Unmasking "Bandit with Concave Rewards and Convex Knapsack"

Daniel Khashabi

Spring, 2016

Abstract

This is summary of the main ideas from [1] and it is meant to contain a simpler elaboration of the original work. Here I give a somewhat simpler form of the main theorem of the paper and explain my understanding of the rest of the paper.

The paper's goal is an efficient online learning with "concave rewards and convex constraints" in the stochastic bandit setting, i.e. rewards and costs are generated i.i.d. from some unknown underlying distribution. The algorithm proposed here is inspired from UCB algorithm [2, 3] and has strong connections to [4].

1 Introduction

Online learning is the study of the evolution of decision-making game during a T-iteration game, against an adversary which chooses the reward functions. This problem is related to many important problems in existing subfields in science and engineering, for example Convex Optimization, Game Theory, Machine Learning. Due to high-applicability There has been many extensions for this problem.

Here the authors study the effect of hard-constraints on a special family of two-player games know as "bandits". In many existing bandit algorithms it is trivial to apply local constraints, i.e. constraints on the set of valid decisions at each time-stamp. Instead the target is having global constraints, constraints over entire decision process. There are plenty of examples that show importance of global constraints for an online learning agent. For example, consider a scenario in which a robot is interacting with an environment to learn an action. Probably we do not want to the robot to be able to do anything; e.g. actions have different power consumption levels and the robot has a limit energy available; or overdoing some actions might harm the robot.

2 Preliminaries

2.1 Online Learning and Bandits

Suppose there is game between a player and an adversary, over T steps (time horizon). The game has m arms (possible actions that the player can do). At time t, the player plays the action i_t (one of the m actions). Thereafter the adversary returns a reward of r_{i_t}.

The current setting is commonly known as *Online Learning*. The performance of such games is usually measured with regret, which is the performance of the online decision maker compared to the cost of the best single offline action OPT:

$$R(T) = OPT - \sum_{t=1}^{T} r_t.$$

In the cases when the reward contains the information for all of the possible actions, the setting is called "full information". In the setting we are studying the reward is given only for the action done by the decision maker which is commonly known as "bandit" setting.

There are many assumptions we can make about the adversary. If the adversary choses its reward by sampling from an underlying distribution, it is called "stochastic" bandit problem [2]. If there is no assumption on the distributional on form of the choice of the costs, the setting is called "adversarial". The setting studied here is stochastic.

Stochastic Bandits and UCB algorithms

Here we briefly present the *Upper Confidence Bound (UCB)* algorithm [2], a standard algorithm for solving the stochastic bandit problems. Specifically, r_t the rewards for an action at time t, are sampled i.i.d. from an underlying distribution. In other words we can an have mean $\mu_i, \forall i \in \{1, \dots m\}$, such that $\mu_{i_t} = \mathbb{E}[r_{i_t}|i_t]$. The goal is to minimize regret [1]

$$\bar{R}(T) = \max_{i \in \{1,\dots,m\}} \sum_{t=1}^{T} \mu_i - \sum_{t=1}^{T} \mu_{i_t}$$

The idea is to create upper bounds estimates $UCB_{t,i}$ of the mean parameters μ_i, at each time of the algorithm t, and for each action i. In other words, with a high probability,

$$UCB_{t,i} \geq \mu_i, \forall i, t$$

The player at each iteration chooses the action with the highest estimated reward (upper-bound):

$$i_t = \arg \max_{i \in \{1,\dots,m\}} UCB_{t,i}$$

Intuitively speaking, as the agent continues the game, it will get more accurate estimation of the mean parameters μ_i. This is why the approach is sometimes referred to as *optimism in the face of uncertainty*. This is related to the notion of *exploration-exploitation*. In the early stages of the algorithm when the upper-bounds are looser, the algorithm does more *exploration*. As it continues and the confidence bounds get more accurate, the algorithm does more of a *exploitation*.

3 Bandits with Concave Objective and Convex Constraint (BwCR)

We first introduce the setting, then the proposed algorithm which is the modification of the UCB algorithm, and then study the guarantees on the regret.

3.1 Setting

We are playing a T iteration game as explained in Section 2.1. We make the following modifications to our setting:

- **Feedback:** At each iteration t we receive a reward vector feedback $\mathbf{v}_t \in [0,1]^d$. d represents the number of resources.

[1] To be accurate, the definition here is a relaxed form of the regret, commonly called "pseudo-regret", in which the order of expectation and maximization is swiched.

- **Stochastic Assumption:** Reward vectors are generated i.i.d. In other words, there is an unknown underlying distribution on the reward vector, and hence there is a fixed (unknown) mean $\mathbf{V}$ such that

$$\mathbb{E}[\mathbf{v}_t|i_t] = \mathbf{V}_{i_t} \in \mathbb{R}^d, \quad \text{and} \quad \mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_m] \in \mathbb{R}^{d\times m} \tag{1}$$

- **Regret in reward:** We wish to maximize our benefits (in the online game), by comparing our performance to the best offline strategy. Hence minimizing the following:

$$\text{avg-regret}_1(T) \triangleq \text{OPT}_f - f\left(\frac{1}{T}\sum_{t=1}^{T} v_t\right)$$

where the function $f(.)$ is concave, and OPT_f is the best offline value. Suppose there is a best offline strategy for choosing actions denoted with p^*. The best offline value is defined as $\text{OPT}_f \triangleq \mathbb{E}_{\mathbf{v}\sim p^*} f(\mathbf{v})$.

- **Regret in resource consumption:** Define a convex set S which is known to the algorithm. We wish to keep the average consumption as close as possible to this set. Hence minimizing the following:

$$\text{avg-regret}_2(T) \triangleq d\left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{v}_t, S\right)$$

where the distance measure $d(\mathbf{x}, S)$ is defined as $\|\mathbf{x} - \Pi_S(\mathbf{x})\|$, and $\Pi_S(\mathbf{x})$ is the projection onto the convex set S.

- **Special Cases:** The model generalizes the constrained bandit model in [4]. In other words the knapsack constraint can be converted into convex constraint. In the next section we introduce an algorithm which has strong similarities to the UCB algorithm. The authors show that the special case of knapsack constraint will result simplify their algorithm into an LP program, which is (relative) efficiently solvable.

3.2 Proposed algorithm (UCB-BwCR)

Define the empirical average of the cost vector for arm i and component j until time t:

$$\hat{V}_{t,ji} = \frac{\sum_{s=1}^{t} v_t \mathbf{1}\{i_s = i\}}{k_{i,t} + 1} \tag{2}$$

where $k_{i,t}$ is the number of times arm (action) i has been pulled.

Since the problem has two objectives might might be acting againts each other, it would make sense if in addition to UCB estimates (as defined in Section 2.1) we have estimates for lower-bounds on cost vectors. Define Upper-Confidence-Bound (UCB) and Lower-Confidence-Bound (LCB) as followings:

$$\text{UCB}_{t,ji} \triangleq \min\{1, \hat{V}_{t,ji} + 2\text{rad}\left(\hat{V}_{t,ji}, k_{i,t} + 1\right)\} \tag{3}$$

$$\text{LCB}_{t,ji} \triangleq \max\{0, \hat{V}_{t,ji} - 2\text{rad}\left(\hat{V}_{t,ji}, k_{i,t} + 1\right)\} \tag{4}$$

where rad $(\nu, N) = \sqrt{\frac{\gamma\nu}{N}} + \frac{\gamma}{N}$ is the *confidence radius* and $\gamma > 0$ is a positive constant we get to choose.

Denote the set of all cost vectors between UCB and LCB with $\mathcal{H}_t$, which are considered to be estimates for valid actions:

$$\mathcal{H}_t \triangleq \left\{ \tilde{\mathbf{V}} : \tilde{V}_{ji} \in [\text{LCB}_{t,ji}, \text{UCB}_{t,ji}], j = 1, \ldots, d, i = 1, \ldots, m \right\}$$

Define the space of feasible probability distributions over actions with Δ_m:

$$\Delta_m = \left\{ \mathbf{p} : \sum_{i=1}^{m} p_i = 1 \right\}$$

Algorithm 1: UCB-BwCR algorithm

forall the $t = 1, 2, \ldots, T$ **do**

Receive cost vector $\mathbf{v}_t$;

Estimate the empirical mean for cost vectors $\hat{V}_{t,ji}$ (Equation 2) ;

Estimate UCB and LCB estimates (Equation 3 and Equation 4) ;

Calculate

$$\mathbf{p}_t = \arg\max_{p \in \Delta_m} \max_{\substack{\tilde{\mathbf{U}} \in \mathcal{H}_t \\ \text{s.t. } \min_{\tilde{\mathbf{V}} \in \mathcal{H}_t} d(\tilde{\mathbf{V}}\mathbf{p}, S) \leq 0}} f(\tilde{\mathbf{U}}\mathbf{p})$$

If the previous optimization is infeasible, choose $\mathbf{p}_t$ arbitrarily ;

Sample an arm (action) according to p_t ;

end

3.3 Guarantees on the algorithm

Here we provide the guarantees on the Algorithm 1.

Theorem 1. *With probability $1 - \delta$ the regrets of Algorithm 1 is bounded as:*

$$\textit{avg-regret}_1(T) = O\left(L \left\| \mathbf{1} \right\|_d \sqrt{\frac{\gamma m}{T}} \right), \quad \textit{avg-regret}_2(T) = O\left(\left\| \mathbf{1} \right\|_d \sqrt{\frac{\gamma m}{T}} \right)$$

where $\gamma = O\left(\log\left(\frac{mTd}{\delta}\right)\right)$ and $\mathbf{1}_d$ is a d dimensional vector of all 1's.

Before jumping into the proof, first we state two important lemmas that will come handy in the main proof.

Lemma 1 ([5], Lemma 4.9)**.** *If $s_1, \ldots, s_n$ is a sequence of numbers in $[0, 1]$ and $\hat{S}_n = \frac{1}{n}\sum_{i=1}^{n} s_i$ and $S_n = \mathbb{E}\hat{S}_n$, then:*

$$\left| \hat{S}_n - S_n \right| \leq \textit{rad}\left(\hat{S}_n, n \right) \leq 3\textit{rad}\left(S_n, n \right),$$

where rad $(S_n, n) = \sqrt{\frac{\gamma\nu}{n}} - \frac{\gamma}{n}$.

The proof of this lemma is based on direct use Chernoff bound.

Lemma 2 ([4]). *For any two vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}_{+}^{m}$

$$\sum_{i=1}^{m} \textit{rad}\left(a_i, b_i\right) b_i \leq \sqrt{\gamma m (a.b)} + \gamma m$$

Proof. We first use first-order Taylor expansin of the concave function f and use the fact that the function is Lipschitz (its gradient is upper-bounded by a fixed constant). It remains to bound $\left\| \sum_{t=1}^{T} \tilde{\mathbf{V}}_t \mathbf{p}_t - \mathbf{v}_t \right\|$. We break this into multiple inequalities:

1. Based on our model (Equation 1) we know that $\mathbb{E}[\mathbf{v}_t | t_t] = \mathbf{V}_t$. Therefore using Lemma 1 and union bound (over size of the resources d):

$$\left| \sum_{t=1}^{T} \left(V_{ji_t} - v_{t,j}\right) \right| \leq 3\text{rad}\left(\frac{1}{T} \sum_{t=1}^{T} V_{ji_t}, T \right) = O(\sqrt{\gamma T}), \forall j \in [d], \text{ with probability } 1 - de^{-\Omega(\gamma)} \tag{5}$$

2. We know $\mathbb{E}\left[\tilde{\mathbf{V}}_{t,i_t} | \tilde{\mathbf{V}}_t, \mathbf{p}_t\right] = \hat{\mathbf{V}}_t \mathbf{p}_t$. Therefore using Lemma 1 and union bound (over size of the resources d):

$$\left| \sum_{t=1}^{T} \left(\tilde{V}_{t,ji_t} - \tilde{\mathbf{V}}_{t,j}.\mathbf{p}_t \right) \right| \leq 3\text{rad}\left(\frac{1}{T} \sum_{t=1}^{T} \tilde{\mathbf{V}}_{t,j}.\mathbf{p}_t, T \right) = O(\sqrt{\gamma T}), \forall j \in [d], \text{ with probability } 1 - de^{-\Omega(\gamma)} \tag{6}$$

3. First we show that the expected reward vector is close to the actual reward vector, with high probability. From Lemma 1 we know:

$$\begin{aligned}
&\left| \frac{k_{t,i}+1}{k_{t,i}} \hat{V}_{t,ji} - V_{ji} \right| \leq \text{rad}\left(\hat{V}_{t,ji}, k_{t,i} \right) \\
&\Rightarrow \left| \hat{V}_{t,ji} - \frac{k_{t,i}}{k_{t,i}+1} V_{ji} \right| \leq \frac{k_{t,i}}{k_{t,i}+1} \text{rad}\left(\hat{V}_{t,ji}, k_{t,i} \right) \\
&\Rightarrow \left| \hat{V}_{t,ji} - V_{ji} \right| \leq \frac{k_{t,i}}{k_{t,i}+1} \text{rad}\left(\hat{V}_{t,ji}, k_{t,i} \right) + \frac{V_{ji}}{k_{t,i}+1} \leq \text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1 \right) + \frac{V_{ji}}{k_{t,i}+1} \leq 2\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1 \right),
\end{aligned}$$

with probability $1 - e^{-\Omega(\gamma)}$. The last step of the inequality comes from the fact that $\frac{\nu}{n+1} \leq \text{rad}\left(\nu, n\right)$, based definition of $\text{rad}\left(.\right)$.

Next we show with high probability the correct (unknown) mean belongs to $\mathcal{H}_t$.

$$\left| \hat{V}_{t,ji} - V_{ji} \right| \leq 2\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1 \right), \text{ with probability } 1 - e^{-\Omega(\gamma)}$$

$$\hat{V}_{t,ji} - 2\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1 \right) \leq V_{ji} \leq \hat{V}_{t,ji} + 2\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1 \right), \text{ with probability } 1 - e^{-\Omega(\gamma)}$$

Or in other words:

$$\text{LCB}_{t,ji} \leq V_{ji} \leq \text{UCB}_{t,ji}, \text{ with probability } 1 - e^{-\Omega(\gamma)}$$

This needs to hold for any action $i \in [m]$, any resource $j \in [d]$ and any time $t \in [T]$. Therefore with union bound we have:

$$\text{LCB}_{t,ji} \leq V_{ji} \leq \text{UCB}_{t,ji}, \forall i, j, t, \text{ with probability } 1 - mTde^{-\Omega(\gamma)} \tag{7}$$

And also for any $\tilde{V}_{t,ji_t}$ that $\text{LCB}_{t,ji} \leq \tilde{V}_{t,ji_t} \leq \text{UCB}_{t,ji}$ we know that $\left|\tilde{V}_{t,ji_t} - \hat{V}_{t,ji}\right| \leq 2\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1\right)$. Combining this with 7 we get

$$\left|\tilde{V}_{t,ji_t} - \hat{V}_{t,ji}\right| \leq 4\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1\right), \forall i,j,t, \quad \text{with probability } 1 - mTde^{-\Omega(\gamma)}$$

Summing over time t:

$$\begin{aligned}
\left|\sum_{t=1}^{T}\left(\tilde{V}_{t,ji_t} - V_{ji}\right)\right| &\leq \sum_{t=1}^{T}\left|\tilde{V}_{t,ji_t} - V_{ji}\right| \\
&\leq \sum_{t=1}^{T} 4\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1\right) \quad \text{(grouping terms together with the same arm)} \\
&\leq 4\sum_{i=1}^{m}(k_{T,i}+1)\text{rad}\left(\hat{V}_{t,ji}, k_{t,i}+1\right) \quad \text{(using Lemma 2)} \\
&\leq O\left(\sqrt{\gamma mT}\right) \quad \forall i,j, \quad \text{with probability } 1 - mTde^{-\Omega(\gamma)} \qquad (8)
\end{aligned}$$

Combining 6, 5, 8 and using Cauchy Schwarz inequality, we will get:

$$\left\|\sum_{t=1}^{T}\tilde{\mathbf{V}}_t\mathbf{p}_t - \mathbf{v}_t\right\| = O\left(\|\mathbf{1}\|_d\sqrt{\gamma mT}\right) \quad \text{with probability } 1 - mTde^{-\Omega(\gamma)}$$

□

3.4 Hard Constraints

One can make the constraints harder by shrinking the constraint convex set with a margin of ϵ (denote with S^{ϵ}). This will create stronger constraint set; on the other hand this will add ϵK to the regret term. [2]

3.5 Linear Contextual version of BwCR

In contextual bandit unlike the standard bandits, the importance of actions are dependent on the "context" on which they are being done. In other words whether a single action is optimal or not depends on its context. A simple example is to consider two contexts "weekday" and "weekend". An action which might be optimal during "weekend" is not necessarily the best action for the "weekday".

Just like the standard bandits, in the contextual bandit problem, on each of T rounds a learner is presented with the choice of taking one of K actions. Before making the choice of action, the learner observes a feature vector (context) associated with each of its possible choices. In this setting the learner has access to a hypothesis class, in which the hypotheses receives action features (context) and predict which action will give the best reward. If the learner can guarantee to do nearly as well as the prediction of the best hypothesis in hindsight (to have low regret), the learner is said to successfully compete with that class.

If we ignore the contextual information we can just use the existing vanilla bandit algorithms. Therefore having the contextual information one should be able to get better guarantees. More importantly, it is desirable to get rid of the number of actions m in the regret bounds stated in Theorem 1.

[2] As side note, it is important to notice that the constraint is satisfied in expectation in this paper, although it is not explicitly mentioned.

The treatment of authors consider a special case of UCB-BwCR when the objective functions are linear. Also there is a fixed context associated with each action (arm). [3] The only change made to their algorithm is redefining $\mathcal{H}_t$ as an ellipsoid, instead of rectangle with upper and lower confidence bounds [4]

3.6 Efficient Implementation

Although the object function in the UCB-BwCR (Algorithm 1) is convex, existence of an efficient (polynomial-time computational complexity) or not is not clear.

The authors present an approximate reformation of the same objective function resulted from careful linearization of the objective and the constraint. With linearization we can think of the constraint $\mathcal{H}_t$ as a simplex. For satisfying the constraint, we want to find $\mathbf{p}_t$ such that $\tilde{\mathbf{V}}\mathbf{p}_t \in S$, for some feasible cost estimate $\tilde{\mathbf{V}} \in \mathcal{H}_t$. If we think of $\mathcal{H}_t$ as simplex, it can be encoded half-spaces $H_S(\theta) = \{\mathbf{x} : \mathbf{x}.\theta \leq h_S(\theta)\}$ and $h_S(\theta) \triangleq \max_{\mathbf{s}\in S} \mathbf{s}.\theta$ (the half-space is parametrized by θ). With the definition of half-space, we are looking for $\mathbf{p}_t$'s such that $\tilde{\mathbf{V}}\mathbf{p}_t$ is included inside the half-space for some $\tilde{\mathbf{V}} \in \mathcal{H}_t$. For a fixed half-space, a point in $H_S(\theta)$ can be found by minimizing $\theta.\mathbf{x}$. The overall program can be written as the following:

$$(\mathbf{p}_t, \tilde{\mathbf{V}}) = \arg \min_{\mathbf{p}\in\Delta_m} \min_{\tilde{\mathbf{V}}\in H_y} \theta_t.\left(\tilde{\mathbf{V}}\mathbf{p}\right)$$

It can be shown that the inner optimization is trivial and it reaches to its optimal solution on the vertices of the simplex (on UCB/LCB) and objective could equivalently be written as following:

$$\mathbf{p}_t = \arg \min_{\substack{\mathbf{p}\in\Delta_m \\ \text{s.t. } (\phi_t.\mathbf{Z}_t(\phi_t))\mathbf{p}\leq h_S(\phi_t)}} (\theta_t.\mathbf{Z}_t(\theta_t))\,\mathbf{p} \tag{9}$$

where $\mathbf{Z}_t$ encodes a vertex of $\mathcal{H}_t$ and is defined as following:

$$\mathbf{Z}_t(\phi_t)_{ji} \triangleq \begin{cases} \text{UCB}_{t,ji}, \theta_j \leq 0 \\ \text{LCB}_{t,ji}, \theta_j > 0 \end{cases}$$

and θ_t, ϕ_t characterize the half-spaces which are chosen by an online convex optimization (either in primal or dual). Perhaps the most point in this part is that the overall objective can be decomposed as two separate problems: An online problem with concave function, and a constraint satisfaction problem ([1], Theorem 6.1). Hence the regret bound of the overall problem can be calculate based off the decomposed problems.

Given the simplified objective functions the authors present two algorithms for solving it. Essentially we can design algorithms for proper choices of θ_t, ϕ_t in Equation 9. First approach is based on the ideas in [8] which studies the equivalence between online convex optimization and Blackwell approachability. Essentially this works establishes *reductions* from Blackwell approachability to a convex program, and vice versa. The resulting online program optimizes on a modified version version of the Fenchel dual to choose its next point (once for the concave reward, and once for the convex constraint).

Another algorithm is solving the problem in the primal form, it is based on Frank-Wolfe algorithm [9, 10] (also known as *Conditional Gradient* methods). The method is known for its power in converting

[3] Usually in the literature it is assumed that the contexts are sampled from a distribution (e.g. see [6]); however the authors assume that there is a *fixed* context associated with each action. This probably significantly reduces the hardness of the problem. The limitation seem to be addressed in a follow up paper by the same authors.

[4] Ituitively why this change is necessary/important is not clear to me; the authors also don't mention any reason and just refer to a previous work that has introduced it [7].

constrained optimization problems (which usually demand a costly projection step) to an unconstrained program. [5] For optimizing the convex objective, the choice of Frank-Wolfe for θ_t turns our to be the popular gradient.

One issue with the Frank-Wolfe algorithm is that it demands *smooth* functions (i.e. can be upper-bounded by a quadratic function from above). This can be problematic when optimizing with distance constraint. The authors present ideas from [11] on how to find an smooth approximation of the function. The idea is using the notion of the Fenchel duality. Essentially the dual of the dual function is a smooth approximation of the function itself. In the case that a function is smooth, the dual of dual function, is exactly the same as function itself [11]. The ideas presented for speeding up the optimization problem seem to be more general this specific problem here; in fact, a very similar presentation of the idea is presented for constrained online convex optimization in the follow up work by authors [12].

References

[1] Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006. ACM, 2014.

[2] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.

[3] Rajeev Agrawal. Sample mean based index policies with o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995.

[4] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216. IEEE, 2013.

[5] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2008.

[6] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Adv. Neural Info. Proc. Sys. 21* (NIPS), pages 817–824, 2008.

[7] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[8] Jacob Abernethy, Peter L. Bartlett, and Elad Hazan. Blackwell approachability and low-regret learning are equivalent. *CoRR*, abs/1011.1936, 2011.

[9] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[10] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

[5] The algorithm actually has subtle differences from the common Frank-Wolfe algorithm [10].

[11] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Springer, 2004.

[12] Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1405–1424. SIAM, 2015.