Matroid Bandits: Practical Large-Scale Combinatorial Bandits

Branislav Kveton, **Zheng Wen**, **Azin Ashkan**, and **Hoda Eydgahi**
Technicolor Labs
Palo Alto, CA
{*branislav.kveton,zheng.wen,azin.ashkan,hoda.eydgahi*}*@technicolor.com*

Abstract

A matroid is a notion of independence that is closely related to computational efficiency in combinatorial optimization. In this work, we bring together the ideas of matroids and multi-armed bandits, and propose a new class of stochastic combinatorial bandits, *matroid bandits*. A key characteristic of this class is that matroid bandits can be solved both computationally and sample efficiently. We propose a practical algorithm for our problem and bound its regret. The regret scales favorably with all quantities of interest. We evaluate our approach on the problem of learning routing networks for Internet service providers. Our results clearly show that the approach is practical.

Introduction

A multi-armed bandit (Lai and Robbins 1985) is a popular framework for solving learning problems that require exploration. Multi-armed bandits have been successfully applied to a wide range of problems, including stochastic (Gai, Krishnamachari, and Jain 2012) and adversarial (Cesa-Bianchi and Lugosi 2012) combinatorial optimization. The number of feasible solutions in a combinatorial bandit can be huge. In particular, a typical objective is to choose K items out of L, subject to combinatorial constraints. So the total number of potential solutions can be as high as $\binom{K}{L}$. Therefore, it is challenging to design a practical learning algorithm.

In this paper, we propose the first algorithm for solving a broad class of combinatorial bandits that is guaranteed to be computationally and sample efficient. We refer to this class of problems as matroid bandits. A matroid (Whitney 1935) is a generalization of linear independence in combinatorial optimization which is closely related to computational efficiency. In particular, it is well known that the maximum of a constrained modular function can be found greedily if and only if all feasible solutions to the problem are the independent sets of a matroid (Edmonds 1971). Many optimization problems, such as finding a minimum spanning tree, can be formulated as maximizing a modular function on a matroid. As a result, they can be solved greedily.

In this work, we study a learning variant of maximizing a modular function on a matroid. We formalize this problem as finding a maximum-weight basis of a matroid, where all items e in the ground set of the matroid are associated with stochastic weights $\mathbf{w}(e)$. The weights are drawn i.i.d. from some joint probability distribution P. The distribution P is initially unknown, and we learn it by interacting repeatedly with the environment.

We make three contributions. First, we bring together the ideas of matroids (Whitney 1935) and bandits (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002), and propose a novel learning problem of *matroid bandits*. Second, we propose a conceptually simple algorithm for solving our problem, which explores based on the optimism in the face of uncertainty. We refer to the algorithm as *Optimistic Matroid Maximization (*`OMM`*)*. `OMM` is computationally efficient, because the maximum-weight basis in each episode can be found in $O(L \log L)$ time, where L is the number of items. `OMM` is also sample efficient, because its regret is at most linear in all parameters of interest and sublinear in the number of episodes. Finally, we evaluate `OMM` on a real-world problem and demonstrate that it is practical.

Combinatorial bandits have been studied extensively. Gai *et al.* (2012) proposed a UCB-type algorithm for stochastic combinatorial bandits and proved that its expected cumulative regret is $O(K^3 L(1/\Delta^2) \log n)$. This upper bound was later reduced to $O(K^2 L(1/\Delta) \log n)$ by Chen *et al.* (2013). COMBAND (Cesa-Bianchi and Lugosi 2012) and OSMD (Audibert, Bubeck, and Lugosi 2014) are two recently proposed algorithms for adversarial combinatorial bandits. The main limitation of both algorithms is that they are not guaranteed to be computationally efficient. In particular, OSMD projects to the convex hull of exponentially many solutions and COMBAND needs to sample from the distribution over these solutions.

Our problem is a stochastic combinatorial bandit, where all feasible solutions are the independent sets of a matroid. Our gap-dependent regret bound is $O(L(1/\Delta) \log n)$, a factor of K^2 tighter than the best bound for stochastic combinatorial bandits. Our gap-free bound is $O(\sqrt{KLn \log n})$, a factor of $\sqrt{\log n}$ worse than the regret bound of OSMD. In practice, $\sqrt{\log n}$ is small and negligible. On the other hand, OSMD may not be computationally efficient.

Matroid

A *matroid* is a pair $M = (E, \mathcal{I})$, where $E = \{1, \ldots, L\}$ is a set of L items, called the *ground set*, and $\mathcal{I}$ is a family of subsets of E, called the *independent sets*. The family $\mathcal{I}$ has three properties. First, $\emptyset$ is an independent set. Second, all subsets of an independent set are independent. Finally, for all $X \in \mathcal{I}$ and $Y \in \mathcal{I}$ such that $|X| = |Y| + 1$ there exists an item $e \in X \setminus Y$ such that $Y \cup \{e\} \in \mathcal{I}$. This is known as the *augmentation property*. We denote by:

$$E(X) = \{e : e \notin X, X \cup \{e\} \in \mathcal{I}\} \tag{1}$$

the set of items that can be added to set X such that the set remains independent.

A maximal independent set of a matroid is its *basis*. It is well known that all bases of a matroid have the same cardinality, the *rank* of a matroid (Whitney 1935). In this paper, we denote the rank by K.

A *weighted matroid* is a matroid associated with a vector of non-negative weights $\mathbf{w} \in (\mathbb{R}^+)^L$. The e-th entry of $\mathbf{w}$, $\mathbf{w}(e)$, is the weight of item e. The total weight of items in a set $A \subseteq E$ is:

$$f(A, \mathbf{w}) = \sum_{e \in A} \mathbf{w}(e). \tag{2}$$

A classical problem in combinatorial optimization is to find a *maximum-weight basis* of a matroid:

$$A^* = \arg\max_{A \in \mathcal{I}} f(A, \mathbf{w}) = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \mathbf{w}(e). \tag{3}$$

The basis A^* can be constructed greedily (Edmonds 1971) as follows. First, A^* is initialized to $\emptyset$. Second, A^* is iteratively expanded by the items with the highest weight that do not make A^* dependent, until $|A^*| = K$.

Matroids are common in combinatorial optimization. For instance, a *cycle matroid* is the set of all forests in a graph. The ground set of this matroid are the edges of the graph. A set of edges is considered independent if it does not contain a cycle. The basis is a spanning tree. So naturally, the basis with the lowest weight is a minimum spanning tree, a well-known problem in combinatorial optimization.

The weights of the items may not be always known. For instance, suppose that we want to build a spanning tree for network routing where the delays of the links are unknown and have to be learned. In this work, we propose a learning algorithm that can address this type of problems.

Matroid Bandits

We formalize our learning problem as a matroid bandit. A *matroid bandit* is a pair (M, P), where M is a matroid and P is a probability distribution over the weights $\mathbf{w} \in \mathbb{R}^L$ of the items in the ground set E of M. The e-th entry of $\mathbf{w}$, $\mathbf{w}(e)$, is the weight of item e. We assume that the weights $\mathbf{w}$ are drawn i.i.d. from P. The mean weight is denoted by $\bar{\mathbf{w}} = \mathbb{E}[\mathbf{w}]$ and we assume that $\bar{\mathbf{w}}(e) \geq 0$ for all $e \in E$.

Each item is associated with an *arm* and we assume that *multiple arms* can be pulled. A set of arms A can be pulled if and only if it is an independent set. The return for pulling arms A is $f(A, \mathbf{w})$ (Equation 2), the sum of the weights of all items in A. After the arms A are pulled, we observe the individual return of each arm, $\{\mathbf{w}(e) : e \in A\}$. This model of feedback is commonly known as *semi-bandit* (Audibert, Bubeck, and Lugosi 2014).

We assume that the matroid is known and that the distribution of weights P is unknown. Without loss of generality, we assume that the support of P is bounded and is a subset of $[0, 1]^L$. We would like to stress that we do not make any structural assumptions on P.

The solution to our problem is a maximum-weight basis A^* in expectation:

$$A^* = \arg\max_{A \in \mathcal{I}} \mathbb{E}_{\mathbf{w}}[f(A, \mathbf{w})] = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \bar{\mathbf{w}}(e). \tag{4}$$

From the mathematical point of view, this objective is identical to Equation 3. As a result, the maximum-weight basis in expectation can be also found greedily.

Our learning problem is *episodic*. In episode t, we select basis A^t and then gain $f(A^t, \mathbf{w}_t)$, where $\mathbf{w}_t$ is a realization of the weights in episode t. Our goal is to design a policy, a sequence of bases A^t, that minimizes the *expected cumulative regret* in n episodes:

$$R(n) = \mathbb{E}_{\mathbf{w}_1, \ldots, \mathbf{w}_n} \left[\sum_{t=1}^{n} R_t(\mathbf{w}_t) \right], \tag{5}$$

where $R_t(\mathbf{w}_t) = f(A^*, \mathbf{w}_t) - f(A^t, \mathbf{w}_t)$ is the difference in the returns of the optimal and suboptimal bases.

Algorithm

Our solution is designed based on the *optimism in the face of uncertainty* principle (Munos 2012). More specifically, it is a greedy method for finding a maximum-weight basis of a matroid where the expected weights $\bar{\mathbf{w}}(e)$ are substituted for their optimistic estimates $U_t(e)$. We refer to our method as *Optimistic Matroid Maximization (*`OMM`*)*.

The pseudocode of our method is given in Algorithm 1. In each episode t, the method consists of three main steps. First, we compute an *upper confidence bound (UCB)* on the expected weight of each item:

$$U_t(e) = \hat{w}_{e, T_e(t-1)} + c_{t-1, T_e(t-1)}, \tag{6}$$

where $\hat{w}_{e, T_e(t-1)}$ is our estimate of the mean weight $\bar{\mathbf{w}}(e)$ from the first $t-1$ episodes, $c_{t-1, T_e(t-1)} = \sqrt{\frac{2 \log(t-1)}{T_e(t-1)}}$ is the radius of the confidence interval around $\hat{w}_{e, T_e(t-1)}$, and $T_e(t-1)$ is the number of times that item e is chosen prior to episode t.

Second, we order all items e by their UCBs (Equation 6), from the highest to the lowest, and greedily add them to the independent set A^t in this order. The item can be added to the set A^t only if it does not make the set dependent. Since our problem is a matroid, the final set A^t is a basis and is of cardinality K. Finally, we choose the basis A^t, observe the weights of its items, and update our model of the world.

Algorithm 1 OMM: Optimistic matroid maximization.

Input: Matroid $M = (E, \mathcal{I})$

// Initialization
Observe $\mathbf{w}_0 \sim P$
$\hat{w}_{e,1} \leftarrow \mathbf{w}_0(e)$ $\quad \forall e \in E$
$T_e(0) \leftarrow 1$ $\quad \forall e \in E$

for all $t = 1, \ldots, n$ **do**
  // Compute UCBs
  $U_t(e) \leftarrow \hat{w}_{e,T_e(t-1)} + c_{t-1,T_e(t-1)}$ $\quad \forall e \in E$

  // Find a maximum-weight basis with respect to U_t
  Let $e_1^t, \ldots, e_L^t$ be an ordering of items such that:
    $U_t(e_1^t) \geq \ldots \geq U_t(e_L^t)$
  $A^t \leftarrow \emptyset$
  for all $i = 1, \ldots, L$ **do**
    if $(e_i^t \in E(A^t))$ **then**
      $A^t \leftarrow A^t \cup \{e_i^t\}$
    end if
  end for
  Observe $\{\mathbf{w}_t(e) : e \in A^t\}$, where $\mathbf{w}_t \sim P$

  // Update statistics
  $T_e(t) \leftarrow T_e(t-1)$ $\quad \forall e \in E$
  $T_e(t) \leftarrow T_e(t) + 1$ $\quad \forall e \in A^t$
  $\hat{w}_{e,T_e(t)} \leftarrow \dfrac{T_e(t-1)\hat{w}_{e,T_e(t-1)} + \mathbf{w}_t(e)}{T_e(t)}$ $\quad \forall e \in A^t$
end for

Analysis

We prove two upper bounds on the expected cumulative regret of OMM. These bounds can be summarized as:

$$\begin{aligned} \text{Gap-dependent bound:} \quad & O(L(1/\Delta)\log n) \\ \text{Gap-free bound:} \quad & O(\sqrt{KLn\log n}), \end{aligned} \tag{7}$$

where $\Delta = \min_e \min_{k \in \mathcal{O}_e} \Delta_{e,k}$ and $\Delta_{e,k}$ is the *gap* between the expected weights of the k-th item with the highest weight in A^* and suboptimal item e. An item is *suboptimal* if it does belong to A^*. The set $\mathcal{O}_e = \{k : \Delta_{e,k} > 0\}$ are the indices of items in A^* whose expected weight is higher than that of item e. Please refer to Kveton *et al.* (2014) for more details and the proofs of the bounds.

Our theoretical results are significant for several reasons. First, both of our regret bounds are at most linear in K and L, and sublinear in n. In other words, they scale favorably with all quantities of interest and therefore we expect them to be practical.

Second, the total number of solutions in a matroid bandit is usually on the order of $\binom{K}{L}$, exponential in K. Note that our regret bounds do not depend linearly on this quantity.

Finally, we note that our gap-dependent regret bound has the same form as the regret bound of Auer *et al.* (2002) for multi-armed bandits. As a result, we may conclude that the problem of learning a maximum-weight basis of a matroid is not much harder than identifying the best arm in a multi-armed bandit. This result is surprising, because it may seem that the combinatorial structure of a problem could prevent efficient exploration by simple policies, like OMM. This cannot happen in matroid bandits.

Experiments

In this section, we evaluate OMM on the problem of learning routing networks for Internet service providers. Please refer to Kveton *et al.* (2014) for additional results.

Our goal is to learn a *routing network* for an Internet service provider (ISP) that minimizes the expected sum of latencies on its edges. We formulate this problem as learning a minimum spanning tree. We experiment with 6 networks from the *RocketFuel* dataset (Spring, Mahajan, and Wetherall 2004). These networks contain up to 300 nodes and one thousand edges (Table 1). The latency of edge e is modeled as $\mathbf{w}(e) = \bar{\mathbf{w}}(e) - 1 + \varepsilon$, where $\bar{\mathbf{w}}(e)$ is the mean latency, which is recorded in our dataset; and $\varepsilon \sim \mathrm{Exp}(1)$ is noise. The mean $\bar{\mathbf{w}}(e)$ ranges from 1 to 64 milliseconds (Table 1). The main reason for choosing our noise model is that most of the latency in ISP networks can be explained by distance (Choi et al. 2004), the mean latency $\bar{\mathbf{w}}(e)$. The noise is typically small, on the order of several hundred microseconds, and high latency due to noise is unlikely.

Our learning problem can be solved as a matroid bandit. The ground set of the matroid are the edges of the network. A set of edges is independent if it does not contain a cycle. The weight $\mathbf{w}(e)$ is the latency of edge e.

All experiments are episodic. The performance of OMM is measured by the *expected per-step cost* in n episodes:

$$\frac{1}{n}\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_n}\left[\sum_{t=1}^{n} f(A^t, \mathbf{w}_t)\right]. \tag{8}$$

Our approach is compared to two baselines. The first baseline is a minimum-weight basis A^*. This basis is computed based on the expected latencies $\bar{\mathbf{w}}$ and is our notion of optimality. The second baseline is an ε-greedy policy, where the parameter ε is set to 0.1. This setting is common in practice and corresponds to 10% exploration.

In Figure 1, we report our results on three ISP networks. We observe the same trends on all three networks. First, the expected cost of OMM approaches the expected cost of A^* as the number of episodes increases. Second, OMM consistently outperforms the ε-greedy policy in just a few episodes. The expected costs on all networks are reported in Table 1. OMM outperforms the ε-greedy policy, usually by a large margin.

OMM learns relatively quickly because all of our networks are sparse. Specifically, the number of edges in all networks is smaller than four times the number of edges in their spanning trees. Therefore, in theory, each edge can be observed at least once in four episodes and OMM can quickly learn the mean latency of each edge.

Conclusions

In this work, we study the problem of learning how to maximize a modular function on a matroid. The function is initially unknown and we learn it by interacting with the envi-

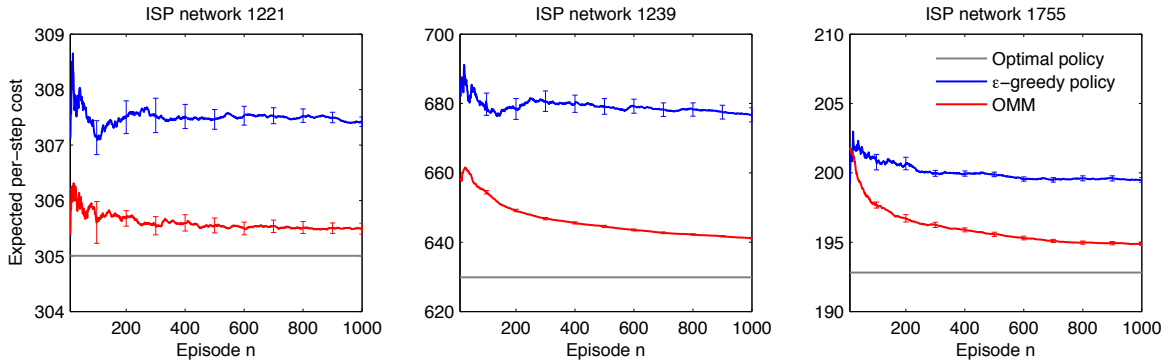

Figure 1: The expected per-step cost of building three minimum spanning trees up to episode $n = 10^3$.

ISP network	Number of nodes	Number of edges	Minimum latency	Maximum latency	Average latency	Optimal policy	ε-greedy policy	OMM
1221	108	153	1	17	2.78	305.00	307.42 ± 0.08	305.49 ± 0.10
1239	315	972	1	64	3.20	629.88	676.74 ± 2.03	641.17 ± 0.18
1755	87	161	1	31	2.91	192.81	199.49 ± 0.16	194.88 ± 0.11
3257	161	328	1	47	4.30	550.85	570.35 ± 0.63	559.80 ± 0.10
3967	79	147	1	44	5.19	306.80	320.30 ± 0.52	308.54 ± 0.08
6461	141	374	1	45	6.32	376.27	424.78 ± 1.54	381.48 ± 0.07

Table 1: Six ISP networks from our experiments and the expected per-step cost of building minimum spanning trees on these networks in episode $n = 10^3$. All latencies and costs are in milliseconds.

ronment. We propose a practical bandit algorithm for solving our problem and analyze its expected cumulative regret. The regret grows sublinearly with time and is at most linear in all quantities of interest. Finally, we evaluate our method on a real-world problem and show that it is practical.

Matroids are a notion of independence that is closely related to computational efficiency in combinatorial optimization (Papadimitriou and Steiglitz 1998). In a sense, they are the hardest problems that can be solved in polynomial time. Our paper shows that one of these problems, maximization of a modular function on a matroid, is efficiently learnable. The key ideas in the design and analysis of our solution are general, and we strongly believe that they can be applied to other problems that involve matroids. One such problem is *maximum-weight matching* on bipartite graphs, which is an instance of maximizing a modular function on the intersection of two matroids. *Minimum-cost flows* are an instance of maximizing a modular function on a *polymatroid*, which is a generalization of a matroid.

References

Audibert, J.-Y.; Bubeck, S.; and Lugosi, G. 2014. Regret in online combinatorial optimization. *Mathematics of Operations Research* 39(1):31–45.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47:235–256.

Cesa-Bianchi, N., and Lugosi, G. 2012. Combinatorial bandits. *Journal of Computer and System Sciences* 78(5):1404–1422.

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*, 151–159.

Choi, B.-Y.; Moon, S.; Zhang, Z.-L.; Papagiannaki, K.; and Diot, C. 2004. Analysis of point-to-point packet delay in an operational network. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*.

Edmonds, J. 1971. Matroids and the greedy algorithm. *Mathematical Programming* 1(1):127–136.

Gai, Y.; Krishnamachari, B.; and Jain, R. 2012. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking* 20(5):1466–1478.

Kveton, B.; Wen, Z.; Ashkan, A.; Eydgahi, H.; and Eriksson, B. 2014. Matroid bandits: Fast combinatorial optimization with learning. *CoRR* abs/1403.5045.

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

Munos, R. 2012. The optimistic principle applied to games, optimization, and planning: Towards foundations of Monte-Carlo tree search. *Foundations and Trends in Machine Learning*.

Papadimitriou, C., and Steiglitz, K. 1998. *Combinatorial Optimization*. Mineola, NY: Dover Publications.

Spring, N.; Mahajan, R.; and Wetherall, D. 2004. Measuring ISP topologies with Rocketfuel. *IEEE / ACM Transactions on Networking* 12(1):2–16.

Whitney, H. 1935. On the abstract properties of linear dependence. *American Journal of Mathematics* 57(3):509–533.