Thompson Sampling for Combinatorial Bandits and Its Application to Online Feature Selection

Audrey Durand and **Christian Gagné**
Laboratoire de vision et systèmes numériques, Université Laval, Québec (QC), Canada
audrey.durand.2@ulaval.ca, christian.gagne@gel.ulaval.ca

Abstract

In this work, we address the combinatorial optimization problem in the stochastic bandit setting with bandit feedback. We propose to use the seminal Thompson Sampling algorithm under an assumption on rewards expectations. More specifically, we tackle the online feature selection problem where results show that Thompson Sampling performs well. Additionnally, we discuss the challenges associated with online feature selection and highlight relevant future work directions.

Introduction

The standard Multi-Armed Bandit (MAB) setting assumes that a reward is directly associated with an arm. In many real-world applications, the problem has a combinatorial nature, where observed rewards correspond to a function of multiple arms. However, if it is possible to consider every set as a regular arm and apply classical MAB algorithms, this naive approach may lead to a combinatorial explosion of the possible sets. Moreover, by considering each set independently, it does not share information among sets even when they share several single arms. The Combinatorial Multi-Armed Bandit (CMAB) problem recently introduced (Chen, Wang, and Yuan 2013) addresses these issues and covers a large class of combinatorial online learning problems. However, it assumes that the outcome associated with each arm in a set is observable after a play, that is semi-bandit feedback. In this work, we address the combinatorial optimization problem in the bandit setting with full bandit feedback, that is a feedback for the whole set of arms played.

Combinatorial Bandits

The general combinatorial optimization problem in the bandit setting consists of a set of arms $\mathcal{K}$ associated with a set of variables $\{x_{k,t}|t \geq 1\}$, for all k in $\mathcal{K}$. Variable $x_{k,t} \in \mathbb{R}$ indicates the outcome of the k-th arm at episode t. The problem relies on a constraint set of arm subsets $\mathcal{S} \subseteq \mathcal{P}(\mathcal{K})$, where $\mathcal{P}(\mathcal{K})$ is the powerset of $\mathcal{K}$, associated with a set of variables $\{y_{\mathcal{M},t}|t \geq 1\}$, for all $\mathcal{M}$ in $\mathcal{S}$. Variable $y_{\mathcal{M},t} \in \mathbb{R}$ indicates the outcome of subset $\mathcal{M}$ at episode t, where $y_{\mathcal{M},t} = f(\{x_{k,t}|k \in \mathcal{M}\})$. The problem can be formulated as a game where a player sequentially selects subsets in $\mathcal{S}$ and observes rewards according to the played subsets. Let $\mathcal{M}(t)$ denote the subset played at episode t and the reward $r(t) = y_{\mathcal{M}(t),t}$. The reward function $f(\cdot)$ used to compute $y_{\mathcal{M}(t),t}$ might be as simple as a summation of the outcomes of the arms in a subset $\mathcal{M}$ such that $y_{\mathcal{M},t} = \sum_{k\in\mathcal{M}} x_{k,t}$. However, more sophisticated non-linear rewards are allowed. The goal is to maximize the reward over time. Let $\mathcal{M}^* = \operatorname{argmax}_{\mathcal{M}\in\mathcal{S}} \mathbb{E}[y_{\mathcal{M}}]$ be the optimal subset. The expected cumulative regret after T episodes is denoted

$$\mathbb{E}[R(T)] = T \cdot \mathbb{E}\left[y_{\mathcal{M}^*}\right] - \sum_{t=1}^{T} \mathbb{E}\left[y_{\mathcal{M}(t)}\right].$$

We consider the stochastic model, where the outcomes $x_{k,t}$ obtained for an arm k are random variables independent and identically distributed according to some unknown distribution with unknown expectation μ_k. The outcomes distribution can be different for each arm. The global rewards $y_{\mathcal{M},t}$ are therefore random variables independent and identically distributed according to some unknown distribution with unknown expectation $\mu_{\mathcal{M}}$. We also consider the full bandit feedback setting, where only the global reward is observed. Note that the classical bandit problem is a special case of this setting with the constraint $\mathcal{S} = \{\{k\}|k \in \mathcal{K}\}$ such that each subset contains a different single arm and the reward function corresponds to its outcome such that $y_{\mathcal{M},t} = x_{k,t}$, where $k \in \mathcal{M}$. Let $\hat{\mu}_k = \mathbb{E}[y_{\mathcal{M}}|k \in \mathcal{M}, \forall \mathcal{M} \in \mathcal{S}]$ denote the outcome expectation of playing arm k in any subset. It corresponds to the arithmetic mean of $\mu_{\mathcal{M}}$s for all $\mathcal{M}$ containing k. We assume that the *optimal* arms k^* in $\mathcal{M}^*$ have the highest $\hat{\mu}_{k^*}$ such that

$$\operatorname*{argmax}_{\mathcal{M}\in\mathcal{S}} \mathbb{E}[y_{\mathcal{M}}] = \operatorname*{argmax}_{\mathcal{M}'\in\mathcal{S}} \sum_{k\in\mathcal{M}'} \hat{\mu}_k.$$

Under this assumption, it is possible to identify the optimal arms independently of one another and of $\mathcal{M}^*$.

Thompson Sampling

Also known as probability matching, Thompson Sampling has led to promising results on the standard MAB (Graepel et al. 2010; Granmo 2010; Scott 2010). Consider the history $D = \{(\mathcal{M}(1), r(1)), \ldots, (\mathcal{M}(T), r(T))\}$ modelled

Algorithm 1 Thompson Sampling with Bernoulli likelihood for combinatorial optimization problem

assume α_0 and β_0 the priors of the Beta distribution
for each arm k, maintain $n_k(t)$ as the number of times arm k has been played so far and $r_k(t)$ as the cumulative reward associated with arm k
$t = 0$
loop
  $t = t + 1$
  for all arms k in $\mathcal{K}$ **do**
    $\alpha_k(t) = \alpha_0 + r_k(t)$
    $\beta_k(t) = \beta_0 + n_k(t) - r_k(t)$
    sample $\theta_k \sim \text{Beta}(\alpha_k(t), \beta_k(t))$
  end for
  play $\mathcal{M}(t) = \text{argmax}_{\mathcal{M} \in \mathcal{S}} \sum_{k \in \mathcal{M}} \theta_k$
  observe $r(t)$
  update $n_k(t+1)$ and $r_k(t+1)$ for $k \in \mathcal{M}(t)$
end loop

using a parametric likelihood function $P(r|k \in \mathcal{M}, \theta)$ depending on some parameters θ. Given some prior distribution $P(\theta)$ on the parameters, their posterior distribution is given by $P(\theta|D) \propto \prod_{t=1}^{T} P(r(t)|k \in \mathcal{M}(t), \theta) P(\theta)$. The Thompson Sampling heuristic selects the arms of the next subset according to their probability of being optimal. That is, arm k is chosen with probability

$$\int \mathbb{I}\left[\mathbb{E}(r|k,\theta) = \max_{k'} \mathbb{E}(r|k',\theta)\right] P(\theta|D) P(\theta) d\theta,$$

where $\mathbb{I}$ is the indicator function. A standard approach is to model the expected outcome $\hat{\mu}_k$ of each arm using θ parameters. Instead of computing the integral, a θ_k is sampled for each arm k. In the combinatorial optimization problem, Thompson Sampling selects the M arms maximizing θ_k.

Suppose that the rewards follow a Bernoulli distribution. Let $n_k(T)$ denote the number of times that arm k has been pulled up to episode T and

$$r_k(T) = \sum_{t=1}^{T} \mathbb{I}[k \in \mathcal{M}(t)] r(t)$$

represent the cumulative reward associated with arm k up to episode T. The conjugate prior distribution is therefore a Beta with priors α_0 and β_0 such that

$$\theta_k \sim \text{Beta}(\alpha_k, \beta_k),$$

where $\alpha_k(t) = \alpha_0 + r_k(t)$ and $\beta_k(t) = \beta_0 + n_k(t) - r_k(t)$. The corresponding Thompson Sampling with Bernoulli likelihood for the combinatorial optimization problem is described by Algorithm 1.

Online Feature Selection

It is often desirable to select a subset of the original features in order to reduce processing complexity and noise, to remove irrelevant data, or because observing every feature might be too expensive. Most existing studies of feature selection are restricted to batch learning, where the feature selection task is conducted in an offline learning fashion and all the features of training instances are given a priori. Such assumptions may not always hold for real-world applications in which data arrive sequentially and collecting full information data is expensive, which are particularly common in the context of learning with Big Data.

Algorithm 2 Online feature selection as a combinatorial optimization problem in the bandit setting

let M denote the subset size
$t = 0$
loop
  $t = t + 1$
  select the subset $\mathcal{M}(t)$ of features
  receive data $\mathbf{v}_t$, perceiving only the features in $\mathcal{M}(t)$
  make prediction p using $\mathbf{v}_t$
  receive real class c
  $r(t) = H(p \cdot c)$
  update classifier using $\mathbf{v}_t$
  update feature selection heuristic using $r(t)$
end loop

The current work considers the online feature selection problem using partial inputs (Wang et al. 2014). In this challenging scenario, the learner is only allowed to access a fixed small number of features for each training instance. The problem is therefore to decide which subset of features to observe in order to maximize the classification rate. The online feature selection problem must be distinguished from the online streaming feature selection problem (Wu et al. 2013), where all the training instances are available at the beginning of the learning process and their features arrive one at a time. This differs significantly from our online learning setting where instances arrive sequentially. Moreover, the partial inputs constraint prevents using budget online learning and online PCA algorithms as they require full inputs.

We model the online feature selection problem as a combinatorial optimization problem in the bandit setting, where each feature corresponds to an arm. On each episode t, the algorithm selects a subset $\mathcal{M}(t)$ of features to observe on the arriving data. The data is classified using its observable features and a reward is obtained according to success ($r(t) = 1$) or failure ($r(t) = 0$). The whole process is described by Algorithm 2, where $H(\cdot)$ is the Heaviside step function, that is $H(x) = 1$ iff $x \geq 0$, and $H(x) = 0$ otherwise.

Wang et al. (2014) introduced the OFS approach to tackle the online feature selection problem. Given by Algorithm 3, OFS for partial inputs uses a perceptron classifier with sparse weights along with an ε-greedy heuristic for feature subset selection, where the truncation process consists in keeping only the M (absolute) largest weights. In this work, we consider the general online feature selection problem where one is not limited to sparse classifiers. The ε-greedy heuristic considered in OFS for partial inputs cannot be used directly with other classifiers as it is tightly coupled with the weights of OFS. Algorithm 4 describes ε-greedy with a non-sparse classic perceptron, where the greedy sub-

Algorithm 3 OFS for partial inputs (Wang et al. 2014)

let K denote the total number of features; M the subset size; λ the maximum L_2 norm; ε the exploration-exploitation trade-off; and η the step size
$\mathbf{w}_1 = \mathbf{0}^K$
$t = 0$
loop
  $t = t + 1$
  sample $z \sim \text{Bernoulli}(\varepsilon)$
  if $z = 1$ **then**
    randomly select a subset $\mathcal{M}(t)$ of features
  else
    select features associated with non-null weights in $\mathbf{w}_t$ such that $\mathcal{M}(t) = \{k : [\mathbf{w}_t]_k \neq 0\}$
  end if
  receive data $\mathbf{v}_t$, perceiving only the features in $\mathcal{M}(t)$
  make prediction $p = \mathbf{w}_t^\top \mathbf{v}_t$
  receive real class c
  if $p \cdot c \leq 0$ **then**
    compute $\mathbf{v}'_t$ where $[\mathbf{v}'_t]_k = \frac{[\mathbf{v}_t]_k}{\frac{M}{K}\varepsilon + \mathbb{I}([\mathbf{w}_t]_k \neq 0)(1-\varepsilon)}$
    $\mathbf{w}'_{t+1} = \mathbf{w}'_t + \eta c \mathbf{v}'_t$
    $\mathbf{w}'_{t+1} = \min\left(1, \frac{\lambda}{||\mathbf{w}'_{t+1}||_2}\right) \mathbf{w}'_{t+1}$
    $\mathbf{w}_{t+1} = \text{truncate}(\mathbf{w}'_{t+1}, M)$
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t$
  end if
end loop

set now corresponds to the features maximizing the absolute weights in the perceptron.

The performance of algorithms corresponds to their online rate of mistakes given by

$$\text{ORM}(T) = \frac{\text{number of mistakes}}{T},$$

where T corresponds to the number of episodes, that is the total number of data received up to that point. As explained previously, data samples are processed sequentially. For each sample, the heuristic selects the feature subset to perceive. The classifier then predicts the class of the sample using only these features, before receiving the label and updating its weights. As there is no training/testing phase, the performance is computed on the entire datasets. Note that the ORM cannot be compared with the offline classification error rate commonly used for offline algorithms as it cumulates errors along the whole learning process.

Experimental Setting

Experiments are conducted on the benchmark datasets described in Table 1, which have been standardized relatively to their mean and standard deviation. These datasets are all available either on the UCI machine learning repository[1] or the LIBSVM website[2].

[1]http://archive.ics.uci.edu/ml/index.html

[2]http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Algorithm 4 Perceptron with ε-greedy

let K denote the total number of features; M the subset size; ε the exploration-exploitation trade-off; and η the step size
$\mathbf{w}_1 = \mathbf{0}^K$
$t = 0$
loop
  $t = t + 1$
  sample $z \sim \text{Bernoulli}(\varepsilon)$
  if $z = 1$ **then**
    randomly select a subset $\mathcal{M}(t)$ of features
  else
    select features maximizing the absolute weights in $\mathbf{w}_t$ such that $\mathcal{M}(t) = \text{argmax}_{\mathcal{M} \in \mathcal{S}} \sum_{k \in \mathcal{M}} |[\mathbf{w}_t]_k|$
  end if
  receive data $\mathbf{v}_t$, perceiving only the features in $\mathcal{M}(t)$
  make prediction $p = \mathbf{w}_t^\top \mathbf{v}_t$
  $p = 2H(p) - 1$
  receive real class c
  if $p \cdot c < 0$ **then**
    $\mathbf{w}'_{t+1} = \mathbf{w}_t + \eta c \mathbf{v}_t$
    $\mathbf{w}_{t+1} = \text{truncate}(\mathbf{w}'_{t+1}, M)$
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t$
  end if
end loop

Table 1: Datasets characteristics

Dataset	# samples	# features
Covertype (binary)	581 012	54
Spambase	4601	57
Adult (a8a)	32 561	123
Web Linear (w8a)	64 700	300

We compare the online feature selection as a combinatorial optimization problem in bandit setting described by Algorithm 2, using:

- a multilayer perceptron (MLP) (K inputs and one hidden layer of 2 neurons) using Thompson Sampling for Bernoulli likelihood, as given by Algorithm 1, for subset feature selection;
- OFS for partial inputs, as given by Algorithm 3;
- a perceptron using ε-greedy for subset feature selection, as given by Algorithm 4.

The MLP with full feature observation (no subset selection) is used as baseline for comparison.

We configure the experimental setting as in Wang et al. and select $M = \lfloor 0.1 \times \text{dimensionality} + 0.5 \rfloor$ features for making the subsets for every dataset. All classifiers share the same parameter $\eta = 0.2$. For the OFS algorithm, $\lambda = 0.1$. Prior parameters $\alpha_0 = 1$ and $\beta_0 = 1$ are used for Thompson Sampling. For ε-greedy feature selection, experiments were conducted with $\varepsilon \in \{0.05, 0.2, 0.5\}$, but only the best results are reported. The experiments were repeated 20 times, each with a random permutation of the dataset, and the results

reported are averaged over these runs.

Results

Figure 1 shows the evolution of the average rate of classification errors on the different datasets. We observe that using Thompson Sampling for feature subset selection combined with a MLP either minimizes the cumulative error rate or leads to a faster convergence. It even outperforms the MLP that observes all features on Adult and Web Linear datasets.

Challenges and Future Work

In the online setting, data arrive sequentially and are only partially observable. A first challenge consists in designing classifiers that are robust to this situation. Moreover, since the whole dataset is not available for data normalization or standardization, preprocessing techniques should rely on prior knowledge or assumptions on the data. Online preprocessing techniques as in Zliobaite and Gabrys (2014) should therefore be considered to provide a realistic setup for a real-world application.

Measuring the long-term performance in the online setting is another challenge. Unlike offline algorithms, online heuristics carry out the tradeoff between exploration and exploitation *forever*. It is difficult to compare the online feature selection algorithms with their offline counterparts, for which performance is measured in exploitation only.

Acknowledgements This work was supported through funding from FRQNT (Québec) and NSERC (Canada).

References

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 151–159.

Graepel, T.; Candela, J. Q.; Borchert, T.; and Herbrich, R. 2010. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 13–20.

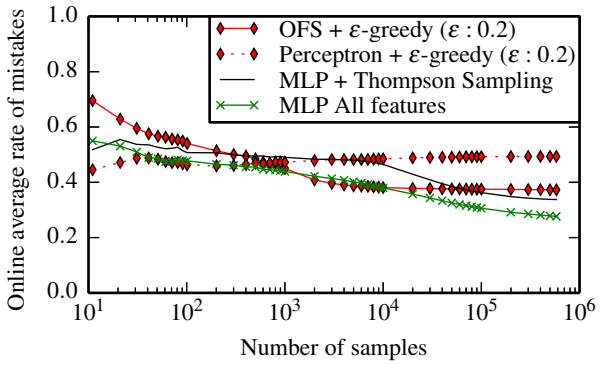
Granmo, O.-C. 2010. Solving two-armed Bernoulli bandit problems using a Bayesian learning automaton. *International Journal of Intelligent Computing and Cybernetics* 3(2):207–234.

Scott, S. L. 2010. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26(6):639–658.

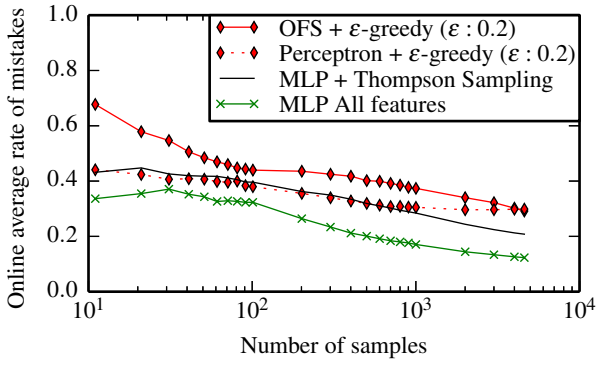
Wang, J.; Zhao, P.; Hoi, S. C. H.; and Jin, R. 2014. Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering* 26(3):698–710.

Wu, X.; Yu, K.; Ding, W.; Wang, H.; and Zhu, X. 2013. Online feature selection with streaming features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 35(5):1178–1192.

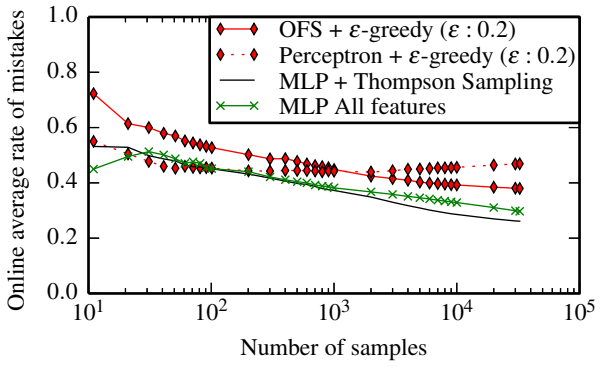
Zliobaite, I., and Gabrys, B. 2014. Adaptive preprocessing for streaming data. *IEEE Transactions on Knowledge and Data Engineering* 26(2):309–321.



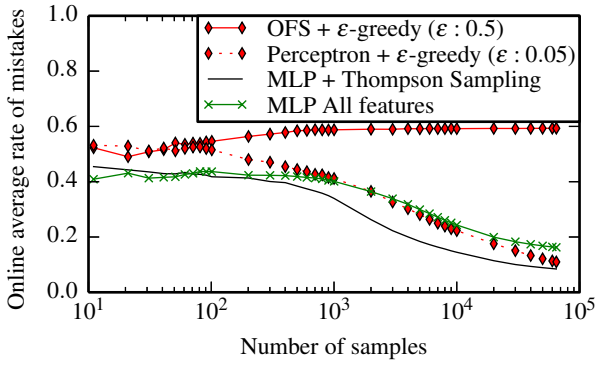
(a) Covertype



(b) Spambase



(c) Adult



(d) Web Linear

Figure 1: ORM on different datasets