Controlling Polarization in Personalization: An Algorithmic Framework

L. Elisa Celis
Yale University
elisa.celis@yale.edu

Sayash Kapoor
IIT Kanpur
sayash@iitk.ac.in

Farnood Salehi
École Polytechnique Fédérale de Lausanne (EPFL)
farnood.salehi@epfl.ch

Nisheeth Vishnoi
Yale University
nisheeth.vishnoi@yale.edu

ABSTRACT

Personalization is pervasive in the online space as it leads to higher efficiency for the user and higher revenue for the platform by individualizing the most relevant content for each user. However, recent studies suggest that such personalization can learn and propagate systemic biases and polarize opinions; this has led to calls for regulatory mechanisms and algorithms that are constrained to combat bias and the resulting echo-chamber effect. We propose a versatile framework that allows for the possibility to reduce polarization in personalized systems by allowing the user to constrain the distribution from which content is selected. We then present a scalable algorithm with provable guarantees that satisfies the given constraints on the types of the content that can be displayed to a user, but – subject to these constraints – will continue to learn and personalize the content in order to maximize utility. We illustrate this framework on a curated dataset of online news articles that are conservative or liberal, show that it can control polarization, and examine the trade-off between decreasing polarization and the resulting loss to revenue. We further exhibit thefl exibility and scalability of our approach by framing the problem in terms of the more general diverse content selection problem and test it empirically on both a News dataset and the MovieLens dataset.

CCS CONCEPTS

• **Information systems** → *Personalization*; • **Theory of computation** → *Online learning algorithms.*

KEYWORDS

Personalization, recommender systems, polarization, bandit optimization, group fairness, diversification

ACM Reference Format:
L. Elisa Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. 2019. Controlling Polarization in Personalization: An Algorithmic Framework. In *FAT* '19: Conference on Fairness, Accountability, and Transparency, January 29–31, 2019, Atlanta, GA, USA.* ACM, Atlanta, GA, USA, 13 pages. https://doi.org/10.1145/3287560.3287601

FAT '19, January 29–31, 2019, Atlanta, GA, USA*

ACM ISBN 978-1-4503-6125-5/19/01.
https://doi.org/10.1145/3287560.3287601

1 INTRODUCTION

News and social media feeds, product recommendation, online advertising and other media that pervades the internet is increasingly personalized. Content selection algorithms consider a user's properties and past behavior in order to produce a personalized list of content to display [21, 27]. This personalization leads to higher utility and efficiency both for the platform, and for the user, who sees content more directly related to their interests [18, 19]. However, it is now known that such personalization may result in propagating or even creating biases that can influence decisions and opinions. In an important study, [17] showed that user opinions about political candidates, and hence elections, can be manipulated by changing the personalized rankings of search results. Other studies show that allowing for personalization of news and other sources of information can result in a "filter bubble" [29] which results in a type of tunnel vision, effectively isolating people into their own cultural or ideological bubbles; e.g., enabled by polarized information, many people did not expect a Brexit vote or Trump election [10]. This phenomenon has been observed on many social media platforms (see, e.g., [13, 23, 39]), and studies have shown that over the past eight years polarization has increased by 20% [20].

Polarization, and the need to combat it, was raised as a problem in [32], where it was shown that Google search results differ significantly based on political preferences in the month following the 2016 elections in the United States. In a different setting, the ease with which algorithmic bias can be introduced and the need for solutions was highlighted in [35] where it was shown that it is very easy to target people on platforms such as Facebook in a discriminatory fashion. Several approaches to quantify bias and polarization of online media have now been developed [31], and interventions forfi ghting polarization have been proposed [11]. One approach to counter such polarization would be to hide certain user properties so that they cannot be used for personalization. However, this could come at a loss to the utility for both the user and the platform – the content displayed would be less relevant and result in decreased attention from the user and less revenue for the platform (see, e.g., [34]).

> *Can we design personalization algorithms that allow us to avoid polarization yet still optimize individual utility?*

1.1 Groups and Polarization

Often, content is classified into different *groups* which are defined by one or more multi-valued *sensitive attributes*; for instance, news stories can have a political leaning (e.g., conservative or liberal), and a topic (e.g., politics, business or entertainment). More generally, search engines and other platforms and applications maintain topic models over their content (see e.g., [8]). At every time-step, the algorithm must select a piece of content to display to a given user,[1] and feedback is obtained in the form of whether they click on, purchase or hover over the item. The goal of the content selection algorithm is to select content for each user in order to maximize the positive feedback (and hence revenue) received; to do so, it must learn about the topics or groups the user is most interested in. Thus, as this optimal topic is a-priori unknown, the process is often modeled as an online learning problem in which a user-specific probability distribution (from which one selects content) is maintained and updated according to feedback given [28]. As the content selection algorithm learns more about a user, the corresponding probability distribution begins to concentrate the mass on a small subset of topics; this results in polarization where the feed is primarily composed of a single type of content.

1.2 Our Contributions

To counter polarization, we introduce a simple framework which allows us to place *constraints* on the probability distribution from which content is sampled. The goal is to control polarization on the content displayed *at all time steps* (see Section 2.2) and ensure that the given recommendations do not specialize to a *single* group. Our constraints are linear and limit the total expected weight that can be allocated to a given group through lower and upper bound parameters on each group. These polarization constraints are taken as input and can be set according to the context or application. Importantly, though simple, these constraints are versatile enough to control polarization with respect to a variety of metrics which can measure the extent of polarization, or lack thereof, in a given algorithm. This is due to the fact that several fairness metrics depend, e.g., on the ratio or difference between the probability mass on two groups, hence can be implemented by picking appropriate lower/upper bound parameters for the constraints in our setting to give an immediate fairness guarantee (such reductions can be formalized following standard techniques, see, e.g., [12]). Thus, by placing such constraints the content shown to different types of users is varied, and polarization is controlled.

While there are several polynomial time algorithms for similar settings, the challenge is to come up with a *scalable* content selection algorithm for the resulting optimization problem of maximizing revenue (via personalization) subject to satisfying the polarization constraints. We show how an adaptation of an existing algorithm for the unconstrained bandit setting, along with the special structure of our constraints, can lead to a scalable algorithm with provable guarantees for this constrained optimization problem (see Theorem 1). We evaluate this framework and our algorithm on a curated dataset of online news articles that are conservative or liberal, show that it can control polarization, and examine the trade-off between decreasing polarization and the resulting loss to revenue. We further illustrate thefl exibility and scalability of this approach by considering the problem of diverse content selection, and evaluate our algorithm on the MovieLens dataset for diverse movie recommendation as well as the YOW dataset for diverse article recommendation. To the best of our knowledge, this is the first algorithm to control polarization in personalized settings that comes with provable guarantees, allows for the specification of general constraints, and is viable in practice.

[1] In order to create a complete feed, content can simply be selected repeatedly in this manner tofi ll the screen as the user scrolls down; for ease of exposition, we describe the one-step process of selecting a single piece of content.

2 FORMAL DEFINITIONS AND OUR MODEL

2.1 Polarization in Existing Models

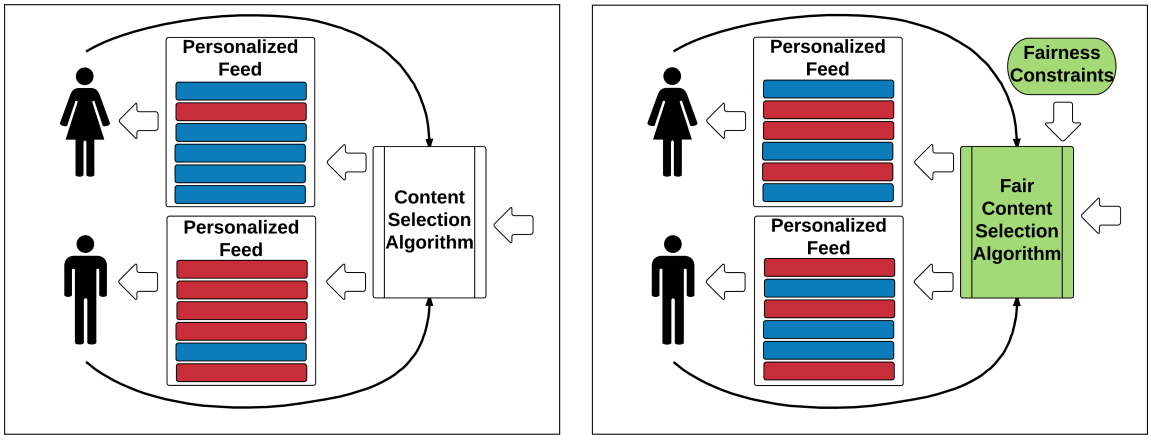
Algorithms for the general (unconstrained, and hence potentially biased) problem of displaying personalized content are often developed in the multi-armed bandit setting (see e.g., [25, 26]). Framed in this manner, at each time step $t = 1, \ldots, T$, a user views a page (e.g., Facebook, Twitter or Google News), and one piece of content (or *arm*) $a^t \in [k]$ must be selected to be displayed. A random *reward* r_a^t, which depends on the selected content is then received by the content selection algorithm. This reward captures resulting clicks, purchases, or time spent viewing the given content.

More formally, at each time step t, a sample $(r_1^t, \ldots, r_k^t)$ is drawn from an unknown distribution $\mathcal{D}$, the player (the content selection algorithm in this case) selects an arm $a \in [k]$ and receives reward $r_a^t \in [0, 1]$. As is standard in the literature, we assume that the r_as are drawn independently across a and t. The rewards $r_{a'}$ for any $a' \neq a$ are assumed to be *unknown* – indeed, there is no way to observe what a user's actions would have been had a different piece of content been displayed. The algorithm computes a probability distribution p^t over the arms based on the previous observations $(a^1, r_a^1), \ldots, (a^{t-1}, r_a^{t-1})$, and then selects arm $a^t \sim p^t$. The goal is to select p^ts in order to maximize the cumulative rewards, and the efficacy of such an algorithm is measured with respect to how well it minimizes *regret* – the difference between the algorithm's reward and the reward obtained from the (unknown) optimal policy. The regret is defined as

$$\mathsf{Regret}_T := \mathbb{E}_{\mathbf{r}^t \sim \mathcal{D}} \left[\sum_{t=1}^{T} r_{a^\star}^t - \sum_{t=1}^{T} r_a^t \right],$$

where $a^\star \in [k]$ is the arm with the highest expected reward: $a^\star := \arg\max_{a \in [k]} \mathbb{E}_{\mathbf{r} \sim \mathcal{D}}[r_a]$. Note that $\mathcal{D}$ (and hence $a^\star$) is a-priori unknown. The regret is a random variable as a^t depends not only on the draws from p^t, but also on the realized *history* of samples $\{(a^t, r_a^t)\}_{t=1}^T$.

The problem with this approach is that, bandit algorithms, by optimizing for that "ideal" $a^\star$, *by definition* strive for polarization. To understand how, let $G_1, \ldots, G_g \subseteq [k]$ be g *groups* of arms which correspond to different types of content across which we do not want to polarize. In the simplest setting, the G_is form a partition (e.g., conservative and liberal news articles when the arms represent news stories), but in general the group structure can be arbitrary. A feature of bandit algorithms is that the probability distribution on the arms, that the algorithm is learning, converges to the action with the best expected reward; i.e., the entire probability mass ends up



(a) (b)

Figure 1: Unfiltered vs. balanced content delivery engines. (a) polarization can occur on using personalized platforms, e.g., primarily showing ads for high-paying jobs (in red) to men and ads for low-paying jobs (in blue) to women (see [15]). (b) With constraints on the extent to which the feeds can differ, our model displays a more balanced feed.

on a single arm, and hence in a single group – causing polarization (see e.g., [25]).

2.2 Our Model

We would like an approach that can control polarization with respect to the groups that the selected arms belong to. Towards this, for each group G_i, let ℓ_i be a lower bound and u_i be an upper bound on the amount of weighted probability mass that we allow the content selection algorithm can place on this group. Formally, we impose the following constraints:

$$\ell_i \leq \sum_{a \in G_i} w_a(G_i) \cdot p_a^t \leq u_i \quad \forall i \in [g], \forall t \in [T], \tag{1}$$

where $w_a(G_i) \in [0,1]$ represents the group weight of arm a on group G_i.

The group weight $w_a(G_i)$ denotes the similarity between arm a and group G_i. For instance, following our earlier discussion on news articles, a conservative leaning news article might have a group weight of 0.9 for the *conservative articles* group and a group weight of 0.1 for the *liberal articles* group, whereas a neutral article might have both of these weights as 0.5. In case of categorical groups (e.g., men vs. women), the group weight can take a binary value. For more general cases (see, e.g., Section 5), this weight can take a real value between 0 and 1. The values for $w_a(G_i)$s can be set using various methods depending on the application and can also take into account error bounds for classifiers that decide whether a given $a \in G_i$ or not. For the case of text documents (e.g., news and scientific articles [37]), the weights can be set using techniques like topic modeling, which give us the percentage of a document that corresponds to a certain topic.

The bounds ℓ_is and u_is provide a handle with which we can ensure that the weighted probability mass placed on any given group is neither too high nor too low at each time step. Rather thanfi xing the values of u_is and ℓ_is, we allow them to be specified as input. This allows one to control the extent of polarization of content depending on the application, and hence (indirectly) encode bounds on a wide variety of existing metrics for different notions of *group fairness* which, in effect, encode the extent of polarization. This requires translating the metric parameters into concrete values of ℓ_is and u_is. For instance, given $\beta > 0$, by setting u_is and ℓ_is such that $u_i - \ell_i \leq \beta$ for all i, we can ensure that the *risk difference* is bounded by β. An additional feature of our model is that no matter what the group structures, or the lower and upper bounds are, the constraints are always linear.

Importantly, note that unlike ignoring user preferences entirely as in [29], the constraints still allow for personalization *across* groups. For instance, if the groups are conservative-leaning vs liberal-leaning articles, and the users are known conservatives or liberals, we may require that $w(\mathrm{C}) \cdot p_{\mathrm{C}}^t \leq 0.75$ and $w(\mathrm{L}) \cdot p_{\mathrm{L}}^t \leq 0.75$ for all t. This ensures that extreme polarization cannot occur – at least 25% of the content a conservative is presented with will be liberal-leaning. Despite these constraints, personalization at the group level can still occur, e.g., by letting $w(\mathrm{C}) \cdot p_{\mathrm{C}}^t = 0.75$ and $w(\mathrm{L}) \cdot p_{\mathrm{L}}^t = 0.25$ for a conservative-leaning user. Furthermore, this framework allows for complete personalization *within* a group; e.g., the conservative-leaning articles shown to conservatives and liberals may differ. This is crucial as the utility maximizing conservative-leaning articles for a conservative may differ from the utility maximizing conservative-leaning articles for a liberal.

The next question we address is how to measure an algorithm's performance against the best *constrained* solution. We say that a probability distribution p on $[k]$ is *constrained* if it satisfies the upper and lower bound constraints in (1), and let C be the set of all such probability distributions. Note that given the linear nature of the constraints, the set C is a polytope (an intersection of a set of half spaces), and hence we can formulate the problem offinding $v^\star$ as a linear programming problem.

Algorithm 1 Constrained-ε-Greedy

Require: Constraint set C, a constrained probability distribution $q_f \in \{q : B_\infty(q, \eta) \subset C\}$, a positive integer T, a constant L that controls the exploration

Initialize $\bar{\mu}_1 := 0$
for $t = 1, \ldots, T$ **do**
  Update $\varepsilon_t := \min\{1, 4/(\eta L^2 t)\}$
  Compute $p^t := \arg\max_{p \in C} \bar{\mu}_t^\top p$
  Sample a from the probability distribution $(1 - \varepsilon_t)p^t + \varepsilon_t q_f$
  Observe reward $r_t = r_a^t$
  Update empirical mean $\bar{\mu}_{t+1}$
end for

An algorithm is said to be constrained if it only selects $p^t \in C$. The *constrained regret* for such an algorithm can be defined as

$$\mathsf{CRegret}_T := \mathbb{E}_{\mathbf{r}^t \sim \mathcal{D}, \tilde{a}^t \sim v^\star} \left[\sum_{t=1}^{T} r_{\tilde{a}}^t - \sum_{t=1}^{T} r_a^t \right],$$

where $v^\star \in C$ represents a point in the constraint set C with the highest expected reward: $v^\star := \arg\max_{p \in C} \mathbb{E}_{\mathbf{r} \sim \mathcal{D}, \tilde{a} \sim p} [r_{\tilde{a}}]$.

Remark1. *We note that there is nothing specific to political polarization (e.g., news articles being grouped according to political leaning with a goal of avoiding polarization) in the model. Instead, we can think of content along with a topic model where the goal is to select content that is diverse across all topics. While the prior notion is our main motivator, the theorems apply to the more general case and we show thefl exibility of the approach by considering both political polarization and diverse recommendation in the empirical evaluation of our approach in Section 5.*

The constraints on the probabilities in (1) can be translated to the constraints on the number of times n_a^T that the arms in groups G_is are selected in T iterations of the algorithm,

$$\ell_i \leq \sum_{a \in G_i} w_a(G_i) \cdot \frac{n_a^T}{T} \leq u_i \quad \forall i \in [g], \tag{2}$$

see the appendix for more details.

3 RELATED WORK

Approaches to Curtail Polarization. There is a large body of work studying the effects of polarization, and ways in which we can combat it. A significant portion of this literature considers interventions to inform or educate users on the effects of personalization and is orthogonal to our work. Pariser, who coined the term "filter bubble", proposes that we simply remove personalization entirely [29]. However, this would come at a complete loss to the utility and efficiency that personalization can bring to both the user and the platform. In contrast, our approach does allow for personalization – up to a point. It ensures that the content is not polarized beyond the given constraints, but within that personalizes in order to maintain high utility. Another approach would be to manipulate the user ratings (e.g., by adding noise or a regularizer to the recommender algorithm) in order to have only approximate preferences; this has been shown to help reduce polarization [4, 5, 38]. We compare against such an approach in our empirical results (Constrained-Ran), and observe that our algorithm significantly outperforms this method. The key difference is that such an approach adds noise to attain de-polarization, while our approach de-polarizes in an informed manner that personalizes content as much as possible subject to the polarization constraints.

Algorithms for Constrained Bandit Optimization. Constrained bandit optimization is a broadfi eld that has arisen in the consideration of a variety of problems unrelated to polarization. For example, knapsack-like constraints on bandit optimization is studied in [6]; however, this work only considers constraints that are placed on the final probability vector p^T, whereas in our setting it is important to satisfy fairness constraints at every time step $\{p^t\}_{t=1}^T$. A different line of work [24] considers *online individual fairness* constraints which require that the probability of selecting all *arms* be approximately equal until enough information is gathered to confidently know which arm is the best. In a similar vein, another work [16] considered budgets on the number of times that any given arm can be selected. Both of these results can be loosely interpreted as working with the special case of our model in which each arm belongs to its own group; their results cannot be applied to our more general setting or be used to curtail polarization.

4 ALGORITHMIC RESULTS

For each arm $a \in [k]$, let its mean reward be $\mu_a^\star$. In this case, the unknown parameters are the expectations of each arm $\mu_a^\star$ for $a \in [k]$. We assume that the reward for the t-th time step is sampled from a Bernoulli distribution with probability of success $\mu_{a^t}^\star$. For a probability distribution $q \in C$ and a small enough constant $\eta > 0$, we define $B_\infty(q, \eta)$ to be the set of all probability distributions that lie inside C, such that a probability distribution $q_f \in B_\infty(q, \eta)$ has at least η probability mass on each arm. More formally, $B_\infty(q, \eta) \in C$ is an ℓ_∞-ball of radius η centered at q. Let $V(C)$ denote the set of vertices of C and $v^\star := \arg\max_{v \in V(C)} \sum_{a \in [k]} \mu_a^\star v_a$.

Theorem1. *Let $\eta > 0$ be a small enough constant. Given the description of C, any probability distribution $q_f \in \{q : B_\infty(q, \eta) \subset C\}$ that lies in the constrained region, and the sequence of rewards, the Constrained-ε-Greedy algorithm (Algorithm 1), run for T iterations, has the following constrained regret bound:*

$$\mathbb{E}\left[\mathsf{CRegret}_T\right] = O\left(\frac{\ln T}{\eta \gamma^2}\right),$$

where $\varepsilon_t = \min\{1, 4/(\eta d^2 t)\}$ and $d = \min\{\gamma, 1/2\}$. The algorithm works for any lower bound L on γ, with a L instead of γ in the regret bound. Here γ is the difference between the maximum and the second maximum expected rewards with respect to the $\mu^\star$s over the vertices of the polytope C. More formally, $\gamma := \sum_{a \in [k]} \mu_a^\star v_a^\star - \max_{v \in V(C) \setminus v^\star} \sum_{a \in [k]} \mu_a^\star v_a$.

Before we present the formal details, wefi rst highlight some key aspects of the algorithm, theorem and proofs.

For general convex sets, γ can be 0 and the regret bound can at best only be $O\left(\sqrt{T}\right)$ [14]. As our constraints result in a constraint set C which is a polytope, unless there are degeneracies, γ is non-zero. In general, γ may be hard to estimate theoretically. However,

Algorithm	Per iteration Running time	Regret Bound
Confidence-Ball$_2$ [14]	NP-Hard problem	$O\left(\frac{k^2}{\gamma}\log^3 T\right)$
OFUL [3]	NP-Hard problem	$\tilde{O}\left(\frac{1}{\gamma}\left(k^2+\log^2 T\right)\right)$
Confidence-Ball$_1$ [14]	$O(k^\omega)+2k$ LP-s	$O\left(\frac{k^3}{\gamma}\log^3 T\right)$
Constrained-ε-Greedy (Algorithm 1)	$O(1)+1$ LP	$O\left(\frac{k}{\gamma^2}\log T\right)$
Constrained-L_1-OFUL (Algorithm 2)	$O(k^\omega)+2k$ LP-s	$\tilde{O}\left(\frac{k}{\gamma}\left(k^2+\log^2 T\right)\right)$

Table 1: The complexity and problem-dependent regret bounds for various algorithms when the decision set is a polytope.

for the settings in which we conduct our experiments, we observe that the value of γ is reasonably large.

When the probability space is unconstrained, it suffices to solve $\arg\max_{i\in[k]}\bar{\mu}_i$, where $\bar{\mu}_i$ is an estimate for the mean reward of the i-th arm. It can be an optimistic estimate for the arm mean in case of the UCB algorithm [9], a sample drawn from the normal distribution with the mean set as the empirical mean for the Thompson Sampling algorithm [7]. When the probability distribution is constrained to lie in a polytope C, instead of a maximum over the arm mean estimates, we need to solve $\arg\max_{p\in C}\bar{\mu}^\top p$. This necessitates the use of a linear program for any algorithm operating in this fashion. At every iteration, Constrained-ε-Greedy solves one LP. We can speed up the LP computation considerably in practice by using the interior points method and warm starting the LP solver from the optimal p found in the previous iteration (see Section 5.1.2).

4.1 Overview of Algorithm 1: Constrained-ε-Greedy.

The algorithm, with probability $1-\varepsilon$ chooses the probability distribution $p^t=\arg\max_{p\in C}\bar{\mu}^\top p$, and with probability ε it samples from a feasible constrained distribution $q_f\in C$ in the η-interior, i.e., there is at least η probability mass on each arm. The reward for each time step t is generated as $r_t\sim\text{Bernoulli}\left(\mu^\star_{a^t}\right)$, where $a^t\sim(1-\varepsilon)p^t+\varepsilon q_f$ is the arm the algorithm chooses at the t^{th} time instant. The algorithm observes this reward and updates its estimate to $\bar{\mu}_{t+1}$ for the next time-step appropriately. Constrained-ε-Greedy is a variant of the classical ε-Greedy approach [9]. Recall that in our setting, an arm is an article (corner of the k-dimensional simplex) and not a vertex of the polytope C. The polytope C sits inside this simplex and may have exponentially many vertices. This is not that case in the setting of [3, 14] – there may not be any ambient simplex in which their polytope sits, and even if there is, they do not use this additional information about which vertex of the simplex was chosen at each time t. Thus, while they are forced to maintain confidence intervals of rewards for all the points in C, this speciality in our model allows us to get away by maintaining confidence intervals only for the k arms (vertices of the simplex) and then use these intervals to obtain a confidence interval for any point in C. Similar to ε-Greedy, if we choose each arm enough number of times, we can build a good confidence interval around the mean of the reward for each arm. The difference is that instead of converging to the optimal arm, our constraints maintain the point inside C and it converges to a vertex of C. To ensure that we choose each arm with high probability, wefi x a constrained point $q_f\in\eta$-interior of C and sample from the point $(1-\varepsilon)p^t+\varepsilon q_f$. Then, as in ε-Greedy, we proceed by bounding the regret showing that if the confidence-interval is tight enough, the optimal of LP with true mean $\mu^\star$ and LP with the empirical mean $\bar{\mu}$ does not change.

4.2 Proof of Theorem 1

Proof. Let $v^\star=\left[v_1^\star,\cdots,v_k^\star\right]\in C$ be the optimal probability distribution. Conditioned on the history at time t, the expected regret of Constrained-ε-Greedy at iteration t can be bounded as follows

$$\begin{aligned}R(t)&=\mu^{\star\top}v^\star-\left((1-\varepsilon_t)\mu^{\star\top}\bar{v}^t+\varepsilon_t\sum_{a=1}^{k}q_{a,f}\mu_a^\star\right)\\&\le(1-\varepsilon_t)\,\mu^{\star\top}\left(v^\star-\bar{v}^t\right)+\varepsilon_t\mu^{\star\top}v^\star\\&\le(1-\varepsilon_t)\,\mu^{\star\top}v^\star 1\left\{\bar{v}^t\neq v^\star\right\}+\varepsilon_t\mu^{\star\top}v^\star,\end{aligned}$$

where $\bar{v}^t=\arg\max_{p\in C}\bar{\mu}^\top p$.

Let $n=4/(\eta d^2)$. For $t\le n$, since $\varepsilon_t=\min\{1,4/(\eta L^2 t)\}$ we have $\varepsilon_t=1$. The expected regret of the ε-greedy is

$$\begin{aligned}&\mathbb{E}\left[\text{CRegret}_T\right]\le\\&\mu^{\star\top}v^\star\sum_{t=n+1}^{T}\mathbb{P}\left(\bar{v}^t\neq v^\star\right)+\mu^{\star\top}v^\star\sum_{t=1}^{T}\varepsilon_t.\end{aligned}\tag{3}$$

Let $\Delta\mu=\bar{\mu}-\mu^\star$. Without loss of generality, let $\mu^{\star\top}v_i>\mu^{\star\top}v_j$ for any $v_i,v_j\in V(C)$ with $i<j$. Hence, $v_1=v^\star$. Let $\Delta_i=\mu^{\star\top}(v_1-v_i)$. As a result $\Delta_1=0$ and $\Delta_2=\gamma$. The event $\bar{v}^t\neq v^\star$ happens when $\bar{\mu}_t^\top v_i>\bar{\mu}_t^\top v_1$ for some $i>1$, that is,

$$\left(\mu^\star+\Delta\mu_t\right)^\top(v_i-v_1)=-\Delta_i+\Delta\mu_t^\top(v_i-v_1)\ge 0.$$

Dataset	# Arms (k)	# Instances	# Iterations (T)	# Groups (g)
PoliticalNews	1356 (avg.)	30 (# days)	10,000	2
MovieLens	25	943 (# users)	1000	19
YOW	81	21 (# users)	10,000	7

Table 2: Overview of datasets used in the empirical results in Section 5.1.3.

As a result, we have

$$\begin{aligned}
\mathbb{P}\left(\bar{v}^t \neq v^\star\right) &= \mathbb{P}\left(\bigcup_{v_i \in V(C)\setminus v_1} \Delta\mu_t^\top (v_i - v_1) \geq \Delta_i\right) \\
&\leq \mathbb{P}\left(\bigcup_{v_i \in V(C)\setminus v_1} \|\Delta\mu_t\|_\infty \|v_i - v_1\|_1 \geq \Delta_i\right) \quad (4) \\
&\leq \mathbb{P}\left(\bigcup_{v_i \in V(C)\setminus v_1} \|\Delta\mu_t\|_\infty \geq \frac{\Delta_i}{2}\right) \\
&= \mathbb{P}\left(\|\Delta\mu_t\|_\infty \geq \frac{\gamma}{2}\right) = \mathbb{P}\left(\bigcup_{j\in[k]} |\Delta\mu_{t,j}| \geq \frac{\gamma}{2}\right) \\
&\leq \sum_{j\in[k]} \mathbb{P}\left(|\Delta\mu_{t,j}| \geq \frac{\gamma}{2}\right). \quad (5)
\end{aligned}$$

In (4) we use Holder's inequality. Let $E_t = \eta \sum_{\tau=1}^{t} \varepsilon_t/2$ and let $N_{t,j}$ be the number of times that we have chosen arm j up to time t. Next, we bound $\mathbb{P}\left\{|\Delta\mu_{t,j}| \geq \frac{\Delta_2}{2}\right\}$.

$$\begin{aligned}
&\mathbb{P}\left(|\Delta\mu_{t,j}| \geq \frac{\gamma}{2}\right) \\
&= \mathbb{P}\left(|\Delta\mu_{t,j}| \geq \frac{\gamma}{2} | N_{t,j} \geq E_t\right)\mathbb{P}\left(N_{t,j} \geq E_t\right) \\
&+ \mathbb{P}\left(|\Delta\mu_{t,j}| \geq \frac{\gamma}{2} | N_{t,j} < E_t\right)\mathbb{P}\left(N_{t,j} < E_t\right) \\
&\leq \mathbb{P}\left(|\Delta\mu_{t,j}| \geq \frac{\gamma}{2} | N_{t,j} \geq E_t\right) + \mathbb{P}\left(N_{t,j} < E_t\right). \quad (6)
\end{aligned}$$

As $q_f \in \{q : B_\infty(q,\eta) \subset C\}$, we have $q_{a,f} > \eta$, i.e., the probability of selecting an arm a is at least $\varepsilon_t q_{a,f}$. Next, we bound each term of (6). First, using Chernoff-Hoeffding bound we have

$$\mathbb{P}\left(|\Delta\mu_{t,j}| \geq \frac{\gamma}{2} \mid N_{t,j} \geq E_t\right) \leq 2\exp\left(-\frac{E_t\gamma^2}{2}\right). \quad (7)$$

Using the Bernstein inequality [36], we have

$$\mathbb{P}\left(N_{t,j} < E_t\right) \leq \exp\left(-\frac{E_t}{5}\right). \quad (8)$$

For $t \leq n$, $\varepsilon_t = 1$ and $E_t = \eta t/2$. For $t > n$ we have

$$\begin{aligned}
E_t &= \frac{\eta\cdot n}{2} + \sum_{i=n+1}^{t} \frac{2}{d^2 i} \geq \frac{2}{d^2} + \frac{2}{d^2}\ln\left(\frac{t}{n}\right) \\
&= \frac{2}{d^2}\ln\left(\frac{et}{n}\right). \quad (9)
\end{aligned}$$

By plugging (7), (8) and (9) in (6) and noting that $\gamma < 1/2$ we get

$$\mathbb{P}\left(\left\|\Delta\mu_{t,j}\right\| \geq \frac{\gamma}{2}\right) \leq \left(\frac{n}{et}\right)^{\frac{\gamma^2}{d^2}} + \left(\frac{n}{et}\right)^{\frac{4}{10d^2}} \leq \left(\frac{n}{et}\right) + \left(\frac{n}{et}\right)^{\frac{4}{10d^2}} \leq 2\left(\frac{n}{et}\right). \quad (10)$$

Plugging (10) in (3) yields

$$\mathbb{E}\left[\text{CRegret}_T\right] \leq \mu^{\star\top} v^\star \left((1 + \frac{2n}{e})\ln T + n\right). \quad (11)$$

By substituting $n = 4/(\eta d^2)$ in the regret above and noting that $\gamma \leq 2d$ we conclude the proof

$$\mathbb{E}\left[\text{CRegret}_T\right] \leq \mu^{\star\top} v^\star \left(\left(1 + \frac{4}{\eta d^2}\right)\ln T + \frac{4}{\eta d^2}\right) = O\left(\frac{\ln T}{\eta\gamma^2}\right).$$

□

4.3 Alternate Approaches and Special Cases

In this section, we briefly outline an alternate approach for solving this problem that results in a different regret / runtime guarantee (see Table 1). We further show that, for certain special cases of the group structure, e.g., if the groups perfectly partition the arms, one can design even faster solutions to the LP.

4.3.1 Algorithm 2: Constrained-L_1-OFUL. Any algorithm for solving the linear bandit problem with an infinite, continuous set of arms can be adapted to solve the constrained multi-armed bandit problem. The constrained multi-armed bandit problem can be thought of as a special case of this type of linear bandit problem, where the continuous space of arms is simply the probability simplex over our discrete arms. Thus, each arm increases the dimensionality of the linear bandit problem by one, and the continuous arm selected at time t corresponds to the probability distribution we select at time t. The difference between these settings is that while one gets rewards for *points in the simplex* in the case of linear bandit problems, we get rewards for the arms themselves (i.e. the vertices of the simplex) in the constrained multi-armed bandit problems. [2] Using these algorithms as a black-box can be inefficient, and does not allow us to come up with practical algorithms for real-world applications.

However, in some cases, we can adapt algorithms for linear bandits to our constrained setting in a way that makes the computations efficient. Consider the OFUL algorithm that appeared in [3]; we will adapt this algorithm to our constrained setting, and we call the adapted algorithm Constrained-L_1-OFUL. Constrained-L_1-OFUL is an example of algorithms for linear bandits being used to solve the constrained multi-armed bandit problem. The key difference between Constrained-L_1-OFUL and OFUL is that instead of using a scaled L_2-ball in each iteration, we use a a scaled L_1-ball, which makes Constrained-L_1-OFUL efficient; without this adaptation the equivalent step in our setting required solving an NP-hard and nonconvex optimization problem.[3] Constrained-L_1-OFUL

[2] This is also what allows us to get fast and efficient algorithms like Constrained-ε-Greedy for the constrained multi-armed bandit setting.

[3] This is similar in spirit to how Confidence-Ball$_2$ can be adapted to Confidence-Ball$_1$ in [14].

incurs $\tilde{O}\left(\frac{k}{\gamma}\left(k^2 + \log^2 T\right)\right)$ regret (see Theorem 3). This gives a worse dependence on k but a better dependence on γ as compared with to Constrained-ε-Greedy (see Table 1), and hence could be beneficial in some settings. However, the runtime is considerably slower than Constrained-ε-Greedy. Instead of maintaining a least-squares estimate of the optimal reward vector, Constrained-ε-Greedy maintains an empirical mean estimate of it denoted by $\bar{\mu}_t$, which is computationally cheaper per iteration. It also solves only one linear program instead of $2k$ linear programs at every iteration. Both of these factors together cause a significant decrease in running time compared to Constrained-L_1-OFUL. Thus, while Constrained-L_1-OFUL theoretically achieves lower regret than Constrained-ε-Greedy in terms of γ, it is not as computationally efficient, and performs worse in practice.

4.3.2 More Efficient LP Solvers for Special Group Structures. For the special case where group weights are binary, i.e.,

$$w_a(G_i) \in \{0, 1\}\ \forall a \in [k],\ i \in [g],$$

and the constraint set have some special structure, we can solve the LP efficiently:

Single Partition. If the groups in the constraint set form a partition, one can solve the linear program in $O(k)$ time via a simple greedy algorithm. Since each part is separate, we can simply put the minimum probability mass as required by the constraints on the best arm of each group, and then put the maximum possible probability mass on arms in descending order of arm utility. This gives a probability vector that satisfies the constraints and is optimal with respect to the reward.

Laminar Constraints. Let the groups $G_1, \ldots, G_g \subseteq [k]$ be such that: $G_i \cap G_j \neq \emptyset$ implies $G_i \subseteq G_j$ or $G_j \subseteq G_i$. The groups form a tree-like data structure, where the children are the largest groups that are subset of the parents. In this case, the LP can be solved efficiently by a greedy algorithm, and we can solve the LP step in $O(gk)$ time exactly. For the sake of brevity and clarity, we defer the full explanation to the appendix.

5 EMPIRICAL RESULTS

In this section we compare the performance of Constrained-ε-Greedy to the unconstrained algorithm, the hypothetical *optimal* constrained algorithm (which we could implement if we knew the rewards of the arms a-priori), a smoothed version of the unconstrained algorithm that satisfies the constraints, and a naive baseline that satisfies the constraints but does not aim to optimize the reward.[4] We briefly outline the experiments and results here, with details in the following subsections.

We conduct counterfactual experiments on three datasets (see Table 2). We consider a curated PoliticalNews where the constraints aim to reduce the political polarization of the presented search results. As mentioned above, we can similarly apply these techniques to the diversification of content in areas beyond political polarization. Towards this, we simulate our algorithm on another dataset of news articles [41] and strive to diversify across topics (e.g., business, entertainment, and world news), and the MovieLens dataset [22] where we strive to diversify recommendations across genres. In all cases, wefi nd that Constrained-ε-Greedy consistently outperforms the smoothed version of the unconstrained algorithm as well as the naive baseline, accumulating much higher reward, while closely approximating the hypothetical optimal. This benefit of Constrained-ε-Greedy is most evident when the constraints are the tightest; e.g., Constrained-ε-Greedy accumulates twice as much reward as the smoothed version of the unconstrained algorithm on the YOW dataset (see Figure 2).

We then compare the polarization and diversification for the constrained and unconstrained algorithms. We aim to reduce polarization by recommending news articles with both, liberal and conservative biases. Similarly, we aim to increase diversity by recommending articles and movies not just from the *best group* in terms of rewards, but from other groups as well. We observe that algorithms in the unconstrained setting quickly converge to the *best group* in terms of rewards, whereas algorithms in the constrained setting always display a certain minimum percentage of content *not* from the best group, hence improving diversification and avoiding polarization.

5.1 Experimental Setup

5.1.1 Algorithm and Benchmarks. In each counterfactual simulation we report the normalized cumulative reward for each of the following algorithms and benchmarks:

Unconstrained-Optimal is the hypothetical *optimal* algorithm when there is no constraint and the expected rewards of all arms $a \in [k]$ are known. It simply chooses the best arm $a^\star$ at each step t.

Unconstrained-ε-Greedy is the *unconstrained* ε-Greedy algorithm, where C is the set of *all* probability distributions over $[k]$.

Constrained-Optimal is the hypothetical *optimal* probability distribution, subject to the polarization constraints, that we could have used if we had known the reward vector $\mu^\star$ for the arms a-priori.

Constrained-ε-Greedy is our implementation of Algorithm 1 with the given polarization constraints as input.[5]

Constrained-Ran is a smoothed version of Unconstrained-ε-Greedy that satisfies the constraints. At each time step, given the probability distribution p^t specified by the unconstrained ε-Greedy algorithm, Constrained-Ran takes the largest $\theta \in [0, 1]$ such that selecting an arm with probability $\theta \cdot p^t$ does not violate the constraints. With the remaining probability $(1 - \theta)$ it follows the same procedure as in Constrained-Naive to select an arm at *random* subject to the constraints.

Constrained-Naive. As a baseline, we consider a simple algorithm that satisfies the constraints as follows: for each group i and arm a, with probability $\frac{\ell_i}{w_a(G_i)}$ it selects an arm at random from G_i, then, with any remaining probability, it selects an arm uniformly at random from the entire collection $[k]$ while respecting the upper bound constraints u_i.

Note that if we know the true rewards of the arms, this optimal distribution is easy to compute via a simple greedy algorithm; it simply places the most probability mass that satisfies the constraints

[4] In our simulations, the regret for Constrained-L_1-OFUL was similar or slightly worse than Constrained-ε-Greedy. As Constrained-ε-Greedy is also much more efficient we use it as the main comparator, and leave open the question as to if or when Constrained-L_1-OFUL performs better as suggested by the theoretical results.

[5] We set $\varepsilon_t = \min(1, 10/t)$. Tuning ε_t could give even better results.

on the best arm, the most probability mass remaining on the second-best arm subject to the constraints, and so on and so forth until the entire probability mass is exhausted. This strategy can be found by solving one LP.

5.1.2 Implementation Details. Instead of solving an LP from scratch at each iteration (in step 4 of Algorithm 1), we warm start the LP by using the solution of the LP from the previous iteration as the starting point for our solver. We modified an implementation of an LP solver [40] which uses the interior points method. "Warm-starting" the LP solver in this way speeds up the LP computation considerably in practice and allows efficient implementation of the algorithm even when there are many groups that do not form a partition and hence many nontrivial constraints. For certain special cases, provably fast algorithms for solving the LP also exist (see Section 4.3.2), however we did not employ these techniques in the simulations.

Note that Constrained-ε-Greedy, Constrained-Ran and Unconstrained-ε-Greedy implementations all use Algorithm 1 as a subroutine; however Constrained-ε-Greedy and Constrained-Ran take the constraints as input, with Constrained-Ran satisfying the polarization constraints via smoothing the probability distribution, and Unconstrained-ε-Greedy need not satisfy the constraints at all.

5.1.3 Description of Datasets and Group Weights.

PoliticalNews. We curate this dataset by using a large scale web-crawler [2] to collect online news articles over a span of 30 days (23^{rd} July – 21^{st} August, 2018), along with the number of Facebook likes that each article received as of 22^{nd} August, 2018. We look at the political leaning of each article's publisher as determined by AllSides [1], which provides labels left, left-leaning, neutral, right-leaning or right for a wide set of publishers. We discard any articles that remain unlabelled or have fewer than 10 likes. This results in a dataset consisting of an average of 1356 articles each day, of which 15% are right, 7% are right-leaning, 31% are neutral, 34% are left-leaning and 13% are left. On average, the most-liked right article has 42, 293 likes, the most-liked right-leaning article has 144, 624 likes, the most-liked neutral article has 48, 647 likes, the most-liked left-leaning article has 117, 267 likes and the most-liked left article has 107, 497 likes. For each day, we encode each article as an arm with Bernoulli reward with mean proportional to the number of likes on Facebook (normalized to lie in the range $[0, 1]$).

We place a group weight of 0, 0.25, 0.5, 0.75 and 1 on right, right-leaning, neutral, left-leaning and left articles respectively for the *liberal* group (w(L)). Similarly, we place a group weight of 1, 0.75, 0.5, 0.25 and 0 on right, right-leaning, neutral, left-leaning and left articles respectively for the *conservative* group (w(C)).

MovieLens. We consider the MovieLens dataset [22], which consists of 100, 000 ratings from 943 users across 1, 682 movies; each user rated at least 20 movies on a scale of $1 - 5$. Each movie is also affiliated with one or more of 19 genres (e.g., sci-fi, romance, thriller). As some genres have significant overlap (e.g., thriller and horror), while others have different meanings at their intersections (e.g., romance vs rom-com vs comedy), wefi rst cluster the movies into different meta-categories based on their genres using a black-box k-means clustering algorithm with $k = 25$ [30].[6] We use the cluster centres as representative *arms*, and associate all movies in that cluster to that arm. For a given user, the reward associated with an arm is given by a Gaussian where the mean is the average rating the user gave to movies associated with the arm, and standard deviation $\sigma = 0.1$.

For a genre i ($i \in [19]$) and movie category a, the group weight $w_a(G_i)$ is set to be the i^{th} coordinate of the cluster centre of movie category a found by the k-means clustering.

YOW. We consider the YOW dataset [41] which contains data from a collection of 24 paid users who read 5921 unique articles over a 4 week time period. The dataset contains the time at which each user read an article, a [0-5] rating for each article read by each user, and (optional) user-generated categories of articles viewed. We use this data to construct reward distributions for each user on a set of arms that one can expect to see from the real world.

We create a simple ontology to categorize the 10010 user-generated labels into a total of $g = 7$ groups of content: Science, Entertainment, Business, World, Politics, Sports, and USA. On average there are $k = 81$ unique articles in a day. We take this to be the number of *arms* in this experiment. Similar to the MovieLens experiments, we cluster the articles into 81 arms based on the news categories they belong to, using k-means clustering ($k = 81$). We use the cluster centres as representative *arms*, and associate all articles in that cluster to that arm. For a given user, the reward associated with that arm is given by a Gaussian where the mean is the average rating the user gave to articles associated with the arm, and standard deviation $\sigma = 0.1$.

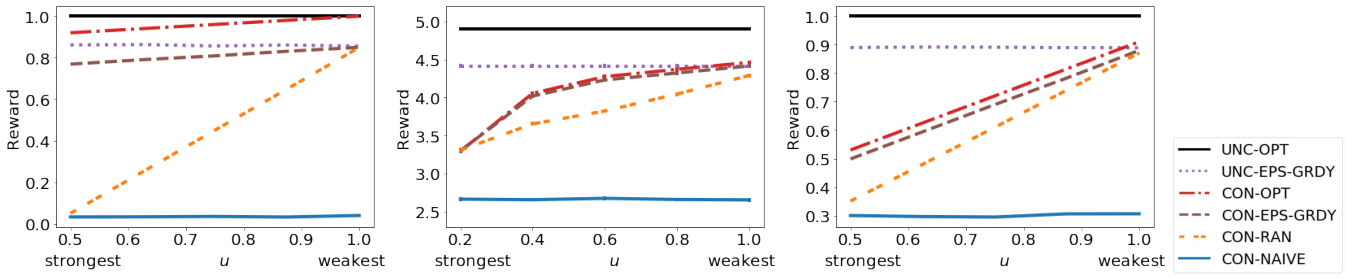
For a news category i ($i \in [7]$) and article a, the group weight $w_a(G_i)$ is set to be the i^{th} coordinate of the cluster centre found by $k-$means clustering.

5.2 Effect of Reducing Polarization on the Reward

We vary the tightness of upper bound constraints on the probability mass of displaying arms of a given group, and report the normalized cumulative reward.

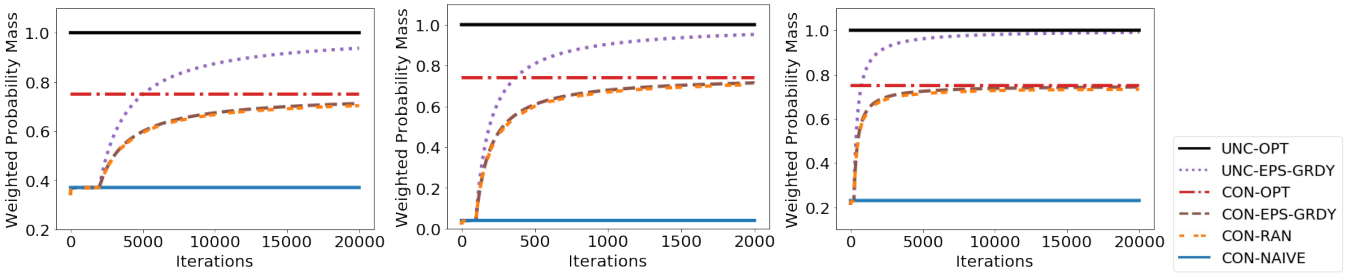
5.2.1 PoliticalNews. For this dataset, there are only two groups: either left or right. However, a news article may have weight on both groups, and it is these weights that determine how right- or left-leaning an article is, and hence how much they contribute towards polarization in a given direction. We simulated each of the 30 days separately, resulting in $n = 30$ datapoints. We report the normalized cumulative reward after $T = 10,000$ iterations, averaged over experiments from all 30 days. As there are only two groups, setting a lower bound constraint $\ell_1 = \zeta$ is equivalent to setting an upper bound constraint $u_2 = 1 - \zeta$. Hence, it suffices to see the effect as we vary the upper bounds. We vary $u_1 = u_2 = u$ from 0.5 to 1; i.e., from a fully constrained one in which each group has exactly 50% weighted probability of being selected to a completely unconstrained setting. We observe in Figure 2a that, even for very large values of u (i.e., when the constraints are loose), the Constrained-ε-Greedy algorithm significantly outperforms

[6] We determine $k = 25$ using the graph of silhouette values [33] vs. k.



(a) *PoliticalNews dataset.* (b) *MovieLens dataset.* (c) YOW dataset.

Figure 2: Effect of Polarization Constraints (u) on Reward. The normalized cumulative reward attained as a function of the strength of the upper-bound constraints is reported for the three datasets infi gures (a), (b) and (c). In all cases, our algorithm Constrained-ε-Greedy does not allow polarization, and performs near-optimally with respect to the reward. The lower the value of u, the stronger are the constraints.



(a) *PoliticalNews.* (b) *MovieLens.* (c) *YOW.*

Figure 3: Visualizing Polarization and Diversification. The weighted probability mass on the *best group* is reported against the number of iterations. While the unconstrained algorithm converges quickly to placing all of its probability mass on the optimal group, the constrained algorithm – by definition – maintains some weight on the non-optimal groups. This is what ensures diversification across content and avoids polarization.

Constrained-Ran with respect to regret, and is only worse than the unconstrained (and hence polarized) algorithm by an additive factor of approximately $\frac{1-u}{5}$ (i.e., less than 10%).

5.2.2 MovieLens. For this dataset, a group corresponds to a *genre*. Note that a movie can belong to multiple genres with varying weights which may not add up to one. We report the normalized cumulative reward averaged across all 943 users after $T = 1000$ iterations. Error bars depict the standard error of the mean. We observe in Figure 2b that Constrained-ε-Greedy significantly outperforms the Constrained-Naive and Constrained-Ran algorithms across constraints. Additionally, as there are fewer arms in the MovieLens dataset as compared to the PoliticalNews dataset, the learning cost is lower and hence the Constrained-ε-Greedy performs essentially as well as the (unattainable) Constrained-Optimal algorithm.

5.2.3 YOW. For this dataset, a group corresponds to an *article category*. Note that an article can belong to multiple categories (e.g., science and business) simultaneously, with varying weights across each category. We report the normalized cumulative reward averaged across all 21 users after $T = 10,000$ iterations. Error bars depict the standard error of the mean. As before, we observe in Figure 2c that Constrained-ε-Greedy significantly outperforms the Constrained-Naive and Constrained-Ran algorithms across constraints, and performs almost as well as the (unattainable) Constrained-Optimal algorithm.

5.3 Polarization Over Time

In order to see how polarization can be avoided and diversification can be enforced using our framework, for each dataset we plot the normalized cumulative weighted probability mass on the *best group* for each datapoint against the number of iterations, with the $u = 0.75$. Initially, the unconstrained and constrained algorithms have the same weighted probability mass for the best group, because the algorithms are simply exploring the arms. However, the difference between the algorithms becomes very apparent once the algorithm begin to learn. Due to a larger number of arms in the PoliticalNews dataset, this process takes longer as compared to the other two datasets. Unconstrained-ε-Greedy quickly polarizes almost-entirely to display content only from the best group. This depicts the necessity for such constraints. However, Constrained-ε-Greedy maintains at least $1-u$ of its weighted probability mass on content not belonging to the best group, increasing diversification and avoiding polarization.

6 CONCLUSION

In this paper we initiate a formal study of combating polarization in personalization algorithms that learn user behavior. We present a general framework that allows one to prevent polarization by ensuring that a balanced set of items are displayed to each user. We show how one can modify a simple bandit algorithm in order to perform well with respect to regret subject to satisfying the polarization constraints, improving the regret bound over the state-of-the-art. Empirically, we observe that the Constrained-ε-Greedy algorithm performs well; it not only converges quickly to the theoretical optimum, but this optimum, even for the tightest constraints on the arm values selected ($u = 0.2$ for MovieLens, $u = 0.5$ for PoliticalNews), is within a factor of 2 of the unconstrained rewards. Furthermore, Constrained-ε-Greedy is fast and we expect it to scale well in web-level applications.

With regard to future work, a limitation of our algorithms is the fact that they assume we are given the group labels and weights for each piece of content. These labels would either need to be inferred from the data, which could bring with it additional bias associated with this learning algorithm, or would need to be self-reported, which can lead to adversarial manipulation. Additionally, it would be important to extend this work to a dynamic setting in which the type of content changes over time, e.g., using restless bandit techniques. From an experimental standpoint, testing this algorithm in thefi eld, in particular to measure user satisfaction given diversified news feeds, would be of significant interest. Such an experiment would give deeper insight into the benefits and tradeoffs between personalization and the diversification of content, which could then be leveraged to determine which kind of constraints can prevent polarization not just of the items in the feed, but of the beliefs and opinions of those viewing them.

REFERENCES

[1] [n. d.]. AllSides Media Bias Ratings. https://www.allsides.com/media-bias/media-bias-ratings.

[2] [n. d.]. Webhose News API. https://webhose.io/data-feeds/news-api/.

[3] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances In Neural Information Processing Systems*.

[4] Gediminas Adomavicius, Jesse Bockstedt, Curley Shawn, and Jingjing Zhang. 2014. De-biasing user preference ratings in recommender systems. In *Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, Co-located with ACM Conference on Recommender Systems*.

[5] Gediminas Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 896–911.

[6] Shipra Agrawal and Nikhil Devanur. 2016. Linear Contextual Bandits with Knapsacks. In *Advances In Neural Information Processing Systems*. 3450–3458.

[7] Shipra Agrawal and Navin Goyal. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*. 39–1.

[8] Rubayyi Alghamdi and Khalid Alfalqi. 2015. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)* 6, 1 (2015).

[9] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[10] Drake Baer. 2016. The 'Filter Bubble' Explains Why Trump Won and You Didn't See It Coming. NY Mag.

[11] Engin Bozdag and Jeroen van den Hoven. 2015. Breaking thefi lter bubble: democracy and design. *Ethics and Information Technology* 17, 4 (01 Dec 2015), 249–265.

[12] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi. 2018. Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees. *ArXiv e-prints* (June 2018). arXiv:1806.06055

[13] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. 2011. Political polarization on twitter. *ICWSM* 133 (2011), 89–96.

[14] Varsha Dani, Thomas P Hayes, and Sham M Kakade. 2008. Stochastic Linear Optimization under Bandit Feedback. In *Proceedings of the Annual Conference on Learning Theory (COLT)*.

[15] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies* 2015, 1 (2015), 92–112.

[16] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. 2013. Multi-Armed Bandit with Budget Constraint and Variable Costs.. In *AAAI*.

[17] Robert Epstein and Ronald E Robertson. 2015. The search engine manipulation effect (SEME) and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences* 112, 33 (2015), E4512–E4521.

[18] Ayman Farahat and Michael C Bailey. 2012. How effective is targeted advertising?. In *Proceedings of the 21st international conference on World Wide Web*. ACM.

[19] Thomas Fox-Brewster. 2017. Creepy Or Cool? Twitter Is Tracking Where You've Been, What You Like And Is Telling Advertisers. Forbes Magazine.

[20] Venkata Rama Kiran Garimella and Ingmar Weber. 2017. A Long-Term Analysis of Polarization on Twitter. In *ICWSM*.

[21] Avi Goldfarb and Catherine Tucker. 2011. Online display advertising: Targeting and obtrusiveness. *Marketing Science* (2011).

[22] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[23] Sounman Hong and Sun Hyoung Kim. 2016. Political polarization on twitter: Implications for the use of social media in digital governments. *Government Information Quarterly* 33, 4 (2016), 777–782.

[24] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. 2016. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*. 325–333.

[25] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[26] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 539–548.

[27] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM.

[28] Sandeep Pandey and Christopher Olston. 2006. Handling Advertisements of Unknown Quality in Search Advertising. In *Advances in Neural Information Processing Systems*.

[29] Eli Pariser. 2011. *Thefi lter bubble: What the Internet is hiding from you*. Penguin UK.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[31] Filipe N. Ribeiro, Lucas Henrique, Fabricio Benevenuto, Abhijnan Chakraborty, Juhi Kulshrestha, Mahmoudreza Babaei, and Krishna P. Gummadi. 2018. Media Bias Monitor: Quantifying Biases of Social Media News Outlets at Scale. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*.

[32] Ronald E. Robertson, David Lazer, and Christo Wilson. 2018. Auditing the Personalization and Composition of Politically-Related Search Engine Results Pages. In *Proceedings of the 2018 World Wide Web Conference*.

[33] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* (1987).

[34] Pranav Sakulkar and Bhaskar Krishnamachari. 2016. Stochastic contextual bandits with known reward functions. *arXiv preprint arXiv:1605.00176* (2016).

[35] Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P. Gummadi, Patrick Loiseau, and Alan Mislove. 2018. Potential for Discrimination in Online Targeted Advertising. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. PMLR.

[36] Karthik Sridharan. 2002. A gentle introduction to concentration inequalities. (2002).

[37] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[38] Jacek Wasilewski and Neil Hurley. 2016. Incorporating Diversity in a Learning to Rank Recommender System.. In *FLAIRS Conference*. 572–578.

[39] Ingmar Weber, Venkata R Kiran Garimella, and Alaa Batayneh. 2013. Secular vs. islamist polarization in egypt on twitter. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM.

[40] Yiming Yan. [n. d.]. Mehrotra's Predictor-Corrector Interior Point Method. https://github.com/YimingYAN/mpc.

[41] Yi Zhang. 2005. Bayesian Graphical Models For Adaptive Filtering. In *PhD Thesis*.