Heavy Tailed Bandit Portfolio Optimisation

Eric Patarin, Nils Reynaud

ML Research Seminar

March 29, 2024

Contents

Introduction

Typical Bandit paper

To summarize : a bandit algorithm is a sequence (policy) that indicates the arm that the agent should select at each timestep in order to minimize the total regret R_T and identify the best arm in hindsight.
Every bandit paper provides :

1. Context and Problem Setting (Linear Stochastic reward, Adversarial convex...)
2. an Algorithm (Online Mirror Descent, Follow the Leader, UCB...)
3. Regret Bound guarantees ($R_T = \tilde{\mathcal{O}}(d^{16}\sqrt{T})$ as in (Suggala21a))

Bandits - Regret

$$R_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \tag{1}$$

- This regret R_T is measured after T rounds.
- f_t is called the loss function at time t. In the adversarial setting it is chosen by an adversary.
- x_t is the arm selected at time t. Here it is a vector of the set $K \subset \mathcal{R}^d$, but keep in mind it can be a scalar or a matrix for example. It all depends on the problem setting.
- Also note that the definition of the regret depends on the problem setting, and that sometimes we prefer the "pseudo regret" or the "expected regret".

Bibliography for Bandits

Bibliography for Bandits	
Authors	Title
Shen et al.	Portfolio choices with orthogonal bandit learnin
Bubeck et al.	Bandits with heavy tail
Moeini et al.	Portfolio optimization by means of a $\chi-armed$ bandit algorithm
Huo et al.	Risk-aware multi-armed bandit problem with application to portfolio selection.
Zhu et al.	Adaptive portfolio by solving multi-armed bandit via thompson sampling
Ni,He et al.	Contextual combinatorial bandit on portfolio management
Dorn et al.	Fast UCB-type algorithms for stochastic bandits with heavy and super heavy symmetric noise
Kang et al.	Heavy-Tailed Linear Bandit with Huber Regression

References - Main inspiration

- Inspiration = Two papers that we seek to combine :
- $[OBP15]$ Shen, W., Wang, J., Jiang, Y. G., Zha, H. (2015, June). Portfolio choices with orthogonal bandit learning. In Twenty-fourth international joint conference on artificial intelligence.
- $[Bubeck13]$ Bubeck, S., Cesa-Bianchi, N., Lugosi, G. (2013). Bandits with heavy tail. IEEE Transactions on Information Theory, 59(11), 7711-7717.
- Goal is robustify $[OBP15]$ using Robust-UCB from $[Bubeck13]$

State of the Art - Bandit Portfolio Optimisation

Motivation

- The tradeoff between *exploration* and *exploitation* to maximize rewards in bandit algorithms naturally establishes a connection to portfolio choice problems
- Financial Returns can be considered Stochastic, Contextual, and Adversarial which all fit possible Bandit modelization

Open challenges - How to handle :

1. Correlation between assets ?
2. The heavy tailed nature of financial returns (Black Scholes underestimates risk) ?

Setting - OBP15

- Multi-Armed Bandit with n arms, n finite
- Stochastic Rewards
- The largest reward $\nu^* = \max_{i=1,\ldots,n} \nu_i$ and the maximum reward after m plays is $m\nu^*$. Thus, the pseudo regret after m plays is defined by

$$m\nu^* - \sum_{k=1}^{m} \mathbb{E}\left[r_{i_k}\left(t_k\right)\right]$$

where i_k is the index of selected arm at time t_k and $r_{i_k}\left(t_k\right)$ is the corresponding reward [Lai and Robbins, 1985]. We want to minimize this quantity using UCB.

Algorithm - OBP15

- Inputs: $m, n, l, \Delta t, \mathbf{R}_k, \tau$ for $k = 1 \to m$ do
- Estimate the average return $\mathbb{E}\left[\mathbf{R}_k\right]$ and covariance matrix of asset returns $\mathbf{\Sigma}_k$ by $\{\mathbf{R}_{-\tau+k}, \ldots, \mathbf{R}_{k-1}\}$;
- Implement the principal component decomposition as equation (5): $\mathbf{\Sigma}_k = \mathbf{H}_k \mathbf{\Lambda}_k \mathbf{H}_k^\top$;
- Compute the renormalized similarity matrices and eigenvalues (8): $\tilde{\mathbf{\Sigma}}_k = \tilde{\mathbf{H}}_k \mathbf{\Sigma}_k \tilde{\mathbf{H}}_k^\top = \tilde{\mathbf{\Lambda}}_k$;
- Compute the Sharpe ratio of each arm (10); Compute the adjusted reward function of each arm (11);
- Select the optimal arms according to **UCB** (11) from the first l and the next $n - l$ orthogonal portfolios, respectively;
- Compute the optimal mixture weight θ_k^* by (12); Compute the optimal portfolio weight $\boldsymbol{\omega}_k$ by (13);
- Output: The portfolio weight vectors $\boldsymbol{\omega}_k$ and the portfolio returns μ_k for $k = 1, \ldots, m$.

Portfolio Optimization χ-armed Bandit Algorithm

1. Moeini, M., Wendt, O., Krumrey, L. (2016). Portfolio optimization by means of a -armed bandit algorithm. In Intelligent Information and Database Systems: 8th Asian Conference, ACIIDS 2016, Da Nang, Vietnam, March 14–16, 2016, Proceedings, Part II 8 (pp. 620-629). Springer Berlin Heidelberg.
2. Based on Bubeck Munos Stoltz Szepesvàri 2011 χ-armed Bandits
3. χ-armed Bandits are a generalization of stochastic bandits with infinitely many arms, χ is a measurable space of arms
4. Use of Hierarchical Optimistic Optimization (HOO) algorithm on NP-Hard variants of the portfolio selection problem
5. Test algorithm on real dataset with a derivative free solution

Risk-Aware Portfolio Selection

1. Huo, X., Fu, F. (2017). Risk-aware multi-armed bandit problem with application to portfolio selection. Royal Society open science, 4(11), 171377.
2. Introduce filtering of assets based on the topological structure of the financial market
3. Add a risk measure into the Bandit Policy
4. Does not seem more interesting than $[OBP15]$ that uses Sharpe Ratio instead of mean return already

Adaptative Portfolio by Solving MAB via Thompson Sampling

- Zhu, M., Zheng, X., Wang, Y., Li, Y., Liang, Q. (2019). Adaptive portfolio by solving multi-armed bandit via thompson sampling. arXiv preprint arXiv:1911.05309.
- Also use MAB, Stochastic modelization, with Sharpe Ratio as objective
- Arms correspond to different strategies (Equal Weights, Mean Variance Markowitz, Orthogonal Bandit Portfolio from $[OBP15]$) and are pulled according to Thompson Sampling

Contextual Combinatorial bandit on portfolio management

- Two-stage investing strategy
- Supervised adaptative decision tree approach to build a pool of candidate portfolios
- Reinforcement LinUCB learning-based selection procedure
- Stocks withing a given portfolio have similar features, return and volatility
- Each sample stock is assigned a utility score based on past returns and associated volatility
- Definition of utility: mix of risk-aversion and risk-seeking

Contextual Combinatorial bandit on portfolio management

- Ni, He, et al. "Contextual combinatorial bandit on portfolio management." Expert Systems with Applications 221 (2023): 119677.
- Decision tree: use of CART
- Tree based on various economic indicators and the market performance of individual stocks
- Qualified portfolio candidates: end leaf nodes with low Gini values and high utility
- Use multi-CART to allow a parallel calculation and generate a number of portfolio candidates
- Bandits: expected payoff is replaced by utility
- Measure the performance of the investment: Sharpe ratio

Initialize:

$\hat{W}_k = W_k^*$ for each portfolio k ▷ For the first m periods

$k^* = \arg\max_k D_k(t) W_k(t)$

Main Loop:

while termination condition is not satisfied **do**

for each $k \in K$ **do**

$D_{k,t} = r_k(t) - \eta v_k(t)$ ▷ η depends on investor

$X_{k,t} = \sum_{i=t}^{t+n} r_k(i) - \eta v_k(i)$

$U_{k,t} = X_{k,t} \cup \hat{D_{k,t}}$

$\hat{W}_k(t+1) = (D_{k,t}^T \cdot D_{k,t} + I)^{-1} D_{k,t} U_{k,t}$ ▷ Update each candidate

end for

$k^* = \arg\max_k (X_{k,t}(\hat{W}_k(t+1) + \beta\sqrt{X_k(t)(D_{k,t}^T \cdot D_{k,t} + I)^{-1} X_k^T(t)}$

Record portfolio performance by candidate k^*

$t = t + 1$

end while

State of the Art - Heavy Tailed Bandits

Bubeck's Robust UCB (2013)

- Introduced by Bubeck et al. in Bandits With Heavy Tail (2013)
- Parameter: $\varepsilon \in (0, 1]$, mean estimator $\widehat{\mu}(t, \delta)$.
- For arm i, define $\widehat{\mu}_{i,s,t}$ as the estimate $\widehat{\mu}\left(s, t^{-2}\right)$ based on the first s observed values $X_{i,1}, \ldots, X_{i,s}$ of the rewards of arm i. Define the index

$$B_{i,s,t} = \widehat{\mu}_{i,s,t} + v^{1/(1+\varepsilon)} \left(\frac{c \log t^2}{s}\right)^{\varepsilon/(1+\varepsilon)},$$

  for $s, t \geq 1$ and $B_{i,0,t} = +\infty$.
- At time t, draw an arm maximizing $B_{i,T_i(t-1),t}$.
- In constrat $[OBP15]$ arms are drawn according to :
  $i_k^* = \underset{i=1,\ldots,n}{\arg\max}(\bar{r}_i\left(t_k\right) + \sqrt{\frac{2\ln(k+\tau)}{\tau+k_i}}),$

Robust UCB (2013)

- What if $[OBP15]$ returns are heavy tailed ? Will James-Stein shrinkage method estimator be enough ?$\rightarrow$ $[Bubeck13]$ Median of means, Catoni's estimator
- Theoretically valid for our Multi-Armed (Finite arms) Stochastic bandits setting when the rewards (e.g Sharpe Ratio hence Returns) are not sub-Gaussian.

Fast UCB-type algorithms for stochastic bandits with heavy and super heavy symmetric noise

- stochastic multi-armed bandit problem with a heavy-tailed reward distribution
- Robust UCB proposed in [Bubeck13]: general template for constructing UCB-type algorithms if one has an accessible and robust mean estimation procedure
- Vanilla UCB: empirical mean
- Robust UCB: truncated empirical mean, median of means, Catoni's M-estimator
- Here: Follows the same UCB template: idea is to construct K auxiliary optimization problems for reward estimation, one for each arm, and use the appropriate optimization method and divergence rate to construct the UCB-type index
- use of g(k, δ) bounded algorithms
- introduction of the Clipped-SGD-UCB algorithm where For each arm i we choose $fi(x) = 0.5 * (x - \mu_i)^2$

1. For each arm i $(i = 1, \ldots, K)$ construct supplementary convex optimization problem
$$\min_{x \in \mathbb{R}} \quad f_i(x),$$
such that $\mu_i = \arg\min_{x \in \mathbb{R}} f_i(x)$, with an accessible stochastic zero/first-order oracle and the corresponding $g(k, \delta)$-bounding algorithm A.
2. Use $\{x_k\}_{k=1}^{T}$ generated by the algorithm A and corresponding bound $g(k, \delta)$ to construct UCB-type estimation on the mean μ_i.
3. At each round play arm with the biggest UCB.

Fast UCB-type algorithms for stochastic bandits with heavy and super heavy symmetric noise

- Number of arms K, period T, batch-size $b = (2m+1)n$, initial estimates $x_1^0 = \cdots = x_K^0 = x^0$, clipping regime $\{\lambda_t\}_{t=1}^{\infty}$, learning rate schedule $\{\gamma_t\}_{t=1}^{\infty}$, parameter δ.
- Run Clipped-SGD-UCB for each arm i $(i = 1, \dots, K)$ independently for b iterations and compute $\nabla_{\epsilon_0} f_i(x_i^0)$ and $x_i^1 = x_i^0 - \gamma_0 \text{clip}(\nabla_{\epsilon_1}(x_i^0), \lambda_0)$.
- For each arm i $(i = 1, \dots, K)$, set $n_i(K) = 1$ and compute $UCB(i, n_i(K), \delta) = x_i^1 + \sqrt{g(1, \delta)}$.
- **for** $t = 1, \dots, T$ **do**
- Choose arm $i_t = \arg\max_{1 \le i \le K} UCB(i, n_i, \delta)$.
- Play i_t arm b times, observe rewards and compute $\nabla_{\epsilon_{n_{it}}} f_{i_t}(x_{i_t}^{n_{it}})$.
- Compute $x_{i_t}^{n_{it}+1} = x_{i_t}^{n_{it}} - \gamma_{n_{it}} \text{clip}(\nabla_{\epsilon_{n_{it}}} f_{i_t}(x_{i_t}^{n_{it}}), \lambda_{n_{it}})$.
- Set $n_{i_t}(t+1) = n_{i_t}(t) + 1$ (increase counter by one).
- **if** $i \neq i_t$ **then**, Set $UCB(i, n_i(t+1), \delta) = UCB(i, n_i(t), \delta)$.
- **else**, Set $UCB(i, n_i(t+1), \delta) = x_{i_t}^{n_{it}} + \sqrt{2g(n_i(t+1), \delta)}$.

Heavy-Tailed Linear Bandit with Huber Regression

- Relaxes the sub-Gaussian assumption on the reward error
- new algo robust to heavy-tailed errors
- use of the Huber estimator to estimate the reward model parameters
- estimates β_i with Huber regression
- With this estimator, the estimation error bound is proportional to the inverse of the minimum eigenvalue of the Gram matrix $->$ need to find this minimum bound

- Input: h, ν, α ; $\beta(\hat{T}_i, 0) = \beta(\hat{S}_i, 0) = 0^d$
- **for** $t \in [T]$ **do**
- Observe $X_t \sim P_X$
- **if** $t \in \hat{T}_i$ **then**, $a(t) = i$
- **else**
- $D = \{i \in [K] | \max_{j\in[K]} X_t^T \beta(\hat{T}_j, t-1) - X_t^T \beta(\hat{T}_i, t-1) \leq h\}$
- $a(t) = \arg\max_{i\in D} X_t^T \beta(\hat{S}_i, t-1)$
- **end if**
- Update $S(a(t), t) = S(a(t), t-1) \cup \{t\}$, Observe reward $y_t = X_t^T \beta_{a(t)} + \epsilon_{a(t),t}$
- **if** $t \in \hat{T}_i$ **then**
- $\tau(\hat{T}_i, t) = \nu(|\hat{T}_i, t| / \log(t^2(d+1)/\alpha))^{1/(1+\nu)}$
- $\beta(\hat{T}_i, t) = \arg\min_{\beta\in\mathbb{R}^d} \sum_{t'\in\hat{T}_i, t} |y_{t'} - X_{t'}^T \beta|^\nu$
- **else**
- $\tau(\hat{S}_i, t) = \nu(|\hat{S}_i, t| / \log(t^2(d+1)/\alpha))^{1/(1+\nu)}$
- $\beta(\hat{S}_i, t) = \arg\min_{\beta\in\mathbb{R}^d} \sum_{t'\in\hat{S}_i, t} |y_{t'} - X_{t'}^T \beta|^\nu$

Recap Bibliography for Bandits

Recap Biblio Bandits			
Title, Contexte	Cadre Bandit	Algo	Regret bound / Improvement
Portfolio choices with orthogonal bandit learning	Stochastic rewards	PCA of Portfolio, UCB	Add Robust UCB
Bandits with heavy tail	Stochastic, distributions have moments of order 1 + ϵ	Refined estimators of the mean	$R_n = \mathcal{O}(\frac{8v \log n}{\Delta_i})$
Portfolio optimization by means of a $\chi - armed$ bandit algorithm	Stochastic Bandits, infinitely many arms	Hierarchical Optimistic optimization algo	...

Recap Bibliography for Bandits

Recap Biblio Bandits			
Title	Bandit Setting	Algo	Improvements, Regret Bounds
Risk-aware multi-armed bandit problem with application to portfolio selection.	Stochastic rewards	Add a risk measure into the Bandit Policy	Optimal selection of parameters
Adaptive portfolio by solving multi-armed bandit via thompson sampling	Stochastic rewards	Each arm is a strategy	Correlation between arms
Contextual Combinatorial bandit on portfolio management	Linear returns	Decision Tree for candidates and Bandit selection	Make the 2 stages interact

Recap Bibliography for Bandits

Recap Biblio Bandits			
Title, Contexte	Cadre Bandit	Algo	Améliorations, Regret Bounds
Fast UCB-type algorithms for stochastic bandits with heavy and super heavy symmetric noise, Improves UCB mean estimator	Stochastic Rewards	K auxiliary optimization problems	$O(logT(TKlog(T))^{\frac{1}{2}})$
Heavy-Tailed Linear Bandit with Huber Regression	Heavy-tailed errors, Stochastic contexts	Huber estimator for reward model parameters	$O(dT^{\frac{1}{1+\delta}}(logdT)^{\frac{\delta}{1+\delta}})$

Experimentation

$[OBP15]$ experimentation under regular returns

We select a list of tickers (your PEA for example) a trading start date and a sliding window. We tune the l factor parameter and τ the sliding window and get :

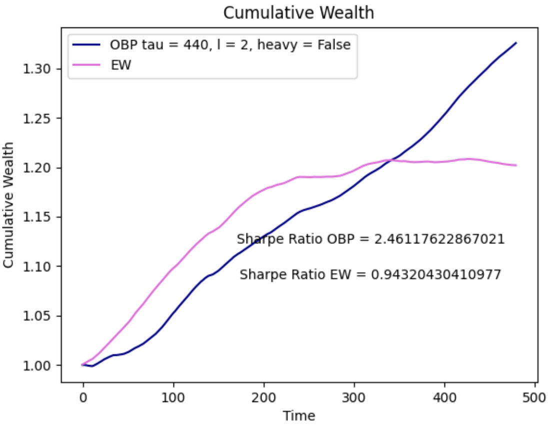

Figure: Notice the Sharpe Ratios

$[OBP15]$ experimentation under regular returns

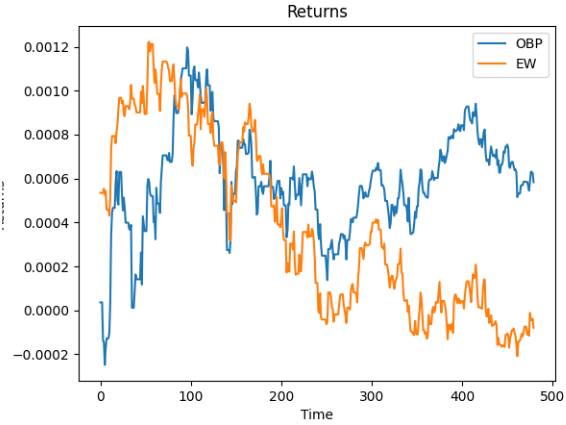

Figure: Gross returns per day

$[OBP15]$ experimentation under regular returns

Our weights show very few spikes (in comparison to what you will see next).

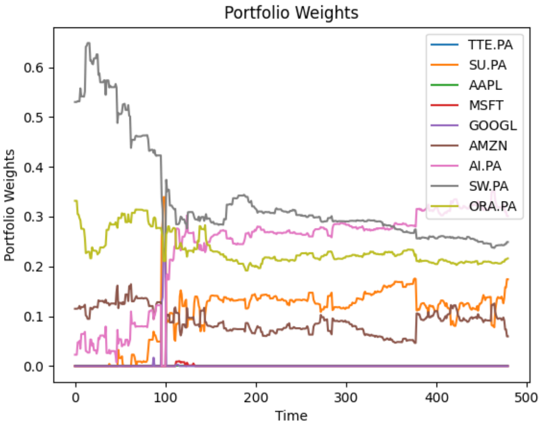

Figure: Evolution of asset weights over time

$[OBP15]$ experimentation under heavy tailed returns

We sample returns from a Student distribution, for $\tau = 120$ and $m = 480$. We perform Grid Search on parameters l and τ the sliding window.

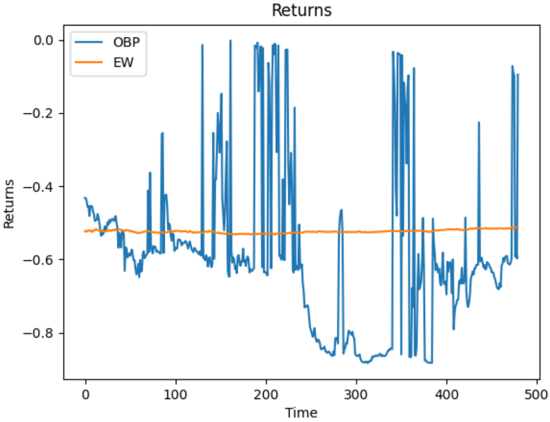

Figure: Returns are catastrophic !

$[OBP15]$ experimentation under heavy tailed returns

Our portfolio weights vary in a non continuous fashion. Unrealistic for big volumes without taking transaction costs into account.

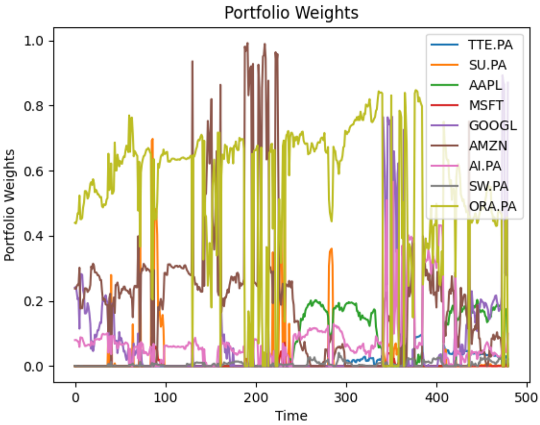

Figure: Weights between assets vary too much

Solution

- Solution is to robustify UCB $\rightarrow$ Robust UCB from $[Bubeck13]$
- For arm i, define $\widehat{\mu}_{i,s,t}$ as the estimate $\widehat{\mu}\left(s, t^{-2}\right)$ based on the first s observed values $X_{i,1}, \ldots, X_{i,s}$ of the rewards of arm i. Define the index

$$B_{i,s,t} = \widehat{\mu}_{i,s,t} + v^{1/(1+\varepsilon)} \left(\frac{c \log t^2}{s}\right)^{\varepsilon/(1+\varepsilon)},$$

  for $s, t \geq 1$ and $B_{i,0,t} = +\infty$.
- At time t, draw an arm maximizing $B_{i,T_i(t-1),t}$.
- In constrat $[OBP15]$ arms are drawn according to :
  $i_k^* = \underset{i=1,\ldots,n}{\arg\max} \bar{r}_i\left(t_k\right) + \sqrt{\frac{2\ln(k+\tau)}{\tau+k_i}}$,

Adaptation required : the Robust UCB does not depend on τ which is key in OBP (when estimating v for example) ! Which subdataset should you use ?

Problem encountered

When we do not adapt Robust UCB to the OBP algorithm (mainly τ) we tend to mostly get plots like this (when a super heavy left tail is added) :

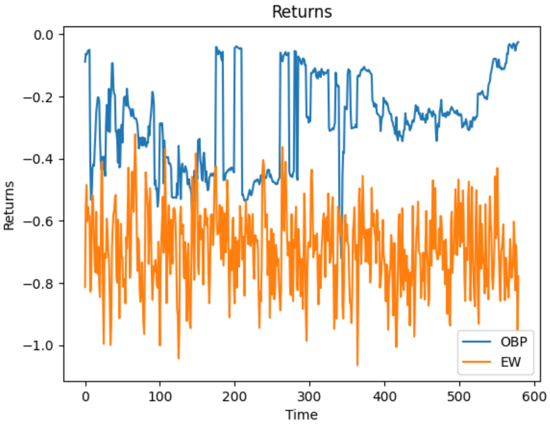

Figure: OBP is better but still negative

Key ideas for improvements

Heavy Tail

1. Use Median of returns in Sharpe Ratio instead of Mean, or any variant : RWM (Robust Weighted Median, in terms of Volume traded), Catoni, MoM...
2. Truncate values in the return distributions or add an ML technique for outlier detection
3. Use Clipped Robust UCB from Dorn, Yuriy, et al. "Fast UCB-type algorithms for stochastic bandits with heavy and super heavy symmetric noise."(2024). or even Huber Loss

Correlation

1. Add a measure of correlation into the objective function (can be rank method e.g Kendall for robustness, nice couple with Robust UCB)
2. Use Eigendecomposition / PCA ($[OBP15]$)