SOLVING NON-STATIONARY BANDIT PROBLEMS WITH AN RNN AND AN ENERGY MINIMIZATION LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider a Multi-Armed Bandit problem in which the rewards are non-stationary and are dependent on past actions and potentially on past contexts. At the heart of our method, we employ a recurrent neural network, which models these sequences. In order to balance between exploration and exploitation, we present an energy minimization term that prevents the neural network from becoming too confident in support of a certain action. This term provably limits the gap between the maximal and minimal probabilities assigned by the network. In a diverse set of experiments, we demonstrate that our method is at least as effective as methods suggested to solve the sub-problem of Rotting Bandits, can solve intuitive extensions of various benchmark problems, and is effective in a real-world recommendation system scenario.

1 INTRODUCTION

The cliché "insanity is doing the same thing over and over again and expecting different results" is obviously wrong. Our experience tells us that we cannot exploit the same action repeatedly and expect to enjoy the same outcome.

In this work, we are concerned with Multi-Armed Bandits (MAB). In the conventional MAB setting, the reward distribution of each arm is assumed to be stationary. Motivated by real-world scenarios, such as online advertising and content recommendation, Levine et al. (2017) have considered the Rotting Bandits setting, in which the reward decays in accordance with the number of times that an arm has been pulled. Rotting Bandits, however, do not address the cases in which the reward is dependent on the complete history of the arm pulling actions, which also takes into account the pulling of other arms, as well as the order of the actions.

A simple scenario, in which the reward for a given arm depends on the timing of past events and on the pulling of other arms is the intuitive scenario in which after k consecutive pulls of the same arm, its reward vanishes. In this scenario, with the pulling of any other arm, the exhausted arm is reset.

In our model, we employ a Recurrent Neural Network (RNN) in order to model the non-stationary reward distributions. For problems in which the decision at each round can benefit from a set of observations, this context can be put as the input to the RNN. Thus, our method naturally extends a non-stationary form of Contextual Bandits (Langford & Zhang, 2007). In our model, unlike the conventional Contextual Bandit case, the optimal action predictions are thus conditioned not only on the most recent context, but also on all previous contexts.

The learned policy selects the action based on a softmax that is applied to a set of logits. Similar to what is observed in other contexts of Reinforcement Learning (Tijsma et al., 2016) and also in Continual Learning (Serra et al., 2018), this may lead to an overconfident network. Such a confident decision increases exploitation at the expense of exploration. In order to overcome this, we add a novel regularization term that reduces the Boltzmann energy of the logits. This term is shown to directly lead to an upper bound on the ratio between the maximal probability assigned by the model to an arm and the minimal probability.

Our experiments are performed along four axes. First, we show that the new method performs well on well-known stationary problems. Second, that it outperforms multiple leading baselines on non-stationary extensions of these problems. Third, we show that our generic method outperforms,

in terms of convergence time, the leading methods for Rotting Bandits. Lastly, we show that our method is appropriate for capturing real-world scenarios. In a set of ablation experiments, we also demonstrate that the energy regularization method we advocate for, outperforms other exploration enhancing alternatives.

2 Related Work

Multi-Arm Bandits The MAB field started with the seminal work of Thompson (1933). Gittins (1974) were the first to propose a scenario where an arm's reward may change. Non-stationary arises when considering time-dependent priors(Besbes et al., 2014) or when only a subset of the bandit arms is available at a given time such as in the Sleeping Bandit(Kanade et al., 2009; Li et al., 2019). However, it also appears when there is no explicit time-dependency. For instance, when the reward of a given arm decays with each pull such as in the Rotting Bandit(Levine et al., 2017; Seznec et al., 2019) or when the rewards are received in delay(Liu et al., 2019).

Contextual Bandits(Li et al., 2010) aim to relate between a given context and the bandit problem. It is a highly applicable method, which was developed to support the personalization of the bandit algorithm per user. The method has been successfully applied for recommendation tasks(Li et al., 2010) and non-linear tasks(Krishnamurthy et al., 2016).

Exploration Strategies In Neural Networks The nature of the Multi-Arm Bandit problem requires the balancing between exploration and exploitation. Several methods have been proposed for the task. The ϵ-greedy method, performs exploration with probability ϵ. Tijsma et al. (2016) have shown that a better alternative is to apply softmax regularization, which smooths the probability distribution using a high temperature. Modifying the temperature of the softmax is also beneficial in other contexts, such as knowledge distillation(Hinton et al., 2015).

Recurrent Neural Networks RNNs have been the architecture of choice, when solving sequential tasks. Many of the successful RNN methods utilize a gating mechanism, where the hidden state h_t can be either suppressed or scaled, depending on a function of the previous hidden state and the input. Among these solutions, there is the Long-Short Term Memory network (LSTM) of Hochreiter & Schmidhuber (1997) that utilizes a gating mechanism together with a memory cell and the powerful Gated Recurrent Unit (GRU) by Cho et al. (2014). The GRU updates at each time a hidden state h_t given an input x_t,

$$h_t = (1 - f_t) \odot h_{t-1} + f_t \odot \tanh\left(W_1 x_t + W_2 h_{t-1} + b_h\right) \tag{1}$$

$$f_t = \sigma\left(W_{1,f} x_t + W_{2,f} h_{t-1} + b_f\right) , \tag{2}$$

where $W_1, W_2, W_{1,f}, W_{2,f}$ are weight matrices and b_f and b_h are biases.

3 Method

Let A be a set of n possible actions, $\{a_i\}_{i=1}^{n}$. At each time step $1 \leq t \leq T$, a context, $c_t \in \mathbb{R}^m$, is presented to the observer and a reward, $r_{i,t}$, is assigned to each action. This framework applies both to cases in which the context is meaningful $m > 0$ and in cases in which the context is void ($m = 0$).

In the bandit setting, only the reward $r_{i,t}$ associated with the action selected at time t is presented to the observer. Let $\{a_{i_t}\}_{t=1}^{T}$ be the set of actions selected by the observer. The objective of the learner is to minimize the regret,

$$R = \sum_t \max_k \{r_{k,t}\} - r_{i_t,t} \, . \tag{3}$$

In order to select an action at time step t, a l-layer stacked GRU(Cho et al., 2014), f, with a time-dependent hidden state, h_t, of size d is employed. The GRU receives as input the previous hidden state, h_{t-1}, together with the context vector, c_t, and updates the hidden state

$$h_t = f\left(c_t, h_{t-1}\right) . \tag{4}$$

Next, an affine layer, o, acts on the hidden state, h_t,

$$z_i = o\left(h_t\right) , \tag{5}$$

and is fed to a softmax layer, which provides the probability of selecting action a_i as

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \,. \tag{6}$$

Under this setting, only partial information is available, as only the reward for action i_t is revealed. In order to minimize the non-differentiable regret R (equation 3), we employ the REINFORCE(Williams, 1992) algorithm. The loss function at time t is

$$\mathcal{L}_R = -\left(r_{i_t,t} - \bar{r}_t\right) \log p_{i_t} \,, \tag{7}$$

where $\bar{r}_t$ is the mean of the obtained rewards up to time t.

Since the observer is unaware of the outcomes of other actions at time t, selecting the next action requires a subtle balance between exploration and exploitation. Whereas in neural networks, exploitation occurs naturally, we encourage exploration in three manners. First, we apply a dropout with probability $p_{dropout}$ after each GRU layer, thus adding uncertainty to the predicted actions. Second, we sample the next action by considering the probability, p_i, of the action and not by choosing the one with maximal p_i. Third, we add an energy conservation term to the loss function, thus regulating the values of the network's parameters, so the output of classification layer cannot become biased.

The softmax function can be interpreted as the Boltzmann's distribution with so that each neuron z_i carries an energy of $E_i = -z_i$. The mean energy of the system, $\langle E \rangle$, is

$$\langle E \rangle = \sum_{i=1}^{n} p_i E_i \,. \tag{8}$$

In order to conserve energy, the following regularization term is added to the loss function,

$$\mathcal{L}_{EC} = \langle E \rangle^2 = \left(\sum_{i=1}^{n} p_{i_t} E_i \right)^2 = \left(\sum_{i=1}^{n} p_{i_t} z_i \right)^2 \,. \tag{9}$$

Since $\mathcal{L}_{EC}$ is quadratic, it has a minimum only when

$$\langle E \rangle = 0 \,. \tag{10}$$

This ensures that if an activation, z_i, is drifting to a high value (which results in a high probability, p_i, for selecting action a_i), $\mathcal{L}_{EC}$ will lead to a compensation by increasing the values of the other neurons z_j as well, such that equation 10 is satisfied. Specifically, the following bound holds.

Theorem 1. *Let z_i be a set of n logits that are passed to a softmax to obtain probabilities p_i and are sufficiently regularized by $\mathcal{L}_{EC}$. Denote by $p_{\max}$ and $p_{\min}$ the maximal and minimal values of p_i. Then*

$$\frac{p_{\max}}{p_{\min}} \leq \frac{e^{W\left(\frac{n-1}{e}\right)}}{e^{W\left(-\frac{1}{e}\right)}} \,, \tag{11}$$

where W is the Lambert-W function defined as

$$W\left(x\right) e^{W(x)} = x \,. \tag{12}$$

Proof. Denote by $z_{\max}$ and $z_{\min}$ the maximal and minimal z_i values and by $i_{\max}$ the index of $z_{\max}$. It follows from $\sum_{i=1}^{n} p_{i_t} z_i = 0$ that

$$e^{z_{\max}} z_{\max} = -\sum_{i \neq i_{\max}} e^{z_i} z_i \leq -\left(n-1\right) e^{z_{\min}} z_{\min} \,. \tag{13}$$

The minimum of the function $y = xe^x$ is located at $y = -\frac{1}{e}$ which means $e^{z_{\min}} z_{\min} \geq -\frac{1}{e}$. Plugging this to equation 13,

$$e^{z_{\max}} z_{\max} \leq \frac{n-1}{e} \tag{14}$$

Since $z_{\max}$ must be positive to satisfy equation 10, and because xe^x is monotonic for $x > 0$,

$$z_{\max} \leq W\left(\frac{n-1}{e}\right) \tag{15}$$

Algorithm 1 The Recurrent Bandit Algorithm

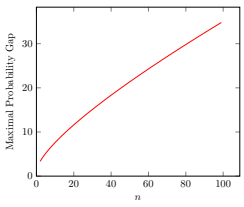
Input: α_{EC}
  $h_0 = 0$
  for $t = 1$ to T **do**
    $h_t = f\left(c_t, h_{t-1}\right)$
    $p_i = \text{SOFTMAX}(o(h_t))$ ▷ Compute probability for all actions p_i, Eq. 6
    Select the next action, a_{i_t} by sampling the multinomial distribution p_i
    Observe the reward $r_{i_t,t}$
    $\mathcal{L} = \mathcal{L}_R + \alpha_{EC}\mathcal{L}_{EC}$ ▷ Compute the loss terms in Eq. 7, 9
    Perform back-propagation and set $\mathcal{L} = 0$
  end for

The ratio between the probabilities $p_{\max}$ and $p_{\min}$ is

$$\frac{p_{\max}}{p_{\min}} = \frac{e^{z_{\max}}}{e^{z_{\min}}} \leq \frac{e^{W\left(\frac{n-1}{e}\right)}}{e^{W\left(-\frac{1}{e}\right)}} \,. \tag{16}$$

□

For small values of n, the bound grows almost linearly (see right). It limits the highest probability gap between the arms, thus allowing less visited actions to be explored.



The two loss terms are combined to the final term $\mathcal{L} = \mathcal{L}_R + \alpha_{EC}\mathcal{L}_{EC}$, where α_{EC} is a regularization hyperparameter. The entire procedure procedure follows Alg. 1.

4 EXPERIMENTS

We apply our method on a variety of different multi-arm bandit benchmarks. For all the tasks, except for the Yahoo! Recommendation Prediction Dataset and the Wheel Bandit, the context vector, c_t, that is fed to the Recurrent Bandit is null. Thus, information from previous time may only flow through the hidden state, h_{t-1}.

We optimize our method using RMSprop(Bengio, 2015) with a learning rate of 0.001. The network f for all the experiments is a 2-layer stacked GRU, each with a hidden size of $d_h = 128$. The dropout used for all the experiments is $p_{\text{dropout}} = 0.1$, except for the Rotting Bandits, where it is set to $p_{\text{dropout}} = 0$, since it does not require much exploration due to its nature. The regularization parameter is fixed at a value of $\alpha_{EC} = 0.1$ for all the benchmarks.

An ablation study is performed to verify the benefits of using the novel loss term $\mathcal{L}_{EC}$. ϵ-greedy ablation uses the same GRU architecture, however at each time step chooses a random action with a probability of $\epsilon = 0.01$, and the most probable action with probability $1 - \epsilon$. The Softmax-reg ablation uses Softmax regularization to smooth the action selection using a temperature of $T = 2$. The no EC ablation is obtained by setting $\alpha_{EC} = 0$. The Bernoulli Bandit baseline is taken from Algorithm 2 in (Chapelle & Li, 2011).

Bernoulli Multi-Arm Bandit The most basic MAB scenario is the Bernoulli one. A set of n handles or actions, are assigned uniformly with a prior $\{\hat{\theta}_k\}_{k=1}^{n}$. At each time step, the user has to pick one action. each of the actions carries a reward, $r_k \in \{0, 1\}$, sampled from the Bernoulli's distribution,

$$r_k \sim Bernoulli\left(\hat{\theta}_k\right) \,. \tag{17}$$

We compare our method to the Bernoulli Bandit (Chapelle & Li, 2011), which is based on Thompson-Sampling, to SWA(Levine et al., 2017) that was designed for the Rotting Bandit problem, and to the three ablation variants of our method. Fig. 1(a) shows the mean accumulated regret for 100 simulations for each of the methods. As can be seen, all methods are able to reduce the mean accumulated regret over time. However, SWA takes longer to accomplish this. Our method works best in this case if regularization is turned off, and its regret over the Bernoulli Bandit is not large.

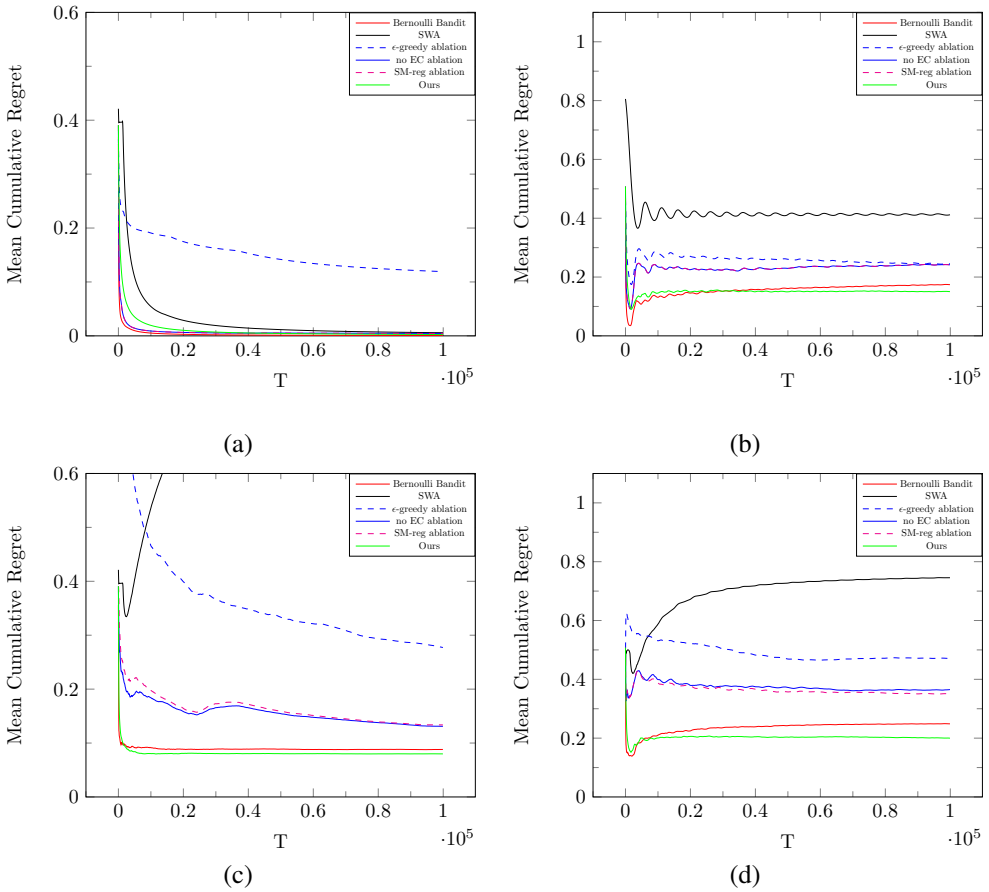


Figure 1: (a) A MAB with 10 actions. (b) A sinusidial MAB with 10 actions. (c) A MAB with 10 actions. We further limit the number of consecutive pulls of a handle by 10. (d) Sinusidial MAB with 10 actions. We further limit the number of consecutive pulls of a handle by 10.

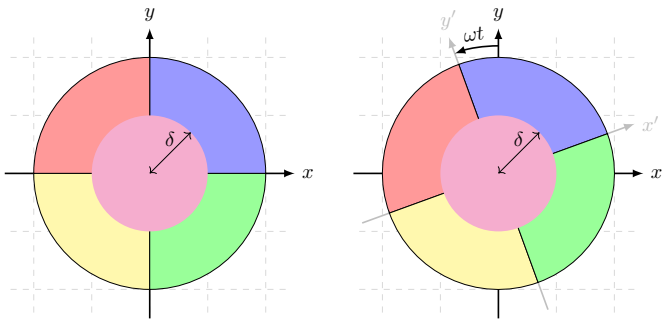
Time Dependent Multi-Arm Bandit One way to create a non-stationary bandit problem, is to add time-dependent priors to the Bernoulli Multi-Arm Bandit task. To accomplish this, time dependency is inserted to the priors $\hat{\theta}_k$ using the mirrored sine function:

$$\theta_k\left(t\right) = \hat{\theta}_k \left|\sin\left(\omega t + \phi_k\right)\right| \tag{18}$$

where $\omega = \frac{2\pi}{T_{\text{period}}}$. A phase, $\phi_k = \frac{2\pi k}{n}$, between the arms, is also added such that the arm holding the highest reward is changes place through time. For this experiment, there are $n = 10$ available arms and a periodicity of $T_{\text{period}} = 10000$. The mean accumulated reward is presented in Fig. 1(b). As can be seen, the drifting of the rewards over time makes it hard for both the Bernoulli Bandit, and the baselines without the Energy Conservation to keep a low regret. The SWA algorithm fails to solve this task, since it was designed specifically for decaying rewards. The variant of our methods that are meant to evaluate other forms of encouraging exploration also fail (this is also true for all other values we tested for their regularization parameter).

Time Dependent Multi-Arm Bandit with Correlative Arms In order to further investigate the non-stationary behavior, we limit the number of consecutive selections of a single arm to 10. After an arm has been selected for 10 times in a row, its reward is set to 0 unless another arm has been selected. The results of applying this setting to the Bernoulli Multi-Arm Bandit problem are shown in Fig. 1(c). In this case, only the Bernoulli Bandit and our method learn fast and keep a low regret. Under this setting, SWA is not able to solve this task.

We then combine the two MAB variants and apply zero the reward after ten consecutive pulls for the Time Dependent MAB. The results are shown in Fig. 1(d). Our method is the only one to maintain



(a) (b)

Figure 2: The Wheel Bandit setup. Each colored region describes a different reward distribution based on points sampled in that region. (a) The Wheel Bandit. (b) The Rotating Wheel Bandit with an angular velocity of ω.

a low mean cumulative regret. The Bernoulli Bandit regret increases with time, and the ablation variants are not able to converge to low solution.

Wheel Bandit The Wheel Bandit by Riquelme et al. (2018) has been introduced to examine the expressiveness of contextual bandits in non-linear tasks. The context $c_t = (x_t, y_t) \in \mathbb{R}^2$ in this task is uniformly sampled in the unit circle ($R = 1$). There are five available actions, a_i. The first, always grants a reward, $r \sim \mathcal{N}(\mu_1, \sigma)$ independent of the context, c_t. Inside the region $d = \sqrt{x^2 + y^2} \leq \delta$, the other four actions grant a reward of $r \sim \mathcal{N}(\mu_2, \sigma)$, for $\mu_2 < \mu_1$. For the region $d > \delta$, depending on the signs on x and y (to which quarter of $\mathbb{R}^2$ they belong to) one of the four arms grants a reward of $r \sim \mathcal{N}(\mu_3, \sigma)$, where $\mu_3 \gg \mu_1$, whereas the remaining arms grant a reward of $r \sim \mathcal{N}(\mu_2, \sigma)$. We use the same settings as the original Wheel bandit,

$$\mu_1 = 1.2 \quad \mu_2 = 1.0 \quad \mu_3 = 50 \quad \sigma = 0.01$$

with an exploration motivating parameter $\delta = 0.5$.

To make this problem non-stationary, a time-dependent rotation is applied. This is accomplished by making the actions' reward distributions depend on the sign of x' and y',

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} . \tag{19}$$

Both setups are depicted in Fig. 2. We run both setups for $T = 10000$, and for the Rotating Wheel Bandit, we use an $\omega = \frac{2\pi}{2000}$ (Total of five rotations). Riquelme et al. (2018) report that the best algorithm to solve the stationary task is their NeuralLinear algorithm, as can also be seen in Fig. 3(a) (our method is second best in this setting). However, when time-dependency is introduced, it quickly diverges, whereas our method is still able to learn the dynamics, as can be seen in Fig. 3(b).

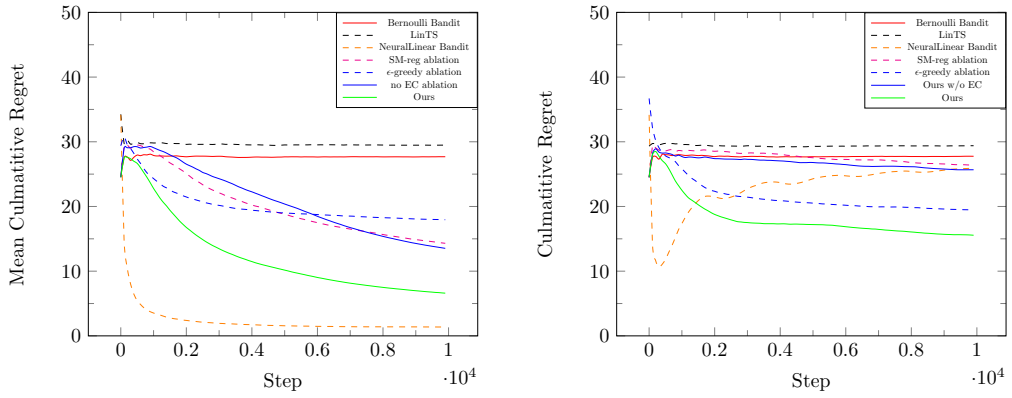
Rotting Bandit The Rotting Bandit framework was introduced by Levine et al. (2017) to handle cases where the Bandit's arms are not stationary, but rather decay over time with each pull. We follow the original scenario, by considering two available actions, a_1 and a_2 for 30000 time steps, with the following rewards, r_i, sampled from,

$$r_1 \sim \mathcal{N}\left(\mu_1 = 0.5, \sigma^2 = 0.2\right) \tag{20}$$

$$r_2 \sim \begin{cases} \mathcal{N}\left(\mu_2 = 1, \sigma^2 = 0.2\right) & n_2 < 7500 \\ \mathcal{N}\left(\mu_2 = 0.4, \sigma^2 = 0.2\right) & else \end{cases} , \tag{21}$$

where n_i is the number of times action a_i has been selected. For this experiment, the regret presented is the Policy Regret (Arora et al., 2012), since the optimal policy (Levine et al., 2017) for this task is,

$$a_{i_t} = \arg\max_i \left\{ r_i \left(N_{i,t} + 1 \right) \right\} , \tag{22}$$



(a) (b)

Figure 3: (a) The Wheel Bandit. (b) Rotating Wheel Bandit with $T_{\text{period}} = 2000$. LinTS is by Cortes (2018) and NeuralLinear is the method of Riquelme et al. (2018).

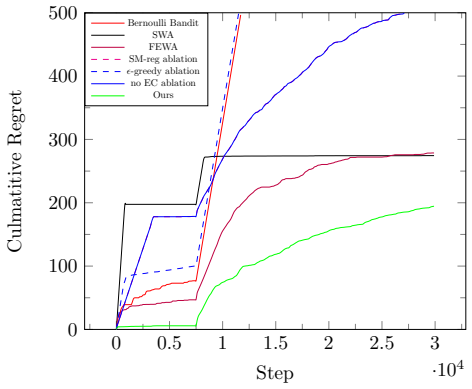


Figure 4: Cumulative regret for Rotting Bandits.

Table 1: Click Through Rate (CTR) for the Yahoo! Front Page Today Module User Click Log Dataset for different contextual bandits.

Method	**CTR**
LinUCB(Li et al., 2010)	0.040
LinTS(Cortes, 2018)	0.036
BootstrappedUCB(Cortes, 2018)	0.044
Softmax-reg ablation	0.090
ϵ-greedy ablation	0.052
No energy constraint ablation	0.090
Ours	0.167

where $N_{i,t}$ is the number of times action i was selected up to time t, and $r_i\ (n_i)$ is the reward given by action i after it was selected for n_i times.

We compare our method to SWA(Levine et al., 2017), FEWA(Seznec et al., 2019) (two Rotting Bandit methods), and to the Vanilla Bernoulli Bandit. The cumulative policy regret obtained by these methods averaged for 10 simulations is shown in Fig. 4. As can be seen, the Bernoulli Bandit and the GRU baselines fail to solve this task. Our method is able to converge much faster than SWA and FEWA. In contrast with SWA and FEWA, our method does not assume anything about the reward distribution, and therefore has to continuously explore. SWA and FEWA were designed to solve tasks with decaying rewards, and therefore do not require further exploration after a certain point. This is the reason our method keeps on accumulating regret, unlike SWA and FEWA that converge to the exact solution.

Yahoo! Recommendation Prediction Dataset The Yahoo! Front Page Today Module User Click Log Dataset (v2.0) by Li et al. (2010) is a real-life multi-arm bandit benchmark that allows the offline evaluation of different algorithms. The entries in this dataset are sorted by the time in which they occurred. Each entry is composed of (i) an article that was randomly presented to a user (ii) the user's response, whether the article was selected or not (iii) an anonymized 136-dimensional vector descriptor of the user (iv) a list of the possible articles that could have been presented at that time.

In order to evaluate the different approaches, we have taken 1.5M samples from the original dataset, together with 100K samples for evaluation. At each time step, all algorithms are fed with a binary

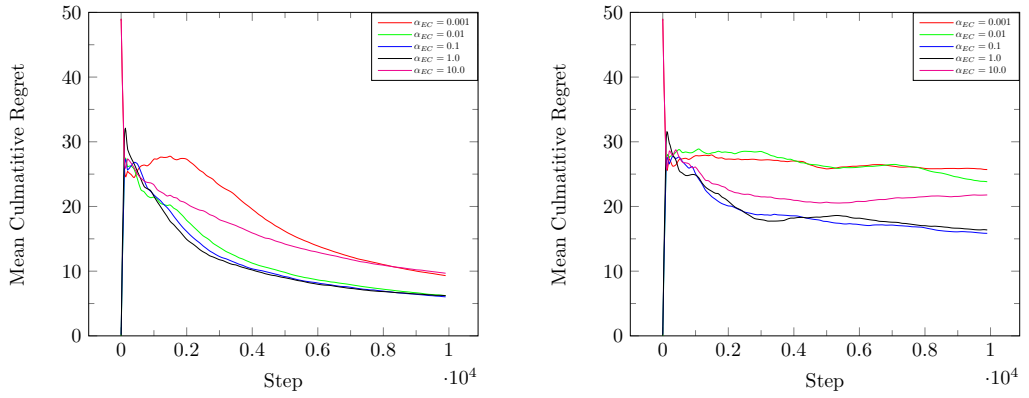


Figure 5: The Mean Cumulative Regret on the Wheel Dataset for different values of α_{EC}. (a) The stationary wheel. (b) The rotating wheel with $T_{\text{period}} = 2000$.

user descriptor vector as the context, $c_t \in \{0,1\}^1$ 36, and are tasked with predicting which article should be presented to that user. If the article was indeed presented to that user and the user clicked on the article, the algorithm is granted a reward of $r_t = 1$.

Since the (counterfactual) information on the user's response to other articles in the pool is absent, the evaluation of the various methods on this dataset is not trivial. To remedy this, we use the offline policy evaluation of Li et al. (2011). Only events in which a method has agreed with the originally displayed article are evaluated. A comparison is made to several algorithms available under the *contextualbandits* package(Cortes, 2018). LinUCB(Li et al., 2010), LinTS(Cortes, 2018), and BootstrappedUCB(Cortes, 2018) together with a comparison to the ablation variants of our method.

We report the click-through-rate (CTR), i.e., the ratio of users' selected articles to the number of articles presented by the algorithm in Tab. 1. As can be seen, our method greatly outperforms all other contextual bandits. In addition, the alternative means for encouraging exploration are outperformed by $\mathcal{L}_{EC}$ that our method uses.

Sensitivity to the regularization parameter In order to test the sensitivity of our method to different values of α_{EC}, We run our method on the Wheel Bandit and the Rotating Wheel Bandit tasks using different α_{EC} values. The results for this sensitivity test appear in Fig. 5. As can be seen, the performance is stable in the range [0.01,1] for the stationary version and in the somewhat smaller range of [0.1,1] for the rotating wheel.

5 Conclusions

The rewards obtained for a specific action are very often time dependent. In many cases, they are also dependent on the previous actions that were taken. In this work, we propose to employ an RNN in order to solve the non-stationary bandit problem. The solution is general enough to solve both the vanilla case, as well as the contextual case, and is robust enough to address stationary cases.

We note that training without proper regularization results in a learner who is too confident and neglects exploration. Such an exploration is especially crucial when the rewards vary over time. We, therefore, suggest a new regularization term that minimizes the Boltzmann energy. This term leads to a bounded gap between the maximal and minimal probability assigned to each arm. Our experiments show that our method addresses multiple non-stationary rewards that vary according to time only or also by action. In addition, our method successfully tackles the specific case of non-stationary bandits called the rotting bandits. The advantage of our method is especially clear on a real world recommendation dataset, where it obtains more than triple the click through rate of the literature baselines.

REFERENCES

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. *arXiv preprint arXiv:1206.6400*, 2012.

Yoshua Bengio. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. *corr abs/1502.04390*, 2015.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems*, pp. 199–207, 2014.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

David Cortes. Adapting multi-armed bandits policies to contextual bandits scenarios. *arXiv preprint arXiv:1811.04383*, 2018.

John Gittins. A dynamic allocation index for the sequential design of experiments. *Progress in statistics*, pp. 241–266, 1974.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

Varun Kanade, H Brendan McMahan, and Brent Bryan. Sleeping experts and bandits with stochastic action availability and adversarial rewards. *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, 2009.

Akshay Krishnamurthy, Alekh Agarwal, and Miro Dudik. Contextual semibandits via supervised learning oracles. In *Advances In Neural Information Processing Systems*, pp. 2388–2396, 2016.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 817–824. Citeseer, 2007.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in neural information processing systems*, pp. 3074–3083, 2017.

Fengjiao Li, Jia Liu, and Bo Ji. Combinatorial sleeping bandits with fairness constraints. *IEEE Transactions on Network Science and Engineering*, 2019.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 297–306, 2011.

Larkin Liu, Richard Downe, and Joshua Reid. Multi-armed bandit strategies for non-stationary reward distributions and delayed feedback processes. *arXiv preprint arXiv:1902.08593*, 2019.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pp. 4548–4557, 2018.

Julien Seznec, Andrea Locatelli, Alexandra Carpentier, Alessandro Lazaric, and Michal Valko. Rotting bandits are no harder than stochastic ones. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2564–2572, 2019.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Arryon D Tijsma, Madalina M Drugan, and Marco A Wiering. Comparing exploration strategies for q-learning in random stochastic mazes. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8. IEEE, 2016.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.