Adaptive Representation Selection in Contextual Bandit

Baihan Lin\textsuperscript{1}, Guillermo A. Cecchi\textsuperscript{2}, Djallel Bouneffouf\textsuperscript{2}, Irina Rish\textsuperscript{2}
\textsuperscript{1} Columbia University, New York, NY, USA
\textsuperscript{2} IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA
Baihan.Lin@columbia.edu, {gcecchi, dbouneffouf, rish}@us.ibm.com

Abstract

We consider an extension of the contextual bandit setting, motivated by several practical applications, where an unlabeled history of contexts can become available for pre-training before the online decision-making begins. We propose an approach for improving the performance of contextual bandit in such setting, via adaptive, dynamic representation learning, which combines offline pre-training on unlabeled history of contexts with online selection and modification of embedding functions. Our experiments on a variety of datasets and in different nonstationary environments demonstrate clear advantages of our approach over the standard contextual bandit.

1 Introduction

Sequential decision making is a common problem in many practical applications where the agent must choose the best action to perform at each iteration in order to maximize the cumulative reward over some period of time. One of the key challenges is achieving a good trade-off between the exploration of new actions and the exploitation of known actions. This exploration vs. exploitation trade-off in sequential decision making problems is often formulated as the multi-armed bandit (MAB) problem: given a set of bandit “arms” (actions), each associated with a fixed but unknown reward probability distribution [Lai and Robbins, 1985; Auer et al., 2002a], an agent selects an arm to play at each iteration, and receives a reward, drawn according to the selected arm’s distribution, independently from the previous actions.

A particularly useful version of MAB is the contextual multi-armed bandit (CMAB), or simply the contextual bandit problem, where at each iteration, before choosing an arm, the agent observes an N-dimensional context, or feature vector. Over time, the goal is to learn the relationship between the context vectors and the rewards, in order to make better prediction which action to choose given the context [Agrawal and Goyal, 2013].

For example, the contextual bandit approach is commonly used in various practical sequential decision problems with side information (context), from clinical trials [Villar et al., 2015] to recommender system [Mary et al., 2015], where the patient’s information (medical history, etc.) or an online user’s profile provide a context for making a better decision about the treatment to propose or an ad to show, and the reward represents the outcome of the selected action, such as, for example, success or failure of a particular treatment option.

Having an access to unlabeled data makes it possible to pre-train some model of the input (contexts) in an offline mode, and use it later to improve the online decision making. For example, we can learn an autoencoder to map the raw inputs into potentially better representations. Moreover, when the inputs are non-homogeneous, we may want to cluster the unlabeled data and learn separate representations for each cluster. Then, in the online mode, we can decide which representation to use for a given context; such context-driven representation selection has a potential to further improve the subsequent decision-making. These representation models (e.g., autoencoders) can (and should) continue to be updated online as more contexts become available, especially in nonstationary environments abundant in practical applications, where both the context distribution and the reward distribution can change in various ways.

Motivated by the above scenarios, we consider here a contextual bandit setting, called Contextual Bandit with Representation learning and unlabeled History (CBRH). In this setting, it is assumed that (1) a set of unlabeled contexts is available for pre-training before the online decision-making starts, (2) the bandit’s performance can be improved by learning a good context representation (embedding) rather than using the raw input, the (3) embedding functions are pre-trained on the unlabeled history and adaptively selected (and updated) based on the context during the online decision-making. Next, we propose an algorithm for the above CBRH setting, called Adaptive Representation Selection in Contextual Bandit.
Bandit with Context-Driven Embeddings (ABaCoDE), which implements online, clustering-based embedding selection and learning coupled with Thompson-Sampling contextual bandit approach.

As demonstrated by empirical results on multiple datasets, our approach consistently (and quite considerably) outperforms the standard contextual bandit, and appears to be particularly beneficial in nonstationary environments of several types, involving both context and reward nonstationarity.

2 Related Work

The multi-armed bandit problem has been extensively studied. Optimal solutions have been provided using a stochastic formulation [Lai and Robbins 1985; Auer et al. 2002a], a Bayesian formulation [Thompson 1933; Bouneffouf and Féraud 2016; Agrawal and Goyal 2012], or using an adversarial formulation [Auer and Cesa-Bianchi 1998; Auer et al. 2002b]. However, these approaches do not take into account the observation of transformation of these rewards through a stochastic multi-armed bandit (MAB) problem in which the learner interacts with the context which may affect to the arm’s performance. In LINUCB [Li et al. 2010] and Pre-Training [Agrawal and Goyal 2012], or using an adversarial formulation Thompson-Sampling contextual bandit algorithms [Auer and Cesa-Bianchi 1998; Auer et al. 2002b], the authors assume a linear dependency between the expected reward of an action and its context: the representation space is modeled using a set of linear predictors. This assumption is not used in Neural Bandit [Allesiardo et al. 2014]. Authors in [Bartók et al. 2014] studies considering some kind of incomplete feedback called "Partial Monitoring (PM)", which is a general framework for sequential decision making problems with incomplete feedback that allows the learner, when it is possible, to retrieve the expected value of actions through an analysis of the feedback matrix, both of which are assumed to be known to the learner.

In [Gajane et al. 2016] authors study a variant of the stochastic multi-armed bandit (MAB) problem in which the rewards are corrupted. In this framework, motivated by privacy preserving in online recommender systems, the goal is to maximize the sum of the (unobserved) rewards, based on the observation of transformation of these rewards through a stochastic corruption process with known parameters.

We can say that our setting is similar to the on-line semi-supervised learning [Yver 2009; Ororbia et al. 2015], which is a field of machine learning that studies learning from both labeled and unlabeled examples in an online setting. However in the their setting they receive the true label at each iteration, and we receive a bandit feedback.

3 Background

This section introduces some background concepts our approach builds upon, such as contextual bandit and Thompson Sampling.

The contextual bandit problem

Following [Langford and Zhang 2008], this problem is defined as follows. At each time point (iteration) \( t \in \{1, \ldots, T\} \), an agent is presented with a context (feature vector) \( x_t \in \mathbb{R}^N \) before choosing an arm \( k \in A = \{1, \ldots, K\} \). We will denote by \( X = \{X_1, \ldots, X_N\} \) the set of features (variables) defining the context. Let \( r_t = (r_{t1}, \ldots, r_{tK}) \) denote a reward vector, where \( r_{tk} \in [0, 1] \) is a reward at time \( t \) associated with the arm \( k \in A \). Herein, we will primarily focus on the Bernoulli bandit with binary reward, i.e. \( r_{tk} \in \{0, 1\} \). Let \( \pi : X \to A \) denote a policy.

Contextual Thompson Sampling

In this setting, we consider the general Thompson Sampling, where the reward \( r_{tk} \) for choosing arm \( i \) at time \( t \) follows a parametric likelihood function \( Pr(r_{tk} | \mu_i) \). Following [Agrawal and Goyal 2013], the posterior distribution at time \( t+1 \), \( Pr(\mu_i | r_{tk}) \propto Pr(r_{tk} | \mu_i)Pr(\mu_i) \) is given by a multivariate Gaussian distribution \( N(\mu_i(t + 1), \nu^2B_i(t + 1)^{-1}) \), where \( B_i(t) = I_d + \sum_{\tau=1}^{t-1} x_\tau x_\tau^\top \), and where \( d \) is the size of the context vectors \( x_t \), \( \nu = R \sqrt{\frac{24}{d \ln(\frac{1}{\gamma})}} \) with \( R > 0 \), \( \epsilon \in [0, 1] \), \( \gamma \in [0, 1] \) constants, and \( \mu_i(t) = B_i(t)^{-1}(\sum_{\tau=1}^{t-1} x_\tau r_{\tau}) \).

At every step \( t \), the algorithm generates a 1-dimensional sample \( \hat{\mu}_i \) from \( N(\mu_i(t), \nu^2B_i(t)^{-1}) \), for each arm, selects the arm \( i \) that maximizes \( x_i^\top \hat{\mu}_i \), and obtains reward \( r_{ti} \).

Algorithm 1: The Contextual Thompson Sampling Algorithm

1: Initialize: for \( i = 1, \ldots, k \), \( B_i = I_d, \mu_i = 0_d, f_i = 0_d \).
2: for \( t = 1, 2, \ldots, T \) do
3:   Receive context \( x_t \)
4:   for \( i = 1, \ldots, k \), sample \( \hat{\mu}_i \) from the \( N(\mu_i, \nu^2B_i^{-1}) \)
5:   Choose arm \( i_t = \arg \max_{i \in I} x(t)^\top \hat{\mu}_i \)
6:   Receive reward \( r_{ti} \)
7:   \( B_i = B_i + x_i x_i^\top, f_i = f_i + r_i \hat{\mu}_i, \mu_i = B_i^{-1} f_i \)
8: end

4 Problem Formulation

Using the notation introduced in the previous section, we now define our novel bandit setting: Contextual Bandit with Representation learning and unlabeled History (CBRH) (outlined in Alg. 2), based on the following key assumptions.

Firstly, we assume that a context \( x_t \in \mathbb{R}^N \) is mapped into its representation \( z_t \in \mathbb{R}^N \), using an embedding function \( e_i(x_t) \), selected from a set \( E = \{e_1, \ldots, e_c\} \) of currently available embedding functions. Second, we assume that the set of embedding functions \( E \) can be modified online. And third, an access to a set \( D \) of unlabeled contexts, i.e. contexts without the associated rewards, is assumed. This dataset can be used, for example, for pre-training embedding functions \( e(x) \). We then define a set \( \Pi = \cup_{e_i \in E} \{\pi : \mathbb{R}^N \to A, \pi(e) = \pi(e_i(e))\} \) of compound-function policies, where the function \( \pi : \mathbb{R}^N \to A \) maps \( z_t = e_i(e_t) \) to an action in \( A \). The objective is to learn a hypothesis \( \pi \) over \( T \) iterations maximizing the cumulative reward.
We now describe an adaptive, context-driven embedding selection approach to solving the CBRU problem introduced in the previous section. It has two variants, based on online- and offline clustering, respectively: the choice is controlled by a Boolean input parameter isOnline in Algorithm 3. Two more inputs include: an unlabeled pre-training dataset D, as well as the number of embeddings k. The algorithm processes the input contexts sequentially, one by one, but at the end of each mini-batch of data it updates the embeddings to reflect possible changes in the data distribution.

Algorithm 2 The CBRH Problem Setting

1: Obtain unlabeled set of contexts D
2: Learn a context representation model
3: Repeat
4: $(x_t, r_t)$ is drawn according to distribution $D_{c,r}$
5: A context representation $z_t$ is obtained
6: The player chooses an arm $k_t = \hat{\pi}(z_t)$
7: The reward $r_t$ is revealed
8: The player updates its policy $\pi$
9: $t = t + 1$
10: Until $t = T$

5 Adaptive Bandit with Context-Dependent Embeddings (ABaCoDE)

We now describe an adaptive, context-driven embedding selection approach to solving the CBRU problem introduced in the previous section. It has two variants, based on online- and offline clustering, respectively; the choice is controlled by a Boolean input parameter isOnline in Algorithm 3. Two more inputs include: an unlabeled pre-training dataset D, as well as the number of embeddings k. The algorithm processes the input contexts sequentially, one by one, but at the end of each mini-batch of data it updates the embeddings to reflect possible changes in the data distribution.

Algorithm 3 Adaptive Bandit with Context-Dependent Embeddings (ABaCoDE)

1: Input: unlabeled dataset D, a set of unlabeled contexts for pre-training: k, the number of clusters (and corresponding embeddings); a Boolean variable isOnline.
2: Initialization:
3: Cluster D into k clusters: $C = \{c_1, \ldots, c_k\}$
4: For each cluster, train an autoencoder to construct a set of encoding functions (embeddings): $E = e_1, \ldots, e_k$
5: Initialize the contextual Thompson Sampling parameters of bandit $B$ (line 1 in Alg. 1).
6: while there is a next data mini-batch $M$, do
7: foreach $x_t$ from $M$ do
8: $c_t = \text{assignCluster}(x_t)$
9: if isOnline then $\text{updateCluster}(C, x_t, c_j)$
10: $e = \text{selectEmbedding}(c_j)$
11: $z = e(x_t)$ (encoded context/representation)
12: contextualBandit$(B, z)$ (lines 4-7 in Alg. 1)
end
13: if not(isOnline) then $\text{recomputeClusters}(C, B)$
14: $\text{updateEmbedding}(M, C)$
end

The initialization step consists of clustering the pre-training dataset D into k clusters (line 3), training an autoencoder for each cluster, which results into k encoding (embedding) functions (line 4), and initializing parameters of the contextual Thompson Sampling bandit, used later to make classification decisions based on embedded context (line 5).

Next, the algorithm switches to the online mode, processing an online stream of incoming samples (contexts). As mentioned above, we assume that at the end of each fixed-length time window, i.e. a fixed-size mini-batch of data, we update our embeddings.

Within each data mini-batch $M$ (line 7), once the next input sample $x_t$ arrives, it is first assigned to one of the existing clusters $c_j$ (line 8), associated with the corresponding embedding function $e_j$. Next, an online clustering is performed if isOnline is true, i.e. the centroid of the cluster $c_j$ is recomputed, but no changes are made to other clusters (line 9). Otherwise, there are no changes to clusters, until the end of the batch, as we will see shortly. Based on the cluster assignment $c_j$, the corresponding embedding function $e_j$ is used to compute the representation vector $z$ for given input $x_t$ (line 10); given the context $z$, the contextual bandit $B$ makes a decision (line 11), obtains the reward $r_t$ (line 12), and updates its parameters (line 13) using the contextual Thompson Sampling described in the previous section.

After the end of the mini-batch $M$ is reached (line 14), if isOnline was false, the clusters will be recomputed from scratch using all data points received so far (however, no such re-clustering is performed if the online clustering was selected). Finally, the embeddings (i.e., their corresponding autoencoder parameters) are updated respectively using the updated set of clusters $C$.

In the next section, we present empirical results comparing both online and offline clustering methods outlined above with two baseline approaches:

- **Contextual Bandit (CB):** as the baseline, we use the standard contextual multi-armed bandit with Thompson Sampling, based on the raw input (i.e., no embeddings).
- **universal embedding (uE):** a universal embedding denotes a single embedding computed based on all data, and always recomputed to include the data from the most recent mini-batch; no clustering is performed.
- **mini-batch embedding (mE):** this is our offline clustering approach presented in Algorithm 3, when isOnline is false.
- **online embedding (oE):** this is the online version of our algorithm described above, i.e. isOnline is true.

6 Empirical Evaluation

6.1 Datasets

We evaluated our approach on four imaging datasets: MNIST [LeCun, 1998], STL-10 [Coates and Ng, 2011], CIFAR-10 [Coates et al., 2011], Caltech-101 Silhouettes-28 [Griffin et al., 2007] and Warfarin [Consortium and others, 2009] (for details of each dataset, see Table 1). To simulate an online data stream, we draw samples from each dataset sequentially, starting from the beginning each time we draw the last sample. At each round, the algorithm receives reward 1 if the instance is classified correctly, and 0 otherwise. We compute the total number of classification errors as a performance metric.

It is important to keep in mind that the bandit feedback (correct/incorrect classification) makes the classification problem significantly more challenging, as compared to the standard supervised learning, since the true label is never revealed in bandit setting unless the classification is correct. Thus, the
classification accuracy in a bandit setting is expected to be lower than in the supervised learning setting.

We now describe some details of the experiments. For MNIST, we took 10,000 samples from the original test dataset (clearly, not using them later for testing) to pre-train the encodings, and 60,000 samples from the training dataset to simulate the online bandit with 10 arms corresponding to different digits. For STL-10, 100,000 samples of unlabeled data are used to pre-train the encodings; then the 5,000 test samples together with 8,000 training samples are combined to simulate the online bandit, again with 10 different arms corresponding to image classes. For Caltech-101 Silhouettes-28 dataset, out of the original 8671 samples, 671 are used for pre-training and 800 for online learning with 101 different arms (class labels). For CIFAR-10 dataset, 10,000 test set samples are used for pre-training, and 50,000 training samples are left for the online bandit with 10 arms (classes). For Warfarin dataset, 528 test set samples are used for pre-training, and 5,000 training samples are left for the online bandit with 3 arms (classes).

6.2 Nonstationary Environments

We simulated several types of nonstationarity using the above datasets. As mentioned before, we assume that the input data arrive in batches, and the data distribution (i.e., the joint distribution of the context and reward) may change across those batches, while remaining stationary within each batch. We used the batch size of 1,000, and varied the number of embeddings $k$, using $k = 2, 4$, or 8, presenting average results over all $k$.

Nonstationary context: varying cluster distribution

To simulate changes in the context (input) distribution, we first clustered all samples in the corresponding pre-training data subset into $k$ clusters. Next, we generate a sequence of batches, where each batch contained a certain fraction of samples from different clusters, and these fractions were changing across the batches, i.e., the probability distribution of cluster membership was changing, simulating nonstationary input.

Nonstationary context: negative images

Another type of input nonstationarity involved introducing negative images as inputs with same semantics but different textures. Namely, with probability $p$, the negative image of the original image was presented as an input. Experiments were performed in two settings: half ($p = 0.5$) and rand ($0 < p < 1$ randomly assigned for each mini-batch), in both stationary and nonstationary context conditions, with both shuffled and unshuffled rewards (described later).

6.3 Results

We explored different combination of the above nonstationarities. Table 2 summarizes our results for the nonstationary context due to varying cluster distribution, and for mixed-domain (multi-task) settings with unshuffled reward function. As we can see, on three out of six datasets, baseline was still outperforming our embeddings. However, if we consider the mean accuracy in the entire set of experiments, the top three algorithms were: universal embedding (mean accuracy 28.83%), baseline (mean accuracy 27.78%), mini-batch embedding (mean accuracy 27.58%), respectively, suggesting the advantage of representation learning (embedding computation). Moreover, if we take a look at the whole iteration history, for example, for MNIST dataset (Figure 1), we observe that initially, the baseline CB (solid line) is considerably worse than embedding-based approaches, and requires a large number of iteration to finally catch up with them. Figures 2 and 3 show the history of reward accumulation for the STL-10 and CIFAR-10, demonstrating that the baseline is consistently outperformed by embedding selection methods.

Nonstationary reward: multi-task environment

Another type of nonstationarity was assuming that input samples may come from different domains (tasks), and thus can be associated with different subsets of labels (arms). For example, we combined 5,000 randomly selected training samples from each of the two selected domains, MNIST and Warfarin datasets, and extended the set of possible labels (arms) to include 10 labels from MNIST and 3 labels from Warfarin. We used linear stretching to make the input dimensions equal across the two domains. The algorithm had to assign a label to each input without any information about which domain the input came from.

Nonstationary reward: shuffled class labels

We further explored the multi-task setting by introducing a different type of nonstationary reward, where the class labels were shuffled, i.e., randomly permutated, in each batch.

| Datasets | History | Instances | Features | Classes |
|----------|---------|-----------|----------|---------|
| MNIST    | 10,000  | 20,000    | 100      | 10      |
| STL-10   | 20,000  | 10,000    | 100      | 10      |
| CIFAR-10 | 2,000   | 10,000    | 3,072    | 10      |
| Caltech-101 S | 671  | 8,000    | 784      | 101     |
| Warfarin | 528     | 5,000     | 93       | 3       |
| mix: MNIST/Warfarin | 10,528 | 10,000 | 93 | 13 |

Table 2: Nonstationary Environment with Unshuffled Labels

| Datasets   | baseline | uE | mE | oE |
|------------|----------|----|----|----|
| MNIST      | 37.24    | 34.44 | 29.00 | 22.32 |
| STL-10     | 10.29    | 15.81 | 14.77 | 13.43 |
| CIFAR-10   | 9.62     | 14.30 | 13.30 | 11.73 |
| Caltech-101 S | 1.18 | 1.14 | 1.09 | 1.06 |
| Warfarin   | 62.58    | 56.70 | 56.10 | 56.92 |
| mix: MNIST/Warfarin | 45.76 | 50.58 | 51.21 | 47.74 |

Table 3: Nonstationary Environment with Shuffled Labels

| Datasets   | baseline | uE | mE | oE |
|------------|----------|----|----|----|
| MNIST      | 12.19    | 33.75 | 29.04 | 23.83 |
| STL-10     | 10.05    | 16.64 | 15.10 | 12.77 |
| CIFAR-10   | 10.23    | 14.83 | 13.13 | 11.60 |
| Caltech-101 S | 1.00 | 1.09 | 1.23 | 1.30 |
| Warfarin   | 40.66    | 55.10 | 50.86 | 54.44 |
| mix: MNIST/Warfarin | 23.54 | 49.33 | 50.67 | 49.15 |
Next, Table 5 summarizes our results with shuffled reward function, for the nonstationary context due to varying cluster distribution, and for mixed-domain (multi-task) settings. Based on the mean accuracy in the entire experiment, the top three algorithms were: universal embedding (mean accuracy 28.46%), mini-batch embedding (mean accuracy 26.62%), online embedding (mean accuracy 25.52%), respectively. Furthermore, in this experiment, our embedding-based approaches always outperformed the baseline, suggesting that in a setting where reward functions are nonstationary, in addition to the nonstationary input environment, the advantage of representation learning is quite significant, as compared to standard CB (mean accuracy 16.28%). Note that, with nonstationary (shuffled) labels, the reward accumulated by the baseline CB remains significantly below the reward of embedding-based approaches, at all iterations (Figures 4-6). Thus, in a more challenging setting with both context and reward nonstationarities, the embedding-based approaches clearly outperform the standard contextual bandit.

Table 4 summarizes our results for the nonstationary online learning setting with negative environments and unshuffled reward. Based on the mean accuracy in the entire experiment, the top three algorithms were: online embedding (mean accuracy 12.78%), universal embedding (mean accuracy 12.61%), mini-batch embedding (mean accuracy 12.23%), respectively. Again, the embedding-based approaches are always superior to the baseline CB; online embedding achieved the best performance among all methods on MNIST, while universal and batch embeddings were taking their turns outperforming the baseline on other datasets and settings.
Finally, Table 5 summarizes our results for the nonstationary online learning setting with negative environments and shuffled reward function. Based on the mean accuracy in the entire experiment, the top three algorithms were: universal embedding (mean accuracy 12.61%), online embedding (mean accuracy 12.14%), mini-batch embedding (mean accuracy 11.80%), respectively, further confirming the advantage of adaptive encoding over standard CB (mean accuracy 7.12%). In addition, the difference of textures under the same semantics introduced in this experiments demonstrated that embedding selection outperforms single universal embedding in most nonstationary cases.

Figures 7-14 visualize the details of reward accumulation over time by different methods, on MNIST data and all the settings from the Tables 4 and 5. The performance gap between the embedding-based approaches and the baseline is especially large in those settings. Furthermore, we can see that both adaptive, context-dependent embedding approaches (oE and mE) consistently outperform the single-embedding approach (uE), with the online embedding emerging as the best one, especially with increasing number of iterations.

7 Conclusions

We introduced an extension of the contextual bandit problem motivated by several real-world applications in non-stationary environments, including recommendation systems, health monitoring and medical diagnosis, and others. In this setting, which we refer to as Contextual Bandit with Representation learning and unlabeled History (CBRH), a set of unlabeled contexts is available prior to online decision making, which allows, instead of using the raw context, to learn context representations. Next, during the online phase, embeddings are selected adaptively, depending on each context, and updated based on the contexts observed so far. We propose two specific algorithms for the CBRH problem, based on online and offline clustering, which combine online embedding selection and learning with contextual Thompson Sampling bandit. The algorithms are evaluated in several types of nonstationary environments and compared to the standard contextual bandit, as well as universal (single) embedding, on several datasets. Overall, we observe clear advantages of the embedding-based approaches over the standard contextual bandit; moreover, the proposed adaptive embedding selection and learning methods frequently outperform the universal embedding in multiple nonstationary settings.

References

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland, pages 39.1–39.26, 2012.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In ICML (3), pages 127–135, 2013.

Robin Allesiardo, Raphaël Féraud, and Djallel Bouneffouf. A neural networks committee for the contextual bandit problem. In Neural Information Processing - 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part I, pages 374–381, 2014.

Peter Auer and Nicolò Cesa-Bianchi. On-line learning with malicious noise and the closure algorithm. Ann. Math. Artif. Intell., 23(1-2):83–99, 1998.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-
time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

Gábor Bartók, Dean P Foster, Dávid Pál, Alexander Rakhlín, and Csaba Szepesvári. Partial monitoring—classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

Djallel Bounefouf and Raphaël Féraud. Multi-armed bandit problem with known trend. *Neurocomputing*, 205:16–21, 2016.

Djallel Bounefouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. Context attentive bandits: Contextual bandit with restricted context. In *Proceedings of IJCAI-2017*, 2017.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *AISTATS*, volume 15 of *JMLR Proceedings*, pages 208–214. JMLR.org, 2011.

Adam Coates and Andrew Y Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 921–928, 2011.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

International Warfarin Pharmacogenetics Consortium et al. Estimation of the warfarin dose with clinical and pharmacogenetic data. *N Engl J Med.*, 2009(360):753–764, 2009.

Pratik Gajane, Tanguy Urvoy, and Emilie Kaufmann. Corrupt bandits. *EWRL*, 2016.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. *CoRR*, 2010.

Jérémie Mary, Romaric Gaudel, and Philippe Preux. Bandits and recommender systems. In *Machine Learning, Optimization, and Big Data - First International Workshop, MOD 2015*, pages 325–336, 2015.

II Ororbia, G Alexander, C Lee Giles, and David Reitter. Online semi-supervised learning with deep hybrid boltzmann machines and denoising autoencoders. *arXiv preprint arXiv:1511.06964*, 2015.

W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

B. Yver. Online semi-supervised learning: Application to dynamic learning from radar data. In 2009 *International Radar Conference "Surveillance for a Safer World"* (RADAR 2009), pages 1–6, Oct 2009.