Seminar - Online Learning

Bandits

Department of Statistics
Ludwig-Maximilians-Universität München

Vincent Evers

Munich, January 31th, 2026

Submitted in partial fulfillment of the requirements for the degree of B. Sc.
Supervised by Tobias Brock

Abstract

In this report we look at multi-armed bandits through the idea of online mirror descent with estimated gradients. By choosing negative entropy as the regularizer and using the softmax link function, we get the exponentiated gradient algorithm. We then look at its regret bound and show that, with the right learning rate, the algorithm achieves sublinear regret. Finally a small example is included to show how the weights change over time and how the regret behaves in practice.

Contents

1 Introduction

Sequential decision making under uncertainty appears in many real-world applications. A common setting is that a learner needs to repeatedly choose between several actions. The learner only gets feedback on his choice in form of a cost or a reward. The learner does not observe how the other actions would have performed. Thus, the learner needs to choose actions based on limited feedback with the aim to minimize his cumulative cost. This is captured in the Multi-armed Bandit Problem, which was first formally studied by Herbert Robbins in 1952[1].

A common performance measure for these types of problems is regret. Here, we compare the cumulative cost of our algorithm to the cumulative cost of a comparator. This comparator is usually the best fixed decision in hindsight. The notion of a lower regret bound and regret as a performance measure was first introduced by Tze Lung Lai and Herbert Robbins in 1985[2]. In 2002 Auer et al.[3] considered the adversarial setting for the first time, where nothing is assumed about the data generating process of the costs. They also presented the EXP3 algorithm and derived a regret bound for it. Prior to this only stochastic bandits were studied, where the costs of every choice have an underlying fixed iid distribution. This was revolutionary, since it allows us to consider the worst case scenario. Later other authors presented extensions to the EXP3 algorithm, for example, the EXP3++ algorithm presented by Yevgeny Seldin and Aleksandrs Slivkins in 2014[4], which is more optimistic than EXP3. This algorithm can handle stochastic and adversarial regimes, which is useful, because the worst case scenario is often too pessimistic in practice and thus misses out on performance.

The topic of this report is the Multi-armed Bandit Problem with bounded costs, a quick introduction into different types of bandits and the EXP3 algorithm to handle adversarial bandits. We derive the algorithm starting from Online Mirror Descent with estimated Gradient and show how this leads to a regret bound. In the end we also look at a short example.

This report is structured as follows. Section 2 introduces the general Multi-armed Bandit Problem. We then talk about different types of Bandits in Section 3 and a related topic Learning with Expert Advice in Section 4. In Section 5 we start deriving the EXP3 algorithm, by introducing Online Mirror descent with estimated Gradient. We also establish a regret bound for OMD. By choosing the softmax function for link function we arrive at the EXP3 algorithm in Section 6. We derive the regret bound for the EXP3 algorithm and show that it is sublinear. To visualize we look at a small example in Section 7. Finally Section 8 summarizes the main conclusions.

[1] Herbert Robbins. "Some aspects of the sequential design of experiments". In: (1952).

[2] Tze Leung Lai and Herbert Robbins. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[3] Peter Auer et al. "The nonstochastic multiarmed bandit problem". In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

[4] Yevgeny Seldin and Aleksandrs Slivkins. "One practical algorithm for both stochastic and adversarial bandits". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1287–1295.

2 The Multi-armed Bandit Problem

To introduce the Multi-armed Bandit algorithm we imagine d-different Bandits (Slot-Machines), where we will play a total of T rounds. Each round t we pull the arm p_t of one of the bandits and and receive a cost vector $\mathbf{y}_t \in [0,1]^d$. We only get to see $y_t[p_t]$, the cost of the arm we choose. Our goal will be to have the lowest possible cumulative cost. We randomly choose an arm based on our weights $\mathbf{w}_t \in S$, such that $\mathbb{P}[p_t = i] = w_t[i]$, where $S = \{\mathbf{w} : ||\mathbf{w}||_1 = 1 \wedge \mathbf{w} \geq 0\}$ is a probability simplex. Every round we can calculate our loss with $f_t(\mathbf{w}) = \langle \mathbf{w}, \mathbf{y}_t \rangle$ and $\nabla f_t(\mathbf{w}) = \mathbf{y}_t$. Since f is linear, getting the gradient is very straightforward. In each round we try to choose a combination of weights, which minimizes f.

We will measure the performance of our algorithm with regret, where our comparator is the best arm in hindsight. The best arm in hindsight is the arm with the lowest cumulative cost. The best arm in hindsight is of course not known in practice.

Regret

$$\underbrace{\mathbb{E}\left[\sum_{t=1}^{T} y_t[p_t]\right]}_{\substack{\text{cumulative cost}\\ \text{of the algorithm}}} - \underbrace{\min_i \sum_{t=1}^{T} y_t[i]}_{\substack{\text{cumulative cost of}\\ \text{the best arm in hindsight}}}$$

The expected value is being taken with respect to the randomness in choosing an arm.

We need the Gradient $\mathbf{y}_t$ to optimize our algorithm, but only $y_t[p_t]$ is known. That is why we take an unbiased estimate $\mathbf{z}_t$.

Unbiased gradient estimate

$\mathbf{z}_t$ is an unbiased estimate of $\mathbf{y}_t$ and defined as

$$z_t[j] = \begin{cases} \dfrac{y_t[j]}{w_t[j]} & \text{if } j = p_t, \\ 0 & \text{else.} \end{cases}$$

This makes it very clear how there is only information on the loss, based on the arm pulled. Since $\mathbf{z}_t$ is dependent on p_t, we can just write $\mathbf{z}^{(p_t)}$. We can easily show that $\mathbf{z}_t$ is unbiased.

$$\mathbb{E}[z_t[j]|\mathbf{z}_{t-1}, ..., \mathbf{z}_1] = \sum_{i=1}^{d} \mathbb{P}[p_t = i] z_t^{(i)}[j] = w_t[j]\frac{y_t[j]}{w_t[j]} = y_t[j]$$

$\mathbf{y}_t$ being the gradient of f_t at point $\mathbf{w}_t$, means it is the only subgradient at that point.

Subgradient

Let f be a convex function.
y is a sub-gradient of f at w if it satisfies the inequality

$$f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \mathbf{y} \rangle.$$

We will need this inequality to later derive a regret a bound for the Online Mirror Descent algorithm.[5]

3 Stochastic and Adversarial Bandits

In a stochastic bandit problem we assume a fixed iid distribution for the losses of every arm. This assumption can sometimes be difficult to verify, because the costs might suddenly change completely. The goal in stochastic bandits is to find the arm with the lowest cost and exclusively pull this arm. So it perfectly captures an Exploration vs Exploitation setting. Stochastic Bandits are used instead of A/B testing, for example when choosing ads for consumers. Granted we have to assume fixed consumer preferences in the time frame where our algorithm is used.
In an adversarial bandit setting we assume an adversary picked all of the costs. This adversary could either be oblivious or adaptive. An oblivious adversary picks all of the costs before the game even starts. On the other hand an adaptive adversary chooses the costs before every round, meaning he can react to our past decisions. When we are dealing with an adaptive adversary it is crucial to randomize our choice in the arms. If we would not do this our regret would be linear and we would have no chance of achieving sublinear regret. With adversarial bandits the goal is to perform well even in a worst case scenario. Applications here are auction bidding, spam detection or cybersecurity. The examples all have in common that the environment is able to react to our decisions.

4 Learning with Expert Advice

A related Problem to bandits is learning with expert advice. In every round we choose one out of d experts. Each of them receives a cost, but we get see every cost. This is the important difference between learning with expert advice and the multi-armed bandit problem, this would be like seeing the cost of every arm. In this Problem there is no exploration vs exploitation tradeoff and learning is much faster, since we have full information in every round. Just like with bandits the costs could be generated by a fixed iid distribution or an adversary. Example applications are portfolio selection or weather forecasting, in the sense that we can choose between different weather forecasting models. In both cases we choose a portfolio / weather model, but we know how other experts would have performed in the same time frame.

[5] Shai Shalev-Shwartz. "Online learning and online convex optimization". In: *Foundations and Trends® in Machine Learning* 4.2 (2025), pp. 107–194, p. 131.

5 Online Mirror Descent with estimated Gradient

In OMD we optimize a series of convex loss functions $f_1, f_2, ...$, using an unbiased subgradient estimate z_t.

Online Mirror Descent with Estimated Gradient

```
parameter: a link function $g : \mathbb{R}^d \to S$
initialize: $\boldsymbol{\theta}_1 = 0$
for $t = 1, 2, ...$
    predict $\mathbf{w}_t = g(\boldsymbol{\theta}_t)$
    pick $\mathbf{z}_t$ at random such that $\mathbb{E}[\mathbf{z}_t|\mathbf{z}_{t-1}, ..., \mathbf{z}_1] \in \partial f_t(\mathbf{w}_t)$
    update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$
```

In OMD we have a primal space S in which our $\mathbf{w}$ lives, but due to constraints on that primal space, we make our update steps in the dual space using $\boldsymbol{\theta}$. The link function g translates between the two and defines the architecture of our problem.[6]
We have following result for the regret bound of the general OMD with estimated gradient algorithm.

Theorem 1

If the sub-gradients are chosen such that with probability 1 we have

$$\sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle \leq B(\mathbf{u}) + \sum_{t=1}^{T} ||\mathbf{z}_t||_t^2$$

B is some function and $|| \cdot ||_t$ depends on $\mathbf{w}_t$. From this follows

$$\mathbb{E}\left[\sum_{t=1}^{T} (f_t(\mathbf{w}_t) - f_t(\mathbf{u}))\right] \leq B(\mathbf{u}) + \sum_{t=1}^{T} \mathbb{E}[||\mathbf{z}_t||_t^2]$$

The expected value is being taken with respect to the randomness of the learner in choosing the gradient $\mathbf{z}_1, ..., \mathbf{z}_T$.

In the context of bandits the randomness in choosing z_t is equivalent to the randomness in choosing an arm, since z_t depends on p_t. The function B depends on the architecture of our problem. The norm being dependent on $\mathbf{w}_t$ means our position in the primal space influences the way we measure distance.

Proof. We now proof Theorem 1.

$$\sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle \leq B(\mathbf{u}) + \sum_{t=1}^{T} ||\mathbf{z}_t||_t^2$$

We take the expected Value on both sides.

[6] Shalev-Shwartz, see n. 5, p. 178.

$$\mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\right] \le B(\mathbf{u}) + \sum_{t=1}^{T}\mathbb{E}[||\mathbf{z}_t||_t^2]$$

Let $\mathbf{v}_t = \mathbb{E}[\mathbf{z}_t|\mathbf{z}_{t-1}, ..., \mathbf{z}_1]$. Using the law of total probability we can bound the left side.

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\right] &= \sum_{t=1}^{T}\mathbb{E}[\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle] \\ &= \sum_{t=1}^{T}\mathbb{E}[\mathbb{E}[\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle \mid \mathbf{z}_{t-1}, \dots, \mathbf{z}_1]] \\ &= \sum_{t=1}^{T}\mathbb{E}[\langle \mathbf{w}_t - \mathbf{u}, \underbrace{\mathbb{E}[\mathbf{z}_t|\mathbf{z}_{t-1}, \dots, \mathbf{z}_1]}_{=\mathbf{v}_t}\rangle] \\ &= \mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{v}_t\rangle\right]\end{aligned}$$

Now we can use the subgradient property from earlier, since $\mathbf{v}_t = \mathbb{E}[\mathbf{z}_t|\mathbf{z}_{t-1}, ..., \mathbf{z}_1] \in \partial f_t(\mathbf{w}_t)$.

$$\begin{aligned} & f_t(\mathbf{u}) \ge f_t(\mathbf{w}_t) + \langle \mathbf{v}_t, \mathbf{u} - \mathbf{w}_t\rangle \\ \Leftrightarrow\ & \langle \mathbf{u} - \mathbf{w}_t, \mathbf{v}_t\rangle \le f_t(\mathbf{u}) - f_t(\mathbf{w}_t) \\ \Leftrightarrow\ & \langle \mathbf{w}_t - \mathbf{u}, \mathbf{v}_t\rangle \ge f_t(\mathbf{w}_t) - f_t(\mathbf{u})\end{aligned}$$

Which lets us conclude

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=1}^{T}\overbrace{\langle \mathbf{w}_t - \mathbf{u}, \mathbf{v}_t\rangle}^{\ge f_t(\mathbf{w}_t) - f_t(\mathbf{u})}\right] \le B(\mathbf{u}) + \sum_{t=1}^{T}\mathbb{E}[||\mathbf{z}_t||_t^2] \\ \Leftrightarrow\ & \mathbb{E}\left[\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{u}))\right] \le B(\mathbf{u}) + \sum_{t=1}^{T}\mathbb{E}[||\mathbf{z}_t||_t^2].\end{aligned}$$

This gives us the desired result.[7]

□

6 Algorithm and Regret Bound

Now we will use OMD with estimated gradient to derive a bandit algorithm. At first we let the primal space be the probability simplex, so $S = \{\mathbf{w} : ||\mathbf{w}||_1 = 1 \wedge \mathbf{w} \ge 0\}$. We choose the softmax function for g, such that the i-th element of g is

$$g_i(\boldsymbol{\theta}) = \frac{\exp(\eta\theta[i])}{\sum_j \exp(\eta\theta[j])}.$$

[7] Shalev-Shwartz, see n. 5, p. 178-179.

For $||\cdot||_t$ we choose the local norm induced by the regularizer R at point $\mathbf{w}_t$.[8]

$$||\mathbf{z}_t||_t^2 = \mathbf{z}_t^T(\nabla^2 R(\mathbf{w}_t))^{-1}\mathbf{z}_t$$

The negative entropy is our regularizer.[9]

$$R(\mathbf{w}_t) = \frac{1}{\eta}\sum_{i=1}^{d} w[i]\log(w[i])$$

The regularizer R is directly connected to the link function g.[10]

$$g(\boldsymbol{\theta}) = (\nabla R)^\star(\boldsymbol{\theta}) = (\nabla R)^{-1}(\boldsymbol{\theta}) = \mathbf{w}$$

Because the Regularizer R is a Legendre function[11], the fenchel conjugate $R^\star$ is equal to the inverse Regularizer R^{-1}.[12]

Now we can calculate the inverse hessian matrix of the regularizer.

$$\begin{aligned}
R(\mathbf{w}_t) &= \frac{1}{\eta}\sum_{i=1}^{d} w[i]\log(w[i]) \\
\nabla R(\mathbf{w}_t) &= \frac{1}{\eta}(1+\log(w[1]),...,1+\log(w[d])) \\
\nabla^2 R(\mathbf{w}_t) &= \frac{1}{\eta}\text{diag}(\frac{1}{w[1]},...,\frac{1}{w[d]}) \\
(\nabla^2 R(\mathbf{w}_t))^{-1} &= \eta\text{diag}(w[1],...,w[d])
\end{aligned}$$

Using the inverse hessian matrix we can write the norm as follows.

$$\begin{aligned}
||\mathbf{z}_t||_t^2 &= \mathbf{z}_t^T(\nabla^2 R(\mathbf{w}_t))^{-1}\mathbf{z}_t \\
&= \mathbf{z}_t^T\eta\text{diag}(w[1],...,w[d])\mathbf{z}_t \\
&= \eta\sum_{i=1}^{d} w_t[i]z_t[i]^2.
\end{aligned}$$

Using the Fenchel-Young inequality it can be shown that the regret bound is valid. For this proof it is not necessary that R is a Legendre function.[13]
This a special case of the general OMD algorithm, called the normalized exponentiated gradient algorithm. For the regret bound of this algorithm we have the following result.

[8] Francesco Orabona. "A modern introduction to online learning". In: *arXiv preprint arXiv:1912.13213* (2019), p. 60.

[9] Shalev-Shwartz, see n. 5, p. 144.

[10] Shalev-Shwartz, see n. 5, p. 150.

[11] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020, p. 311.

[12] Shalev-Shwartz, see n. 5, p. 152.

[13] Shalev-Shwartz, see n. 5, p. 149-152.

Theorem 2

For a sequence of linear loss functions such that for all t, i we have $\eta z_t[i] \geq -1$, then

$$\sum_{t=1}^{T} \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle \leq \frac{\log(d)}{\eta} + \eta \sum_{t=1}^{T} \sum_i w_t[i] z_t[i]^2$$

The condition of $\eta z_t[i] \geq -1$ always holds since $\eta > 0$ and

$$\underbrace{z_t[j]}_{\geq 0} = \begin{cases} \frac{\overbrace{y_t[j]}^{\in[0,1]}}{\underbrace{w_t[j]}_{>0}} & \text{if } j = p_t, \\ 0 & \text{else} \end{cases},$$

which leads to $\eta z_t[i] \geq 0$.

Using the normalized exponentiated gradient algorithm we have following bandit algorithm

Bandit Exponentiated Gradient Algorithm

```
parameters: $\eta \in (0, 1)$
initialize: $\mathbf{w}_1 = (\frac{1}{d}, ..., \frac{1}{d})$
for $t = 1, 2, ...$
    choose $p_t \sim \mathbf{w}_t$
    receive $y_t[p_t] \in [0, 1]$
    update
        $\tilde{w}[p_t] = w_t[p_t] \exp\left(\frac{-\eta y_t[p_t]}{w_t[p_t]}\right)$
        for $i \neq p_t, \tilde{w}[i] = w_t[i]$
        $\forall i, w_{t+1} = \frac{\tilde{w}[i]}{\sum_j \tilde{w}[j]}$
```

At first we choose a learning rate η, which determines our step size. We initialize a $\mathbf{w}_1$, such that every arm has the same starting probability. In every round we choose an arm based on $\mathbf{w}_t$ and receive a cost $y_t[p_t]$. Then we make our update step, where we initially only update the weight of the arm we choose. The weight of the other arms stays the same and only changes when we normalize to adjust for the updated weight. Notice how the update step only reduces the weight of the arm we chose, which after normalization means the weights of the other arms increases. The exception here is if we receive $y_t[p_t] = 0$, then the weights stay the same. We take bigger steps if we have a high cost $y_t[p_t]$ and a small weight $w_t[p_t]$.

$$\underbrace{w_t[p_t] \overbrace{\exp(\frac{-\eta y_t[p_t]}{w_t[p_t]})}^{\in(0,1]}}_{\leq w_t[p_t]}$$

Proof. To derive a regret bound for this algorithm we start by combining Theorem 1 and Theorem 2.

$$\mathbb{E}\left[\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{u}))\right] \leq \mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t\rangle\right] \leq \frac{\log(d)}{\eta} + \eta\sum_{t=1}^{T}\mathbb{E}\left[\sum_i w_t[i]z_t[i]^2\right]$$

In the last term we can bound the expected value by the number of arms.

$$\begin{aligned}\mathbb{E}\left[\sum_i w_t[i]z_t^{(p_t)}[i]^2|\mathbf{z}_{t-1},..,\mathbf{z}_1\right] &= \sum_j \mathbb{P}[p_t = j]\sum_i w_t[i]z_t^{(j)}[i]^2 \\ &= \sum_j w_t[j]w_t[j]\left(\frac{y_t[j]}{w_t[j]}\right)^2 \\ &= \sum_j \underbrace{y_t[j]^2}_{\leq 1,\text{ since } \mathbf{y}_t\in[0,1]^d} \\ &\leq d\end{aligned}$$

Let the comparator u be the decision to pull the best arm in hindsight.

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{u}))\right] &= \sum_{t=1}^{T}\underbrace{\mathbb{E}[f_t(\mathbf{w}_t)]}_{=\mathbb{E}[y_t[p_t]]} - \underbrace{\sum_{t=1}^{T} f_t(\mathbf{u})}_{\min_i \sum_{t=1}^{T} y_t[i]} \\ &= \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} y_t[p_t]\right] - \min_i \sum_{t=1}^{T} y_t[i]}_{\text{Regret}}\end{aligned}$$

From this we can conclude following regret bound.[14]

Regret Bound

$$\mathbb{E}\left[\sum_{t=1}^{T} y_t[p_t]\right] - \min_i \sum_{t=1}^{T} y_t[i] \leq \frac{\log(d)}{\eta} + \eta T d$$

□

If we know the number of arms d and the number of rounds T, we can find an optimal $\eta*$, which minimizes the upper regret bound. To do this we can take the derivative with respect to η

$$\frac{\log(d)}{\eta} + \eta T d\frac{\partial}{\partial\eta} = -\frac{\log(d)}{\eta^2} + Td$$

[14] Shalev-Shwartz, see n. 5, p. 181.

and set it equal to 0

$$-\frac{\log(d)}{\eta*^2} + Td = 0$$
$$\eta* = \sqrt{\frac{\log(d)}{dT}}.$$

If we plug $\eta*$ into the regret bound, we achieve sublinear regret.

$$\lim_{T\to\infty} \frac{\mathsf{Regret}_T}{T} = 0$$

So if we play an infinite number of rounds, on average we are just as good as the best arm in hindsight. In Landau Notation this can be written as $\mathsf{Regret}_T = O(\sqrt{d\log(d)T})$ and if $T > d$ asymptotically this is equal to $O(\sqrt{T})$.

7 Example Scenario

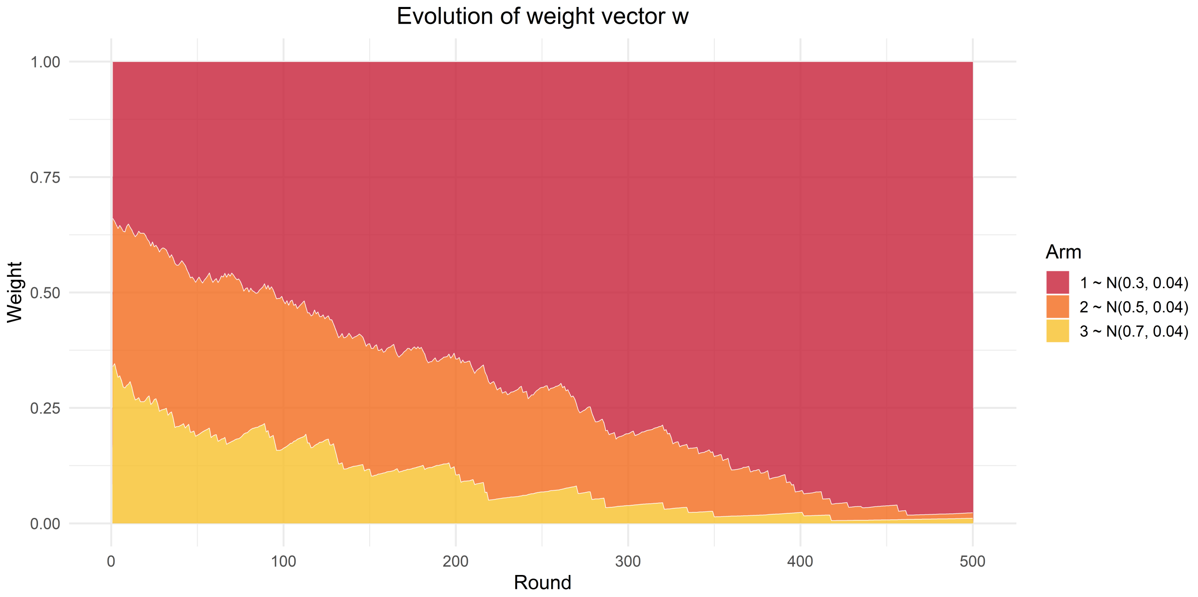
To illustrate the Algorithm and the regret bound we will go through a quick example.
We will generate data, with 500 rounds, 3 arms and a fixed distribution for the costs of every arm. We will bound the cost such that $\mathbf{y}_t \in [0,1]^d$.

$$\text{Arm 1} \sim N(0.3, 0.04),\ \text{Arm 2} \sim N(0.5, 0.04),\ \text{Arm 3} \sim N(0.7, 0.04)$$

We make no assumptions about how the costs were generated. If we would know that the costs were generated by stationary iid distributions there would be more suitable algorithms for this problem. For example the explore-then-commit algorithm mentioned in the presentation about stochastic bandits.

$$\eta* = \sqrt{\frac{\log(d)}{dT}} \approx 0.0271$$

In the following plots we will look at how $\mathbf{w}_t$ changes and how our observed regret compares to the regret if we only pulled one arm.



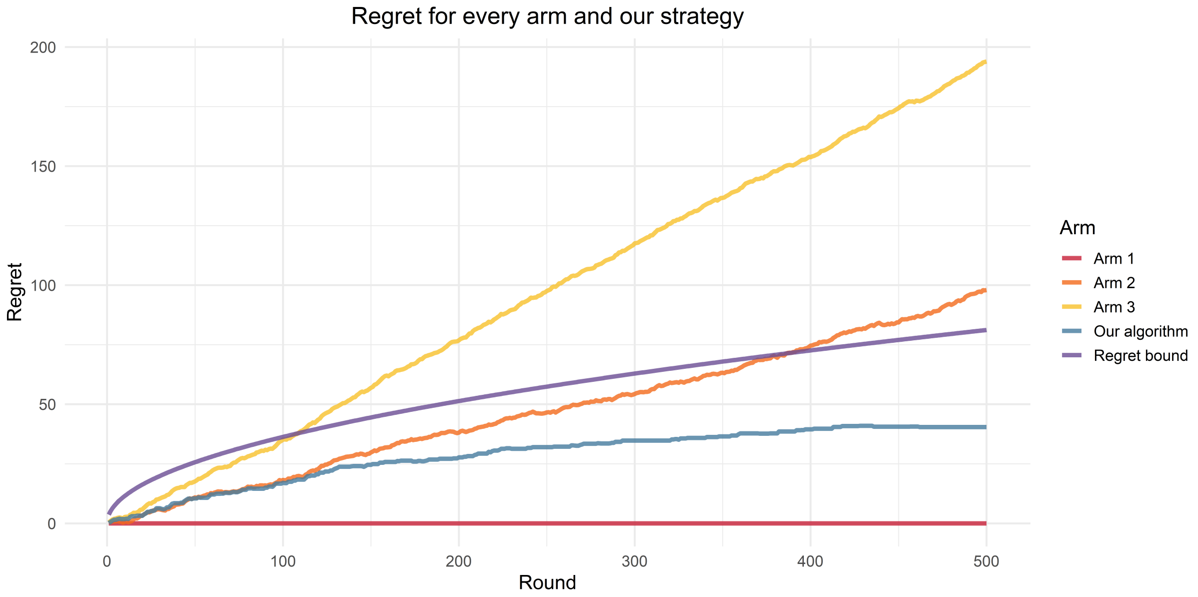
On the x-axis we see the rounds and the y-axis represents our weight vector $\mathbf{w}_t$. We can see that in the first round every arm has the same probability of being chosen, but as the rounds progress we almost exclusively pull arm 1. That makes sense, because it is clearly the best arm.

Notice how the probability of the other arms is never 0. This is an important feature of this algorithm and comes from the way we chose our link function g.

$$g_i(\boldsymbol{\theta}) = \frac{\overbrace{\exp(\eta\theta[i])}^{>0}}{\sum_j \exp(\eta\theta[j])}$$

The exponential function is always larger than 0.

This is important, because the algorithm is designed to handle adversarial bandits and once the probability of an arm would be 0 it will stay there and not change. An adversary could easily exploit that, by distributing costs accordingly.



The blue line represents our regret and the purple line shows the regret bound with an optimal η. We can clearly see that the regret is indeed sublinear. The large gap between the actual regret and its bound is explained by the fact that the costs are not hostile at all. The regret for arm 1 is always 0, because it is the best arm, which is our comparator.

8 Summary

The goal of bandit algorithms is to optimize sequential decision making under uncertainty.
According to the assumptions we have about how the costs were generated we choose an algorithm. The Algorithm presented in this report is called EXP3.
We measure algorithm performance with regret, where we compare the cumulative regret of our strategy to the best arm in hindsight.
When we are dealing with an adversary, it is crucial to randomize which arm we pick to achieve sublinear regret. Sublinear regret means that on average we are just as good as the best arm hindsight.

References

Auer, Peter et al. "The nonstochastic multiarmed bandit problem". In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

Lai, Tze Leung and Herbert Robbins. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

Lattimore, Tor and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Orabona, Francesco. "A modern introduction to online learning". In: *arXiv preprint arXiv:1912.13213* (2019).

Robbins, Herbert. "Some aspects of the sequential design of experiments". In: (1952).

Seldin, Yevgeny and Aleksandrs Slivkins. "One practical algorithm for both stochastic and adversarial bandits". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1287–1295.

Shalev-Shwartz, Shai. "Online learning and online convex optimization". In: *Foundations and Trends® in Machine Learning* 4.2 (2025), pp. 107–194.

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Location, date

Name