dass der Algrorithmus im Laufe der Zeit stetig höhere Belohnungen sammelt. Im vergleich zu der stationären k-armigen Algorithmus wird jedoch deutlich, das der ϵ-greedy Algorithmus hier nur einen Total Reward von ca. 720 erreicht, während im Fall der in Abbildung 2 zu sehen ist ca. 850 Rewards erzielt werden. Hier erschwert die Nicht-Stationarität die langfristige Ausbeutung (Exploitation).

In Abbildung 7 kann man die kummulierte Verteilung der gewählten Aktionen im Zeitpunkt t beobachten. Arm 1 wird insgesamt am häufigsten gewählt. Dies ist darauf zurückzuführen, dass im Zeitpunkt $t = 0$ alle Wahrscheinlichkeiten 50% betragen und in dem Fall Arm 1 (mit Index 0) als erstes gewählt wird mit Wahrscheinlichkeit $1 - \epsilon$. Es fällt auf das die kummulierten Verteilungen der gewählten Arme im Verlauf der Zeit annhähernd

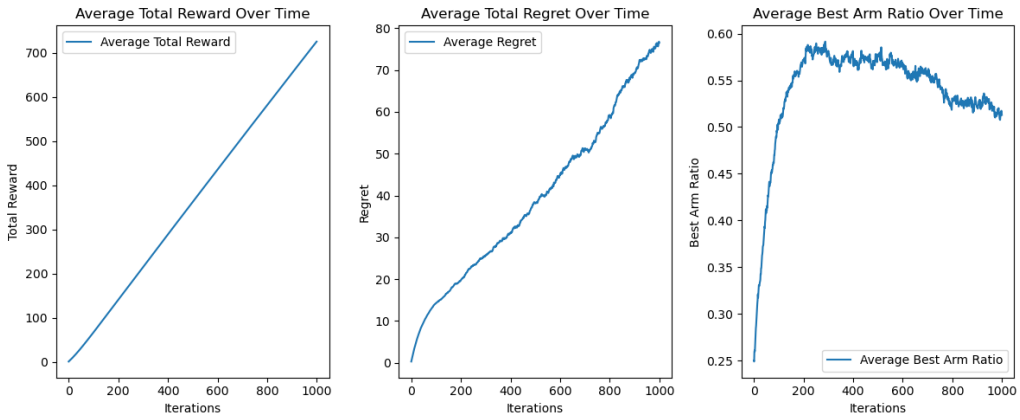

Abbildung 6: $\epsilon = 0.1$; 1000 Iterationen pro Durchlauf; Durschnitt über 1000 Durchläufe; Anfangswahrscheinlichkeiten 0.5 für jeden Arm; Wahrscheinlichkeiten werden nach jeder Iteration aktualisiert durch addieren eines $N(0, 0.01)$-verteilten Wertes

parallel steigen, was darauf hindeutet, dass der ϵ-greedy Algorithmus nicht in der Lage ist, dauerhaft eine Optimale Strategie zu finden, da die Reward-Wahrscheinlichkeiten kontinuierlich variieren.

(d)

Eine Möglichkeit ist die Implementation einer Vergessensrate α, durch welche weiter zurückliegende Berechnungen von $Q_t(a)$ niedriger gewertet werden, wodurch folglich relevantere aktuelle Rewards bevorzugt werden. Dies geht auf die Annahme zurück, dass ältere Werte deutlichem Wechsel unterliegen haben können und demnach nicht mehr repräsentativ sind. Analog dazu könnte ebenso einfach ein Mittelwert der letzten i Iterationen ausgewertet werden.

Außerdem kann ϵ abhängig von der Stärke der Veränderung der Rewards erhöht oder gesenkt werden. Wenn plötzliche stark abweichende Rewards notiert werden, kann dementsprechend die Erkundungsrate erhöht werden um eine möglicherweise neue optimale Action zu finden.

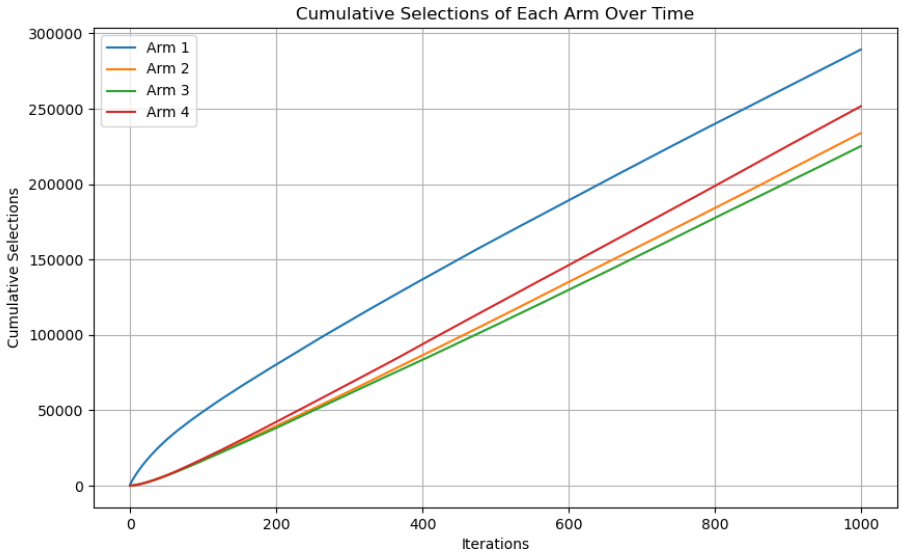

Abbildung 7: Verteilung der gewählten Aktionen: $\epsilon = 0.1$; 1000 Iterationen pro Durchlauf; Durschnitt über 1000 Durchläufe; Anfangswahrscheinlichkeiten 0.5 für jeden Arm; Wahrscheinlichkeiten werden nach jeder Iteration aktualisiert durch addieren eines $N(0, 0.01)$-verteilten Wertes