Learning and Earning Under Noise and Uncertainty*

Su Jia
Tepper School of Business,
Carnegie Mellon University

Overview

Sequential decision-making under uncertainty is central to a range of operations and marketing problems. In the face of an unknown environment, the decision-maker needs to strike a balance between learning the known environment ("learning") and selecting nearly optimal decisions ("earning"). For instance, consider pricing a new product. If the retailer had full information about the demand at every price level, she could determine and select the revenue-maximizing price for the good. However, such information about the demand curve is typically not available in practice, so the seller needs to experiment with different prices to gain information about the demand curve, and then exploit this information by offering a near-optimal selling price.

The trade-off between learning and optimization can be modelled as the Multi-Armed Bandits (MAB) problem, which has attracted significant attention from a range of communities in recent years, including machine learning, operations research and marketing. While most of the fundamental problems in this area have been theoretically well-understood, these algorithms have been rarely deployed in practice. In contrast, while marketing research on sequential decision making has been focused on the practical side, their results are usually empirical and lacking of rigorous analysis.

This thesis serves as a preliminary step towards filling this gap. We will consider *practical* sequential decision-making problems arising from some of the most fundamental marketing areas including survey design, pricing and content recommendation, and provide *theoretical* insights via provable performance guarantees.

*Dissertation committee: R. Ravi (Chair), Andrew A. Li, Alan Scheller-Wolf and Sridhar Tayur.

Optimal Decision Tree Problem Under Noisy Outcomes

From Spotify to Netflix, we are surrounded by extreme personalization every day. Consumers have come to expect that same level of personalization from companies of all sizes. Investing in personalization efforts to build relationships and create better experiences can pay off with serious rewards for brands. And in a world where the vast majority of companies are focused on improving personalization, companies that do not prioritize creating a tailored experience run the risk of getting left behind.

One approach to personalized service for new users is by classifying users into typical user-types and then identifying the user-type based on their responses to survey questions. The problem of designing efficient surveys is accurately modelled by the Optimal Decision Tree Problem (ODT), where the decision maker needs to perform a sequence of tests to identify an unknown hypothesis drawn from a known distribution. The basic version of ODT has been widely studied for decades and an asymptotically best-possible approximation algorithm has been devised. However in practice, the test outcomes are usually noisy, due to, for example, user heterogeneity within each group, rendering these algorithms inapplicable to real world problems.

This motivates us to study a generalization, called the Optimal Decision Tree Problem with Noisy Outcomes (ODTN), where the outcomes are contaminated by persistent noise, that is, the outcomes of certain tests may be flipped, but remains the same each time the test is performed. More generally, we introduce a problem, Submodular Function Ranking with Noise, that further generalizes the above problem.

Despite the extensive literature on ODT, little is known about the noisy version from the perspective of approximation factor. There are two main reasons. First, the persistence of noise disables most of the statistical learning tools such as concentration bounds. Secondly, the structure of the optimal solution becomes significantly more complicated under noisy outcomes, posing substantial challenge for the analysis of approximation ratio.

We design new approximation algorithms for both the non-adaptive setting, where the test sequence must be fixed *a-priori*, and the adaptive setting where the test sequence depends on the outcomes of prior tests. Our new approximation algorithms provide guarantees that are nearly best-possible and work for the general case of a large number of noisy outcomes per test or per hypothesis, with performance degrading smoothly with this number. Moreover, our numerical evaluations show that despite our theoretical logarithmic approximation guarantees, our methods give solutions with cost very close to the information theoretic minimum.

This chapter is based on Jia et al. (2019). Further, our joint work (Gan et al. (2021a,b)) extended the results to the error-budgeted version and received the *2021 Pierskalla Best Paper Award in Healthcare Applications* for its novel application in liquid biopsy, an emerging cancer detection technique.

Markdown Pricing Under Unknown Demand

Dynamic pricing under unknown demand has been theoretically well-understood, usually under the framework of multi-armed bandits. But in practice, these bandit-based policies are rarely deployed by real-world retailers, largely because oscillating prices may cause customer dissatisfaction. For example, Luca and Reshef (2021) found that a "1% price increase (in menu prices) leads to a 3% to 5% decrease in online ratings on average".

This motivates us to consider dynamic pricing under a monotonicity constraint, that is, the prices must be non-increasing. While both markdown pricing under *known* demand and *unconstrained* pricing under unknown demand have been well-understood, little is known for this problem. In particular, the following basic question remains open prior to this work:

What is the optimal regret bound for markdown pricing? In particular, how does this bound compare with the known bounds for unconstrained pricing?

For instance, under the Lipschitz assumption, Kleinberg (2005) showed an asymptoticly best possible $\Theta(T^{2/3})$ regret bound for unconstrained pricing in T rounds. Can we show that the optimal regret bound for markdown pricing is asymptoticly higher than $T^{\frac{2}{3}}$?

We provide a *complete settlement* of this fundamental question. More precisely, we present optimal regret bounds for markdown pricing, under various assumptions, from the most agnostic setting where only the minimal assumptions are imposed for deriving meaningful guarantees, to the most fine-grained setting where the demand curve is assumed to come from certain class of "parametric" functions. Furthermore, in almost every regime, our tight bound is asymptoticly higher than the known bounds under the same assumptions for unconstrained pricing, highlighting the extra complexity introduced by the monotonicity constraint.

Finally, we also investigate various extensions, including the scenario where the monotonicity constraint can be relaxed at a given cost. This work also opens up a wealth of other related new directions for future study. The above results are based on Jia et al. (2021) and Jia et al. (2022).

Short-Lived High-Volume Bandits: Algorithms and Field Experiment

We consider the problem of recommending short-lived contents to users. By and large, recommendation tasks can be classified into four categories based on the *lifetime* and *volume* of contents generated. For persistent (long-lived) content, the problem is arguably straightforward: spend a small amount of time collecting sufficient data in the form of user feedback, and then apply suitable offline predictive model, which may range from a basic collaborative filtering algorithms to deep neural networks (DNNs). Orthogonal to content lifetime, when there is a *low volume* of content relative to the number of users, the problem is similarly well-understood: dedicated exploration methods (e.g. A/B testing) are sufficient for finding the right segments of users for which the content is most appealing.

Naturally then, the most challenging settings are where the content to be recommended is *short-lived* and *high-volume*. Such settings arise, for example, in content aggregation platforms (e.g. Apple News) and platforms with content that is entirely user-generated (e.g. TikTok). In these settings, both previous approaches are prone to failure: offline predictive algorithms do not receive enough data on individual content to achieve meaningful accuracy due to the short lifetime, and dedicated exploration methods are ill-suited to high volume.

The question then is, how should an online platform decide what content to display to each user? In addition to the well-known "learn-and-earn" trade-off in MAB, the online platform needs to resolve an additional concern: the balance between the exploration of newly arriving and older contents. We propose a simple bandit-based approach for recommending short-lived content, which we show to have nearly-optimal performance guarantee. Our *Sieve Policy* iteratively "weeds out" inferior arms, so that it can dedicate the resources to obtaining finer estimates for the promising arms.

Most importantly, on the practical side we demonstrate the effectiveness of our Sieve Policy by via a large scale field experiment on Glance, a large Indian lockscreen content platform, which is faced with exactly the aforementioned challenge. Over the course of two weeks, our policy achieved a 12% improvement in conversions rates, relative to the deep neural network based control policy.

Chapter I Optimal Decision Tree and Submodular Ranking with Noisy Outcomes

The classic Optimal Decision Tree (ODT) problem involves identifying an initially unknown *hypothesis* h that is drawn from a known probability distribution over a set of hypotheses. We can perform *tests* in order to distinguish between these hypotheses. Each test produces a binary outcome (positive or negative) and the precise outcome of each test-hypothesis pair is known beforehand, and thus an instance of ODT can be viewed as a ± 1-valued matrix M with the tests as rows and hypotheses as columns. The goal is to identify the true hypothesis h using the fewest tests.

As a motivating application, consider the following task in medical diagnosis detailed in Loveland (1985). A doctor needs to diagnose a patient's disease by performing tests. Given an *a priori* probability distribution over possible diseases, what sequence of tests should the doctor perform to identify the disease as quickly as possible? Another application is in active learning (e.g. Dasgupta (2005)). Given a set of data points, one wants to learn a classifier that labels the points correctly as positive and negative. There is a set of m possible classifiers which is assumed to contain the true classifier. In the Bayesian setting, which we consider, the true classifier is drawn from some known probability distribution. The goal is to identify the true classifier by querying labels at the minimum number of points in expectation (over the prior distribution).

Despite the considerable literature on the classic ODT problem, an important issue that is not considered is that of unknown or noisy outcomes. In fact, our research was motivated by a data-set involving toxic chemical identification where the outcomes of many hypothesis-test pairs are stated as unknown. While prior work incorporating noise in ODT, for example Golovin et al. (2010), was restricted to settings with very few noisy outcomes, in this paper, we design approximation algorithms for the noisy optimal decision tree problem in full generality.

Specifically, we generalize the ODT problem to allow unknown/noisy entries (denoted by "$*$") in the test-hypothesis matrix M, to obtain the *Optimal Decision Tree with Noise* (ODTN)

problem, in which the outcome of each noisy entry in the test-hypothesis matrix M is a random ± 1 value, independent of other noisy entries. More precisely, if the entry $M_{t,h} = *$ (for hypothesis h and test t) and the realized hypothesis is h, then the outcome of t will be a random ± 1 value. We will assume for simplicity that each noisy outcome is ± 1 with uniform probability, though our results extend directly to the case where each noisy outcome has a different probability. We consider the standard *persistent* noise model, where repeating the same test always produces the same outcome. Note that this model is more general than the non-persistent noise (where repeating a noisy test leads to "fresh" independent ± 1 outcomes), since one may create copies of tests and hypotheses to reduce to the persistent noise model.

We consider both non-adaptive policies, where the test sequence is fixed upfront, and adaptive policies, where the test sequence is built incrementally and depends on observed test outcomes. Evidently, adaptive policies perform at least as well as non-adaptive ones. Indeed, there exists instances where the relative gap between the best adaptive and non-adaptive policies is very large (see for example, Dasgupta (2005)). However, non-adaptive policies are very simple to implement, requiring minimal incremental computation, and may be preferred in time-sensitive applications.

In fact, our results hold in a significantly more general setting, where the goal is to cover *stochastic* submodular functions. In the absence of noisy outcomes, the non-adaptive and adaptive versions of this problem were studied by by Azar and Gamzu (2011) and Navidi et al. (2020). Other than the ODT problem, this submodular setting captures a number of applications such as multiple-intent search ranking, decision region determination and correlated knapsack cover. Our work is the first to handle noisy outcomes in all these applications.

I.1. Contributions

We derive most of our results for the ODTN problem as corollaries of a more general problem, Submodular Function Ranking with Noisy Outcomes, which is a natural extension of the Submodular Function Ranking problem, introduced by Azar and Gamzu (2011).

First, we obtain an $O(\log \frac{1}{\varepsilon})$-approximation algorithm for *Non-Adaptive* Submodular Function Ranking with noisy outcomes (SFRN) where ε is a separability parameter of the underlying submodular functions. As a special case, for the ODTN (both adaptive and non-adaptive) problem, we consider submodular functions with separability $\varepsilon = \frac{1}{m}$, so the above result immediately implies an $O(\log m)$-approximation for non-adaptive ODTN. This bound is the best possible (up to constant factors) even in the noiseless case, assuming $P \neq NP$.

As our second contribution, we obtain an $O(\min\{c \log |\Omega|, r\} + \log \frac{m}{\varepsilon})$-approximation algorithm for *Adaptive* Submodular Ranking with noisy outcomes (ASRN), which implies an $O(\min\{c, r\} + \log m)$ bound for ODTN by setting $\varepsilon = \frac{1}{m}$, where Ω is the set of random outcomes we may observe when selecting elements. The term $\min\{c \log |\Omega|, r\}$ corresponds to the "noise sparsity" of the instance. For the ODTN problem, c (resp. r) is the maximum number of noisy outcomes in each column (resp. row) of the test-hypothesis matrix M. In the noiseless case, $c = r = 0$ and our result matches the best approximation ratio for the ODT and the Adaptive Submodular Ranking problem (Navidi et al. (2020)). In the noisy case, our performance guarantee degrades smoothly with the noise sparsity. For example, we obtain a logarithmic approximation ratio (which is the best possible) as long as the number of noisy outcomes in each row or column is at most logarithmic. For ODTN, Golovin et al. (2010) obtained an $O(\log^2 \frac{1}{p_{min}})$-approximation algorithm which is polynomial-time only when $c = O(\log m)$; here $p_{min} \leq \frac{1}{m}$ is the minimum probability of any hypothesis. Our result improves this result in that (i) the running time is polynomial irrespective of the number of noisy outcomes and (ii) the approximation ratio is better by a logarithmic factor. While the above algorithm admits a nice approximation ratio when there are *few* noisy entries in each row or column of M, as our third contribution, we consider the other extreme, when each test has only a few *deterministic* entries (or equivalently, a *large* number of noisy outcomes). Here, we focus on the special case of ODTN. At first sight, higher noise seems to only render the problem more challenging, but somewhat surprisingly, we obtain a much *better* approximation ratio in this

regime. Specifically, if the number of noisy outcomes in each test is at *least* $m - O(\sqrt{m})$, we obtain an approximation algorithm whose cost is $O(\log m)$ times the optimum and returns the target hypothesis with high probability. We establish this result by relating the cost to a *Stochastic Set Cover* instance, whose cost lower-bounds that of the ODTN instance.

Finally, we tested our algorithms on synthetic as well as a real dataset (arising in toxic chemical identification). We compared the empirical performance guarantee of our algorithms to an information-theoretic lower bound. The cost of the solution returned by our non-adaptive algorithm is typically within 50% of this lower bound, and typically within 20% for the adaptive algorithm, demonstrating the effective practical performance of our algorithms.

Chapter II Markdown Pricing Under Unknown Demand

Consider the problem of dynamic pricing under *unknown* demand. This problem is by now well-studied, and indeed "optimal" solutions exist under numerous variations on (a) the set of demand functions allowed, on (b) how inventory is treated, and on (c) the frequency at which prices are allowed to change, just to name a few. By and large, these problems are modeled as variants of the classic *multi-armed bandit* problem, and optimality (with respect to a performance measure called *regret*) is achieved by striking a carefully-tuned balance between selecting prices to learn the unknown demand function (exploration), and prices to maximize revenue given what has previously been learned (exploitation).

Now a seemingly innocuous assumption made across all of this work, which appears to be critical in achieving meaningful results (i.e. sub-linear regret), is that the price is allowed to be both decreased (*marked down*) and increased (*marked up*). In reality, markdowns are quite common, but this treatment of markups as being equally common and harmless in fact stands in contrast to the *practice* of pricing, where it is well-understood that markups negatively impact customers' perception of a product's value. As observed by Bitran and Mondschein (1997),

"Customers will hardly be willing to buy a product whose price oscillates, from their point of view, randomly over the season...Most retail stores do not increase the price of a seasonal or perishable product despite the fact that the product is being sold successfully."

For this reason, *markdown pricing* (i.e. where markups are not allowed) has long been ubiquitous in retail (Petro (2017)), and remains among the standard set of capabilities that retailers are still seeking to hone – a recent survey (Google (2021)) suggests that up to $39 billion in value is being left on the table due to sub-optimal markdown pricing, and this number is just for one of many sectors of retail ("specialty" retail).

In short, despite the rich literature on dynamic pricing under unknown demand in recent years, a basic question remains open with respect to the salient challenge of markdown pricing: **Is it feasible to achieve any meaningful performance for markdown pricing under unknown demand, and if so, what is the "separation" from ordinary dynamic pricing?** Put another way, does a markdown constraint render dynamic pricing less "effective", and if so, by how much? This work presents the first definitive answer to this basic question by providing an *optimal* policy for markdown pricing, which allows for a precise characterization of the separation between the regret bounds of markdown pricing and ordinary pricing.

II.1. Our Contributions.

We study a canonical pricing problem with an additional *markdown constraint*. Specifically, at each of T discrete time periods, a price x is chosen and a random demand is observed whose mean is given by an unknown demand function $D(x)$. The markdown constraint precisely means that if price x is selected at time period t, then the price at time period $t+1$ can be at most x. We place only minimal assumptions on the demand function: that the corresponding revenue function $R(x) = xD(x)$ be unimodal and Lipschitz (we will see later on that both are necessary), and inventory is assumed to be infinite (though we will later relax this assumption). The goal is

to design a policy which minimizes regret (defined as the difference between the policy's expected total revenue and the maximum total revenue that can be accrued).

Without the markdown constraint, this problem has previously been solved, and it has been shown that there exists a policy which achieves $O(T^{2/3})$ regret (Kleinberg (2005)). This policy selects a certain discrete subset of the prices and treats each price in this discretization as an "arm" in a classic multi-armed bandit problem. So in particular, many (approximately half) of the policy's price changes are markups, and thus the introduction of the markdown constraint seems likely to (a) necessitate a different algorithmic approach, and (b) induce a "separation" in achievable performance as alluded to above.

Against this backdrop, we make the following contributions:

1. **A Markdown Policy and Performance Guarantee:** We introduce a policy which satisfies the markdown constraint, and show that it achieves $\tilde{O}(T^{3/4})$ regret.[†] This immediately answers the first part of our basic question affirmatively: we *are* able to achieve meaningful performance in the form of a sub-linear (in T) regret bound. Moreover, with small but non-trivial modifications to our policy and proof technique, we present a $\tilde{O}(T^{5/7})$ regret can be achieved under twice-differentiability of the revenue function.
2. **Optimality via a Minimax Lower Bound:** We prove that our policy is in fact order-optimal by showing that the regret of *any* policy is at least $\Omega(T^{3/4})$. This answers the second part of our question: the separation between markdown and ordinary pricing is precisely that markdown pricing must incur at least $\Omega(T^{3/4})$ regret, whereas ordinary pricing can achieve $\tilde{O}(T^{2/3})$ regret.

   Our proof uses a novel generalization of the classic Wald-Wolfowitz Theorem for hypothesis testing, which may be of independent interest for proving lower bounds for a broader class of online learning problems.

[†]We use $\tilde{O}$ to hide logarithmic terms in T.

3. **Model Extension with Penalized Markups:** A natural generalization of our model would be one in which markups are allowed, but penalized. While a *complete* treatment of dynamic pricing with penalized markups would be substantial (indeed, we will see that even the choice of how to *model* these penalties is not obvious), we initiate this future direction of research by considering one version in which each markup incurs a fixed, known, additive cost that scales as $\Theta(T^c)$, for some $c \in [0, 1]$. We provide a complete solution for this model, showing that:
   (a) A simple variant of the Successive Elimination Policy, a classical policy for MAB, achieves $\tilde{O}(T^{\text{med}\{\frac{2}{3}, c, \frac{3}{4}\}})$ regret when applied on a suitable discretization of the price space.
   (b) This bound is optimal up to logarithmic factors.

   These results completely characterize the manner in which our penalized markup model interpolates between ordinary pricing and markdown pricing. When the markup penalty is sufficiently low ($c \leq 2/3$), there is effectively no penalty for markups, since the achievable regret matches that for ordinary pricing. This is already quite surprising – for example, one corollary to this is that any sort of *one-time* or constant-sized penalty is an insignificant detractor to marking up (using carefully-constructed policies). When the penalty is sufficiently high ($c \geq 3/4$), this effectively imposes the hard markdown constraint, as it is optimal to *never* markup, and the resulting regret matches that for markdown pricing. Finally, the optimal regret interpolates smoothly between these two regimes for $c \in [\frac{2}{3}, \frac{3}{4}]$.

In practice, the demand functions are usually assumed to have certain functional forms, which may potentially render the demand- learning easier and lead to lower regret bounds. In the next chapter, we investigate this problem and provide a complete settlement.

Chapter III Markdown Pricing Under Unknown Parametric Demand Models

In the previous chapter, we considered the markdown pricing problem where the underlying demand function is unknown, and showed a tight $T^{3/4}$ regret bound over T rounds under *minimal* assump-

tions of unimodality and Lipschitzness in the revenue function. However, in practice the demand functions are usually assumed to have certain functional forms, which may potentially render the demand- learning easier and lead to lower regret bounds. This motivates the following questions:

Q1) Can we strengthen the $T^{3/4}$ regret bound for markdown pricing in this setting?

To see why such improvement is possible, we observe that the proof of the $T^{3/4}$ lower bound in the previous chapter considers pairs of "roof-shaped" revenue functions that are *completely* identical when the price is higher than some p, and diverging for prices lower than p. Thus, any reasonable policy has to carefully reduce the price, halting only when there is sufficient evidence for *overshooting* the optimal price.

However, this is not true in the parametric case. Take linear demand functions as an example. A policy may simply learn the slope and intercept of the underlying demand function at high prices, and then select the optimal price of the estimated demand function in all future rounds. This enables us to design a more powerful class of learn-then-earn type of policies. Now that we surmise that the $T^{3/4}$ regret can be improved under certain parametric assumption, we naturally arrive at our second question:

Q2) Is markdown pricing still harder than unconstrained pricing under these assumptions?

Or more precisely, can we still show a *separation* between markdown and unconstrained pricing, under various parametric assumptions?

While one may answer these two questions for particular families such as linear family, the following is the real challenge.

Q3) Can we find a general framework to unify the regret bounds for different *categories* of families, rather than specific results for specific families?

In this work, we propose such a framework, by introducing a complexity index called *markdown dimension*, that captures the hardness of performing markdown pricing on a given family that

contains the unknown true demand function. Under this framework, we provide efficient markdown policies for each dimension, which we also show to be *best* possible, thereby completely settling the problem of markdown pricing under unknown demand.

In this work, we make the following contributions.

1. **New Complexity Measure of Demand Families:** We introduce a new concept called *markdown dimension* denoted by d, that captures the complexity of performing markdown pricing on a family, answering the third research question. Within this framework, we provide a complete settlement of the problem, as specified below.
2. **Markdown Policies with Theoretical Guarantees:** For each finite $d \geq 0$, we present a efficient markdown pricing policy. Our policies proceed in *phases*, wherein the seller learns the demand by selecting prices at suitable spacing to estimate the true parameter and then makes conservative decisions. We show that for $d = 0$ and $d \geq 1$, our policies achieve regret $O(\log^2 T)$ and $\tilde{O}(T^{\frac{d}{d+1}})$ respectively, settling our first research question.
3. **Tight Minimax Lower Bound:** We complement our upper bounds with a matching lower bound for each markdown dimension. More precisely, we show that $\Omega(\log^2 T)$ regret is tight for dimension $d = 0$, which *separates* it from the $O(\log T)$ regret bound without this monotonicity constraint. For finite $d \geq 1$, we show a $\Omega(T^{d/(d+1)})$ lower bound, which not only matches our upper bound (up to logarithmic factors) but is also asymptoticly higher than the tight $\tilde{\Theta}(T^{1/2})$ bound (see Broder and Rusmevichientong (2012)) without the markdown constraint, settling our second question.
4. **Impact of Smoothness:** We go further in refining our bounds and investigate the impact of smoothness of the revenue function around the optimal price, and extend our upper bounds for a generalization of smoothness that we call the sensitivity parameter $s \geq 2$. For both finite and infinite d, we obtained decreasing upper bounds as s increases from 2. Moreover for $d = \infty$, our tight $T^{\frac{2s+1}{3s+1}}$ regret bound is asymptoticly higher than that for unconstrained pricing, whose optimal regret is known to be $T^{\frac{s+1}{2s+1}}$ (Auer et al. (2007)).

Chapter IV Short-Lived High-Volume Bandits: Algorithms and Field Experiment

IV.1. Introduction

There has been a long history where online platforms leverage the scale of data, especially user attention, to make better decisions for newly-arriving products or contents. By and large, recommendation tasks can be classified into four categories based on the *lifetime* and *volume* of the contents generated (see Figure 1). For persistent (long-lived) content, the problem is arguably straightforward: spend a small amount of time collecting sufficient data in the form of user feedback, and then choose a suitable offline predictive model, which might range from a basic collaborative filtering algorithm to a deep neural network (DNN).

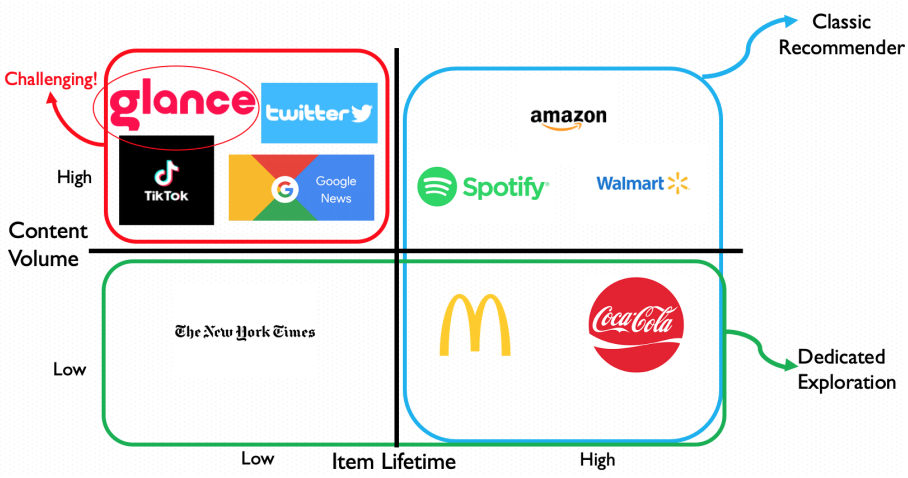

Figure 1 Classification of Recommendation Tasks: Lifetime and Volume

Orthogonal to the lifetime, when there is a *low volume* of contents relative to the number of users, the problem is similarly well-understood: dedicated exploration methods (e.g. A/B testing) are sufficient for finding the right user-segment for which the content is most appealing. For example, LinkedIn runs over 400 concurrent experiments per day to compare different designs of their

website with the goal of, for example, encouraging users to better establish their personal profile, or increasing the subscriptions to LinkedIn Premium (Xu et al. (2015)).

Naturally then, the most challenging setting is where the content to be recommended is *short-lived* and *high-volume* at the same time. Such settings arise, for example, in content aggregation platforms (e.g. Apple News) and platforms with content that is entirely user-generated (e.g. TikTok). In these settings, both of the previous approaches are prone to failure: offline predictive algorithms do not receive enough data on individual content to achieve meaningful accuracy due to the short lifetime, and dedicated exploration methods are ill-suited to the high volume of contents.

To address this challenge, in practice, platform such as Tiktok, Google and Kwai have deployed DNN-based recommender system for short-lived contents, which are frequently re-trained to incorporate the latest data. However, both retrieval of data and retraining of DNN require considerable amount of time and space, posing substantial challenge for the companies in terms of both human and computational resources. To minimize the resources used for exploration, it is better to instead focus on recommendation policies that are (i) operationally simple and (ii) statistically interpretable.

Multi-Armed Bandits (MAB, or simply "bandits") provide a good framework for such policies. In fact, they are not only interpretable, but easy to implement and maintain in a timely manner, as they usually involve simple computation or sampling.

We formulate the aforementioned problem using an MAB model. To model the short-lived contents, we assume in each round K arms arrive, each assumed to be available for exactly W rounds after arrival. Here both the *volume* K and the *lifetime* W are assumed to be known, and in our analysis we assume W is much smaller relative to K. To capture the underlying uncertainty in the conversion rates (e.g. click through rates) of those contents, we assume the *reward* rate $\mu(a)$ of each arm a is unknown. Finally, to model the interaction between users and recommended items,

we assume there are n fixed, identical users, each to be assigned exactly one arm by the platform in every round.

As opposed to most previous work on MAB problems where the worst case input is considered, in this work we assume that the reward rates are independently and identically drawn from a known distribution D. The reason is two-fold. First, this assumption better captures the uncertainty in conversion rates in reality compared to the adversarial model since, in practice, the platform may have access to such distribution using past data. Further, it brings extra *structure* that the learner may utilize for balancing the exploration between arms of different ages. In this work, we will for simplicity assume D to be uniform distribution, but our results still hold as long as D admits (i) a finite support $[a, b]$ and (ii) a cumulative density function F satisfying $1 - F(b - \varepsilon) = O(\varepsilon)$ for any $\varepsilon \in [0, 1]$.

A recommendation policy selects a batch of n available arms in each round, where each arm may be possibly selected for multiple times. It then observes the realized rewards of the selected arms immediately. We also assume that each time arm a is selected, a random reward is identically independently drawn from a subgaussian distribution with mean $\mu(a)$. We measure the quality of a policy by *regret*, that is, the difference between the expected reward of the policy under consideration and that of the optimal policy which knows all reward rates beforehand. In contrast to most previous work on MAB where the worst-case (over all input instances) regret is considered, here we consider the *average regret*, which measures the suboptimality of a policy by averaging over not only the realization of rewards but also the input instances, as we assumed the reward rates are generated from a fixed distribution.

IV.2. Overview of Theoretical Results

IV.2.1. Modeling Recommendation Problem For Short-Lived Contents Our first theoretical contribution is formulating a problem that models the recommendation problem for short-lived items, faced by many online platforms. To be more precise, we consider a batched bandits

	One-By-One	Batched
Immortal	MAB (Lai and Robbins (1985))	Batched Bandits (Perchet et al. (2016))
Mortal	Mortal Bandits (Chakrabarti et al. (2008))	This Work

Table 1 Related Work

model where arms are arriving and expiring over time, with unknown reward rates that are drawn from the uniform distribution.

To expose the inherent hardness of this decision- making problem, we first show two lower bounds. Recall that there are K new arms arriving each period and the policy needs to select n available arms in each round. We first show that any policy suffers $\Omega\left(\frac{1}{K}\right)$ regret. However this lower bound becomes very weak when K is large compared to n. This motivates us our second lower bound: we show that when $K = \Omega(\sqrt{n})$, any policy suffers $\Omega(n^{-1/2})$ regret. At a high level, if the policy aims at $O(n^{-1/2})$ regret, then it needs to "identify" an arm whose reward rate is $O(n^{-1/2})$ lower than the optimal arm. Due to the uniform distribution assumption, the policy has to explore $\sim n^{1/2}$ distinct arms, each incurring $\Omega(1)$ regret on average, leading to $\sim \Omega(n^{1/2}) \cdot \Omega(1)/n = \Omega(n^{-1/2})$ average regret.

IV.2.2. Survival of the Hottest: the Sieve Policy Our second contribution is proposing a novel policy called the *Sieve Policy* which we prove to have a nearly-optimal upper bound. Loosely speaking, the policy iteratively removes the arms that are unlikely to be optimal in its cohort, based on the current reward estimate, and hence focuses on finer estimates of the remaining arms.

We explain the algorithmic idea from the simplest case. Consider the following simple Learn-Then-Optimize policy: assign the newly-arriving arms to a random subset of users, use their interaction data to estimate the reward rates of the arms, and assign the empirically optimal arm to the remaining users. While this policy can be shown to have sublinear regret, it may potentially waste unnecessary resources on obviously suboptimal arms. Rather, a better policy would first use a small

	$K < \sqrt{n}$	$K \geq \sqrt{n}$
Lower Bound	$\frac{1}{K}$	$\frac{1}{\sqrt{n}}$
Upper Bound	$\left(\frac{K}{n}\right)^{\frac{\ell}{2\ell+2}}, \quad \forall \ell \leq W$	$\left(\frac{1}{\sqrt{n}}\right)^{-\frac{W}{2W+2}}$

Table 2 Regret Bounds

amount of resource to "weed out" a significant fraction of inferior arms, and then spend more effort on carefully examining the remaining arms. By choosing suitable parameters, one can show that such a "bi-level" learning policy does improve the regret bound upon the Learn-Then-Optimize policy.

Building upon this observation, we may generalize the bi-level learning policy to any number ℓ of levels, which we call ℓ-Layered Sieve Policy, as long as ℓ does not exceed the lifetime W. We prove that an ℓ-layered Sieve Policy admits regret $O\big(\frac{1}{K} + (\frac{K}{n})^{\frac{\ell}{\ell+2}}\big)$, which decreases as we increase ℓ. One may find the optimal choice of ℓ by setting the two terms to be equal, that is, when $n \sim K^{2+\frac{2}{\ell}}$.

This improvement relies crucially on the assumption that the reward rates are uniformly drawn from a fixed distribution, with which we may quantitatively determine the fraction of arms that can be ruled out using a given amount of exploration resource. Such an improvement is not possible in the worst-case analysis. In fact, no matter how the policy allocates its exploration resources between the first and second level of learning, the "adversary" can always construct set of arms that renders this predetermined allocation either too "careless" (and may hence mistakenly eliminate the optimal arm), or excessively cautious (and may hence waste much resources exploring obviously inferior arms).

Observe that the above upper bound goes to infinity as K tends to infinity. To circumvent this issue, if $K = \Omega(\sqrt{n})$, by replacing randomly sampling $\sim \sqrt{n}$ arms from each batch (and removing the remaining arms), the regret of our sieve policy becomes $O(n^{-\frac{\ell}{2(\ell+1)}})$. Note that this bound converges to the aforementioned lower bound $\Omega(n^{-1/2})$ as ℓ increases. We summarize our theoretical results in Table 2.

Figure 2 Example of Glance Cards

IV.3. Large-Scale Field Experiment

Most importantly, we demonstrate the efficacy of our recommendation policy via a field experiment on the real system of Glance, a large Indian online content-aggregation platform faced with exactly the aforementioned challenge. Glance produces around 200 "Glance cards" (see Figure 2) per hour, over 70% of which expire within 48 hours. Each card consists of a background picture crafted by the Glance marketing team, and a link to an external information source, ranging from news articles to short videos. Swiping through the cards sequentially, a user may click through a card if she finds it interesting, wherein she will be redirected to an external site for further engagement, and then back to the next card in Glance app.

As the main algorithmic challenge, Glance needs to decide which cards to send to each user in a timely manner, based on the users' feedback. Currently, they have deployed a state-of-the-art (DNN) based recommender system. For each new card, they use their DNN to predict its conversion rate for each user, according to which they assign greedily. However, they can only manage to update the DNN every 12 hours, and thus an estimation error may, in the worst case, be adjusted using user-feedback only after 12 hours.

Table 3 Overall Statistics for All Users

			May		July	
			NN	MAB	NN	MAB
Per User-Day	Duration	Mean	175.910	175.548	137.059	142.618
		SE Mean	0.699	0.659	0.6081	0.597
		Median	44.250	44.279	32.973	34.430
	#CT	Mean	1.275	1.273	0.941	1.010
		SE Mean	9.251e-03	8.814e-03	7.276e-03	7.549e-03
Per Impression	Duration	Mean	3.9697	4.0195	4.1183	4.2391
		SE Mean	4.529e-03	4.402e-03	5.738e-03	5.599e-03
		Median	0.693	0.697	0.702	0.703
	CTR	Mean	2.887e-02	2.915e-02	2.827e-02	3.001e-02
		SE Mean	4.698e-05	4.568e-05	5.804e-05	5.671e-05

We implemented our Sieve Policy in a large-scale field experiment on Glance's real system in the first half of July 2021. We observed that our One-Layer Sieve Policy, adapted to practical concerns, outperforms their current DNN-based recommender system by 6% in the number of impressions per user and 12% in the number of conversions per user-day pair, as shown in Figure 3 and 4. We provide the details in the rest of this section.

We analyze the user engagement on two levels: *per-user-per-day* and *per-impression*. Moreover, we consider two natural metrics (duration and click-throughs) for user engagement, hence giving *four* metrics in total. We first consider the overall statistics as summarized in Table 3, where the duration is in seconds. We observe that the user engagement of the two groups are approximately identical in May, but in July the MAB group has a significantly higher mean user engagement. Moreover, such improvement also appeared in *median* duration, indicating that such

Table 4 Significance Testing

		Basic		Bootstrap	
		Z-score	p-value	Z-score	p-value
Per User-Day	Duration	4.610	2.018e-06	4.6197	1.921e-06
	CT	4.259	1.027e-05	4.2556	1.042e-05
Per Impression	Duration	6.963	1.665e-12	6.972	1.556e-12
	CT	12.999	6.127e-39	12.933	1.469e-38

improvement is more likely to be caused by an overall inflation in duration, rather than just a heavier tail in the distribution.

We will show the statistical significance of our improvement using two approaches: significance testing and difference-in-differences (DID) regression.

Significance Test. Suppose the true distribution of the metric of interest (e.g. duration or click-thru) are X_{May}, X_{July} for the NN group and Y_{May}, Y_{July} for the MAB group. For each of these two metrics, we are interested in the *difference-in-differences* before and after the bandit policy was deployed, i.e. $\Delta = (Y^{July} - X^{July}) - (Y^{May} - X^{May})$. We aim to test between the hypotheses

$$H_0 : E[\Delta] \leq 0 \quad \text{vs.} \quad H_1 : E[\Delta] > 0.$$

For $m \in \{\text{May, July}\}$, let $\overline{X}^m, \overline{Y}^m$ be the sample mean in month m, possibly over different number of samples. We first consider the basic Z-score, defined as

$$Z = \frac{\left(\overline{Y}^{July} - \overline{X}^{July}\right) - \left(\overline{Y}^{May} - \overline{X}^{May}\right)}{\sqrt{Var\left(\left(\overline{Y}^{July} - \overline{X}^{July}\right) - \left(\overline{Y}^{May} - \overline{X}^{May}\right)\right)}} \tag{1}$$

Assuming the samples are nearly independent, we may approximate the above as

$$\sqrt{\frac{1}{N_X^{May}} S^2_{X^{May}} + \frac{1}{N_X^{July}} S^2_{X^{July}} + \frac{1}{N_Y^{May}} S^2_{Y^{May}} + \frac{1}{N_Y^{July}} S^2_{Y^{July}}},$$

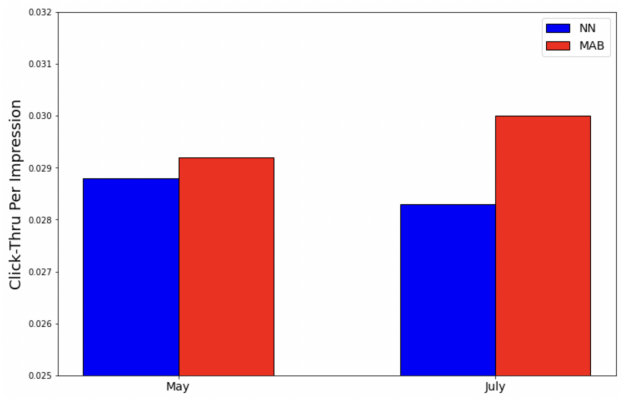

Figure 3 **Click-Through Per Impression**

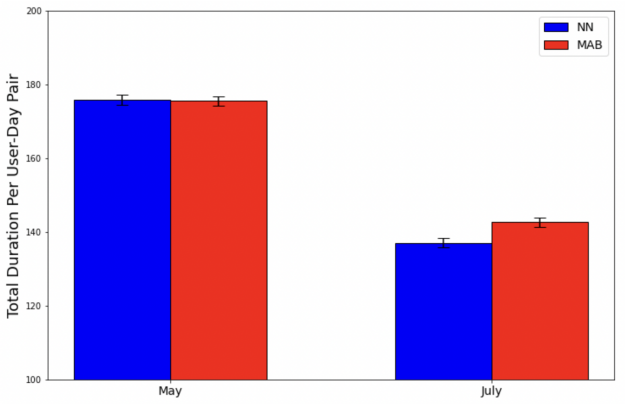

Figure 4 **Duration Per User-Day Pair**

where S_Z^2 is the sample variance of a pool Z of samples. As summarized in the "Basic" column of Table 4, we reject the null hypothesis that the treatment effect is insignificant.

However, in reality the samples are not independent, since (1) each user may appear in both months, (2) a user may have multiple data points in a month, and (3) the same set of glance cards are shown to both the treatment and control group. We remove the dependence by bootstrapping as follows. From each of these four pools of data points, we randomly draw 10^6 samples with replacement, and redefine each $\bar{Z}$ in (1) for each of $Z = X^{May}, X^{July}, Y^{May}, Y^{July}$ to be the bootstrap sample mean. The results with bootstrapping is consistent with our earlier findings, as illustrated in the "Bootstrap" column in Table 4.

Difference-In-Differences Regression. We first illustrate DID regression for per-user-per-day user engagement. To this aim, we vectorize each tuple (u, d, Y_{ud}) into a vector $(t_{ud}, i_{ud}, t_{ud} \cdot i_{ud}, Y_{ud})$ where

$$t_{ud} = \mathbb{1}[\text{day } d \text{ is in July}] \quad \text{and} \quad i_{ud} = \mathbb{1}[\text{user } u \text{ is in MAB group}]$$

denote the time and intervention indicators respectively, and $Y_{ud} \in \{C_{ud}, D_{ud}\}$ is the metric under consideration (i.e. click-throughs or duration of user u on day d). We assume the metric Y_{ud} follows the linear model

$$Y_{ud} = \beta_0 + \beta_1 t_{ud} + \beta_2 i_{ud} + \beta_3 t_{ud} i_{ud} + \varepsilon_{ud} \tag{2}$$

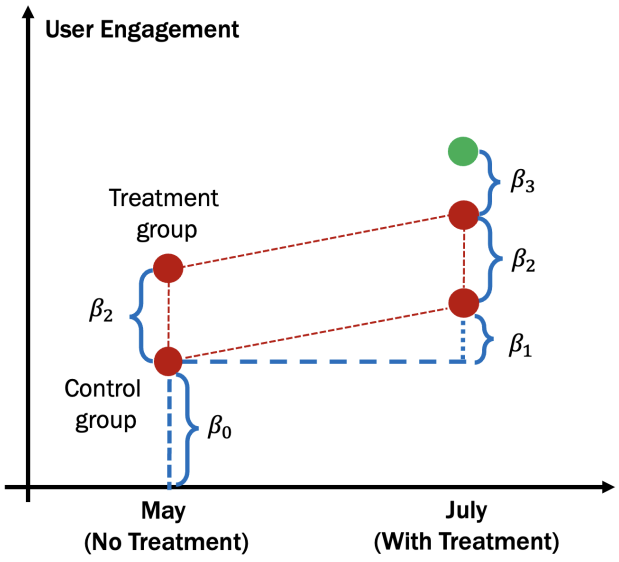

Figure 5 **Illustration of DID regression.**

where $\varepsilon_{ud} \sim N(0, \sigma^2)$ with unknown variance σ^2.

Now suppose there is indeed a positive treatment effect, then the top-right corner of this quadrilateral in Figure 5 (the highest solid red dot) will be raised. The variable β_3 for the composite variable measures exactly this lift. In fact, for day d in July and user u in MAB group, we have $i_{ud} = t_{ud} = 1$, we have $\mathbb{E}Y_{ud} = \beta_0 + \beta_1 + \beta_2 + \beta_3$, which is higher than the hollow dot by β_3. Finally, one can easily verify that β_0 is simply the mean engagement of control group users in May, by setting $t_{ud} = i_{ud} = 0$.

Under the Gaussian noise assumption, we are able to compute confidence intervals and p-values for the coefficients β_i's, as shown in Tables 5. For both duration and CT, the coefficients β_3 are positive, with very low p-values, therefore the treatment effect (i.e. whether or not a user is assigned to the MAB group) is indeed significant. Meanwhile, the coefficients β_2 for the intervention variables have high p-values, confirming that the partition of users is sufficiently random, at least on the per-user-per-day level.

Table 5 Difference-In-Differences Regression

			Coef.	Std. Dev.	t	p-value	0.025Q	0.975Q
Per User-Day	Duration	β_0	175.9103	0.640	274.941	0.000	174.656	177.164
		β_1	-38.8514	0.942	-41.263	0.000	-40.697	-37.006
		β_2	-0.3622	0.887	-0.409	0.683	-2.100	1.375
		β_3	**5.9208**	1.303	4.544	**2.759e-06**	3.367	8.475
	#CT	β_0	1.2750	0.008	153.851	0.000	1.259	1.291
		β_1	-0.3341	0.012	-27.394	1.616e-165	-0.358	-0.310
		β_2	-0.0016	0.011	-0.141	0.888	-0.024	0.021
		β_3	**0.0704**	0.017	4.171	**1.516e-05**	0.037	0.103
Per Impression	Duration	β_0	3.9697	0.005	863.796	0.000	3.961	3.979
		β_1	0.1486	0.007	20.234	2.753e-89	0.134	0.163
		β_2	0.0497	0.006	7.781	3.597e-15	0.037	0.062
		β_3	**0.0711**	0.010	6.998	**1.298e-12**	0.051	0.091
	CTR	β_0	-3.5198	0.002	-2092.794	0.000	-3.523	-3.517
		β_1	-0.0161	0.003	-5.947	1.365e-09	-0.021	-0.011
		β_2	0.0133	0.002	5.712	5.582e-09	0.009	0.018
		β_3	**0.0474**	0.004	12.819	**6.417e-38**	0.040	0.055

Note: All regression are linear regression except for per impression CT, where we applied logistic regression due to binary labels.

References

Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *International Conference on Computational Learning Theory*, pages 454–468. Springer, 2007.

Yossi Azar and Iftah Gamzu. Ranking with submodular valuations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms (SODA'11)*, pages 1070–1079. SIAM, 2011.

Gabriel R Bitran and Susana V Mondschein. Periodic pricing of seasonal products in retailing. *Management science*, 43(1):64–79, 1997.

Josef Broder and Paat Rusmevichientong. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980, 2012.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. *Advances in neural information processing systems*, 21:273–280, 2008.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems*, pages 337–344, 2005.

Kyra Gan, Su Jia, and Andrew Li. Greedy approximation algorithms for active sequential hypothesis testing. *Advances in Neural Information Processing Systems*, 34:5012–5024, 2021a.

Kyra Gan, Su Jia, Andrew Li, and Sridhar R Tayur. Toward a liquid biopsy: Greedy approximation algorithmsfor active sequential hypothesis testing. *Available at SSRN*, 2021b.

Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010 (NIPS'10), Vancouver, British Columbia, Canada.*, pages 766–774, 2010.

Google. Transforming specialty retail with ai. Technical report, 2021.

Su Jia, Viswanath Nagarajan, Fatemeh Navidi, and R. Ravi. Optimal decision tree with noisy outcomes. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 3298–3308, 2019.

Su Jia, Andrew Li, and R Ravi. Markdown pricing under unknown demand. *Available at SSRN 3861379*, 2021.

Su Jia, Andrew A Li, and R Ravi. Markdown pricing under unknown parametric demand models. *Manuscript*, 2022.

Robert D Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704, 2005.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

D. W. Loveland. Performance bounds for binary testing with arbitrary weights. *Acta Inform.*, 22(1):101–114, 1985.

Michael Luca and Oren Reshef. The effect of price on firm reputation. *Management Science*, 2021.

Fatemeh Navidi, Prabhanjan Kambadur, and Viswanath Nagarajan. Adaptive submodular ranking and routing. *Oper. Res.*, 68(3):856–877, 2020.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, Erik Snowberg, et al. Batched bandit problems. *Annals of Statistics*, 44(2):660–681, 2016.

Greg Petro. Markdown mania: A symptom of the wrong product at the wrong price. In *Total Retail*, page Feb 20, 2017.

Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. From infrastructure to culture: A/b testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2227–2236, 2015.