Adaptive Submodular Maximization in Bandit Setting

Authored by:

Branislav Kveton
Victor Gabillon
Zheng Wen
Brian Eriksson
S. Muthukrishnan

Abstract

Maximization of submodular functions has wide applications in machine learning and artificial intelligence. Adaptive submodular maximization has been traditionally studied under the assumption that the model of the world, the expected gain of choosing an item given previously selected items and their states, is known. In this paper, we study the scenario where the expected gain is initially unknown and it is learned by interacting repeatedly with the optimized function. We propose an efficient algorithm for solving our problem and prove that its expected cumulative regret increases logarithmically with time. Our regret bound captures the inherent property of submodular maximization, earlier mistakes are more costly than later ones. We refer to our approach as Optimistic Adaptive Submodular Maximization (OASM) because it trades off exploration and exploitation based on the optimism in the face of uncertainty principle. We evaluate our method on a preference elicitation problem and show that non-trivial K-step policies can be learned from just a few hundred interactions with the problem.

1 Paper Body

Maximization of submodular functions [14] has wide applications in machine learning and artificial intelligence, such as social network analysis [9], sensor placement [10], and recommender systems [7, 2]. In this paper, we study the problem of adaptive submodular maximization [5]. This problem is a variant of submodular maximization where each item has a state and this state is revealed when the item is chosen. The goal is to learn a policy that maximizes the expected return for choosing K items. Adaptive submodular maximization has been traditionally studied in the setting where the model of the world, the expected gain of choosing an item given previously selected items and their states, is known. This is the first paper that studies the setting where the model is initially unknown, and it is learned by interacting repeatedly with the environment. We bring together the concepts of adaptive submodular maximization and bandits, and the result is an efficient solution to our problem. We make four major contributions. First, we propose a model where the expected gain of choosing an item can be learned efficiently. The main assumption in the model is that the state of each item is distributed independently of the other states. Second, we propose Optimistic Adaptive Submodular Maximization (OASM), a bandit algorithm that selects items with the highest upper confidence bound on the expected gain. This algorithm is computationally efficient and easy to implement. Third, we prove that the expected cumulative regret of our algorithm increases logarithmically with time. Our regret bound captures the inherent property of adaptive submodular maximization, earlier mistakes are more costly than later ones. Finally, we apply our approach to a real-world preference elicitation 1

problem and show that non-trivial policies can be learned from just a few hundred interactions with the problem.

2

Adaptive Submodularity

In adaptive submodular maximization, the objective is to maximize, under constraints, a function of the form: L f : 2I ? {?1, 1} ? R, (1) where I = {1, . . . , L} is a set of L items and 2I is its power set. The first argument of f is a subset L of chosen items A ? I. The second argument is the state ? ? {?1, 1} of all items. The i-th entry of ?, ?[i], is the state of item i. The state ? is drawn i.i.d. from some probability distribution P (?). The reward for choosing items A in state ? is f (A, ?). For simplicity of exposition, we assume that f (?, ?) = 0 in all ?. In problems of our interest, the state is only partially observed. To capture this L phenomenon, we introduce the notion of observations. An observation is a vector y ? {?1, 0, 1} whose non-zero entries are the observed states of items. We say that y is an observation of state ?, and write ? ? y, if y[i] = ?[i] in all non-zero entries of y. Alternatively, the state ? can be viewed as a realization of y, one of many. We denote by dom (y) = {i : y[i] 6= 0} the observed items in y and by ?hAi the observation of items A in state ?. We define a partial ordering on observations and write y0 y if y0 [i] = y[i] in all non-zero entries of y, y0 is a more specific observation than y. In the terminology of Golovin and Krause [5], y is a subrealization of y0 . We illustrate our notation on a simple example. Let ? = (1, 1, ?1) be a state, and y1 = (1, 0, 0) and y2 = (1, 0, ?1) be observations. Then all of the following claims are true: ? ? y1 , ? ? y2 , y2 y1 , dom (y2) = {1, 3} , ?h{1, 3}i = y2 , ?hdom (y1)i = y1 . Our goal is to maximize the expected value of f by adaptively choosing K items. This problem can be viewed as a K step game, where at each step we choose an item according to some policy ? and L then observe its state. A policy ? : {?1, 0, 1} ? I is a function from observations y to items. The observations represent our past decisions and their outcomes. A k-step policy in state ?, ?k (?), is a collection of the first k items chosen by policy ?. The policy is defined recursively as:

?k (?) = ?k?1 (?) ? ?[k] (?) , ?[k] (?) = ?(?h?k?1 (?)i), ?0 (?) = ?, (2) where ?[k] (?) is the k-th item chosen by policy ? in state ?. The optimal K-step policy satisfies: ? ? = arg max? E? [f (?K (?), ?)] . (3) In general, the problem of computing ? ? is NP-hard [14, 5]. However, near-optimal policies can be computed efficiently when the maximized function has a diminishing return property. Formally, we require that the function is adaptive submodular and adaptive monotonic [5]. Definition 1. Function f is adaptive submodular if: E? [f (A ? {i} , ?) ? f (A, ?) — ? ? yA] ? E? [f (B ? {i} , ?) ? f (B, ?) — ? ? yB] for all items i ? I B and observations yB yA , where A = dom (yA) and B = dom (yB). Definition 2. Function f is adaptive monotonic if E? [f (A ? {i} , ?) ? f (A, ?) — ? ? yA] ? 0 for all items i ? I A and observations yA , where A = dom (yA). In other words, the expected gain of choosing an item is always non-negative and does not increase as the observations become more specific. Let ? g be the greedy policy for maximizing f , a policy that always selects the item with the highest expected gain: ? g (y) = arg max gi (y), (4) i?Idom(y)

where: gi (y) = E? [f (dom (y) ? {i} , ?) ? f (dom (y) , ?) — ? ? y] (5) is the expected gain of choosing item i after observing y. Then, based on the result of Golovin and g ? Krause [5], ? g is a (1 ? 1/e)-approximation to ? ? , E? [f (?K (?), ?)] ? (1 ? 1/e)E? [f (?K (?), ?)], if f is adaptive submodular and adaptive monotonic. In the rest of this paper, we say that an observation y is a context if it can be observed under the greedy policy ? g . Specifically, there exist k and ? such that y = ?h?kg (?)i. 2

3

Adaptive Submodularity in Bandit Setting

The greedy policy ? g can be computed only if the objective function f and the distribution of states P (?) are known, because both of these quantities are needed to compute the marginal benefit gi (y) (Equation 5). In practice, the distribution P (?) is often unknown, for instance in a newly deployed sensor network where the failure rates of the sensors are unknown. In this paper, we study a natural variant of adaptive submodular maximization that can model such problems. The distribution P (?) is assumed to be unknown and we learn it by interacting repeatedly with the problem. 3.1

Model

The problem of learning P (?) can be cast in many ways. One approach is to directly learn the joint P (?). This approach is not practical for two reasons. First, the number of states ? is exponential in the number of items L. Second, the state of our problem is observed only partially. As a result, it is generally impossible to identify the distribution that generates ?. Another possibility is to learn the probability of individual states ?[i] conditioned on context, observations y under the greedy policy ? g in up to K steps. This is impractical because the number of contexts is exponential in K. Clearly, additional structural assumptions are necessary to obtain a practical solution. In this paper, we assume that the states of items are independent of the context in which the items are chosen. In particular, the state ?[i] of each item i is drawn i.i.d. from a Bernoulli distribution with mean pi . In this setting, the joint probability distribution factors as: P (? = ?) =

L Y
1{?[i]=1}
pi
(1 ? pi)1?1{?[i]=1}
(6)
i=1

and the problem of learning P (?) reduces to estimating L parameters, the means of the Bernoullis. A major question is how restrictive is our independence assumption. We argue that this assumption is fairly natural in many applications. For instance, consider a sensor network where the sensors fail at random due to manufacturing defects. The failures of these sensors are independent of each other and thus can be modeled in our framework. To validate our assumption, we conduct an experiment (Section 4) that shows that it does not greatly affect the performance of our method on a real-world problem. Correlations obviously exist and we discuss how to model them in Section 6. Based on the independence assumption, we rewrite the expected gain (Equation 5) as: gi (y) = pi g?i (y),

(7)

g?i (y) = E? [f (dom (y) ? {i} , ?) ? f (dom (y) , ?) — ? ? y, ?[i] = 1]

(8)

where:

is the expected gain when item i is in state 1. For simplicity of exposition, we assume that the gain is zero when the item is in state ?1. We discuss how to relax this assumption in Appendix. In general, the gain g?i (y) depends on P (?) and thus cannot be computed when P (?) is unknown. In this paper, we assume that g?i (y) can be computed without knowing P (?). This scenario is quite common in practice. In maximum coverage problems, for instance, it is quite reasonable to assume that the covered area is only a function of the chosen items and their states. In other words, the gain can be computed as g?i (y) = f (dom (y) ? {i} , ?) ? f (dom (y) , ?), where ? is any state such that ? ? y and ?[i] = 1. Our learning problem comprises n episodes. In episode t, we adaptively choose K items according to some policy ? t , which may differ from episode to P episode. The quality of the policy is measured n t by the expected cumulative K-step return E?1 ,...,?n [t=1 f (?K (?t), ?t)]. We compare this return g to that of the greedy policy ? and measure the difference between the two returns by the expected cumulative regret: " n # " n # X X g t R(n) = E?1 ,...,?n Rt (?t) = E?1 ,...,?n f (?K (?t), ?t) ? f (?K (?t), ?t) . (9) t=1

t=1

In maximum coverage problems, the greedy policy ? is a good surrogate for the optimal policy ? ? because it is a (1 ? 1/e)-approximation to ? ? (Section 2). g

3

Algorithm 1 OASM: Optimistic adaptive submodular maximization. Input: States ?1 , . . . , ?n for all i ? I do Select item i and set p?i,1 to its state, Ti (0) ? 1 end for . Initialization for all t = 1, 2, . . . , n do A?? for all k = 1, 2, . . . , K do . K-step maximization y ? ?t hAi( ) A?A?

arg max (? pi,Ti (t?1) + ct?1,Ti (t?1))? gi (y)

. Choose the highest index

i?IA

end for for all i ? I do Ti (t) ? Ti (t ? 1) end for for all i ? A do Ti (t) ? Ti (t) + 1 p?i,Ti (t) ? Ti1(t) (? pi,Ti (t?1) Ti (t ? 1) + 21 (?t [i] + 1)) end for end for 3.2

. Update statistics

Algorithm

Our algorithm is designed based on the optimism in the face of uncertainty principle, a strategy that is at the core of many bandit algorithms [1, 8, 13]. More specifically, it is a greedy policy where the expected gain gi (y) (Equation 7) is substituted for its optimistic estimate. The algorithm adaptively maximizes a submodular function in an optimistic fashion and therefore we refer to it as Optimistic Adaptive Submodular Maximization (OASM). The pseudocode of our method is given in Algorithm 1. In each episode, we maximize the function f in K steps. At each step, we compute the index (? pi,Ti (t?1) + ct?1,Ti (t?1))? gi (y) of each item that has not been selected yet and then choose the item with the highest index. The terms p?i,Ti (t?1) and ct?1,Ti (t?1) are the maximum-likelihood estimate of the probability pi from the first t ? 1 episodes and the radius of the confidence interval around this estimate, respectively. Formally: r s 1X1 2 log(t) p?i,s = (?? (i,z) [i] + 1), ct,s = , (10) s z=1 2 s where s is the number of times that item i is chosen and ? (i, z) is the index of the episode in which item i is chosen for the z-th time. In episode t, we set s to Ti (t ? 1), the number of times that item i is selected in the first t ? 1 episodes. The radius ct,s is designed such that each index is with high probability an upper bound on the corresponding gain. The index enforces exploration of items that have not been chosen very often. As the number of past episodes increases, all confidence intervals shrink and our method starts exploiting most profitable items. The log(t) term guarantees that each item is explored infinitely often as t ? ?, to avoid linear regret. Algorithm OASM has several notable properties. First, it is a greedy method. Therefore, our policies can be computed very fast. Second, it is guaranteed to behave near optimally as our estimates of the gain gi (y) become more accurate. We prove this claim in Section 3.3. Finally, our algorithm learns only L parameters and therefore is quite practical. Specifically, note that if an item is chosen in one context, it helps in refining the estimate of the gain gi (y) in all other contexts. 3.3

Analysis

In this section, we prove an upper bound on the expected cumulative regret of Algorithm OASM in n episodes. Before we present the main result, we define notation used in our analysis. We denote by i? (y) = ? g (y) the item chosen by the greedy policy ? g in context y. Without loss of generality, we assume that this item is unique in all contexts. The hardness of discriminating between items i and i? (y) is measured by a gap between the expected gains of the items: ?i (y) = gi? (y) (y) ? gi (y). (11) Our analysis is based on counting how many times the policies ? t and ? g choose a different item at step k. Therefore, we define several variables that describe the state of our problem at this step. We

S denote by Yk (?) = ? {?h?k?1 (?)i} the set of all possible observations after policy ? is executed for k ? 1 steps. We write Yk = Yk (? g) and Ykt = Yk (? t) when we refer to the policies ? g and ? t , respectively. Finally, we denote by Yk,i = Yk ? {y : i 6= i? (y)} the set of contexts where item i is suboptimal at step k. Our main result is Theorem 1. Supplementary material for its proof is in Appendix. The terms item and arm are treated as synonyms, and we use whichever is more appropriate in a given context. Theorem 1. The expected cumulative regret of Algorithm OASM is bounded as: L K K X X X 2 R(n) ? ‘i Gk ?i,k + ? 2 L(L + 1) Gk , 3 i=1 k=1 k=1 {z } — {z } — O(log n)

(12)

O(1)

where Gk = (K ? k + 1) max max gi (y) is an upper bound on the expected gain of the policy ? g y?Yk i

g ?2 (y) from step k forward, ‘i,k = 8 max ?i2 (y) log n is the number of pulls after which arm i is not y?Yk,i

i

+ 1 likely to be pulled suboptimally at step k, ‘i = max ‘i,k , and ?i,k = ‘i,k ? max ‘i,k0 ? [0, 1] 0 k k ¡k PK ‘i is a weight that associates the regret of arm i to step k such that k=1 ?i,k = 1. Proof. Our theorem is proved in three steps. First, we associate the regret in episode t with the first step where our policy ? t selects a different item from the greedy policy ? g . For simplicity, suppose that this step is step k. Then the regret in episode t can be written as: g t Rt (?t) = f (?K (?t), ?t) ? f (?K (?t), ?t) g g t t = f (?K (?t), ?t) ? f (?k?1 (?t), ?t) ?[f (?K (?t), ?t) ? f (?k?1 (?t), ?t)], {z } — {z } — g Fk? (?t)

(13)

t (?) Fk? t

g g t where the last equality is due to the assumption that ?[j] (?t) = ?[j] (?t) for all j ¡ k; and Fk? (?t) t g t and Fk? (?t) are the gains of the policies ? and ? , respectively, in state ?t from step k forward. In practice, the first step where the policies ? t and ? g choose a different item is unknown, because ? g is unknown. In this case, the regret can be written as:

Rt (?t) =

K L X X

g t 1i,k,t (?t)(Fk? (?t) ? Fk? (?t)),

(14)

i=1 k=1

where: n

o g g t t t 1i,k,t (?) = 1 ?j ¡ k : ?[j] (?) = ?[j] (?) , ?[k] (?) 6= ?[k] (?), ?[k] (?) = i

(15)

is the indicator of the event that the policies ? t and ? g choose the same first k ? 1 items in state ?, disagree in the k-th item, and i is the k-th item chosen by ? t . The commas in the indicator function represent logical conjunction. Second, in Lemma 1 we bound the expected loss associated with choosing the first different item at step k by the probability of this event and an upper bound on the expected loss Gk , which does not depend on ? t and ?t . Based on this result, we bound the expected cumulative regret as: ” n # ” n L K # X XXX g t E?1 ,...,?n Rt (?t) = E?1 ,...,?n 1i,k,t (?t)(Fk? (?t) ? Fk? (?t)) t=1

t=1 i=1 k=1

=

L X K X n X

g t E?1 ,...,?t?1 E?t 1i,k,t (?t)(Fk? (?t) ? Fk? (?t))

i=1 k=1 t=1

?

L X K X n X

E?1 ,...,?t?1 [E?t [1i,k,t (?t)] Gk]

i=1 k=1 t=1

=

L X K X

Gk E?1 ,...,?n

” n X t=1

i=1 k=1

5

1i,k,t (?t) .

(16)

Finally, motivated by the analysis of UCB1 [1], we rewrite the indicator 1i,k,t (?t) as: 1i,k,t (?t) = 1i,k,t (?t)1{Ti (t ? 1) ? ‘i,k } + 1i,k,t (?t)1{Ti (t ? 1) ¿ ‘i,k } ,

(17)

where ‘i,k is a problem-specific constant. In Lemma 4, we show how to choose ‘i,k such that arm i at step k is pulled suboptimally a constant number of times in expectation after ‘i,k pulls. Based on this result, the regret corresponding to the events 1{Ti (t ? 1) ¿ ‘i,k } is bounded as: ” n # L X K K X X X 2 Gk . (18) Gk E?1 ,...,?n 1i,k,t (?t)1{Ti (t ? 1) ¿ ‘i,k } ? ? 2 L(L + 1) 3 t=1 i=1 k=1

k=1

On the other hand, the regret associated with the events 1{Ti (t ? 1) ? ‘i,k } is trivially bounded by PL PK i=1 k=1 Gk ‘i,k . A tighter upper bound is proved below: # ”K L n X X X E?1 ,...,?n Gk 1i,k,t (?t)1{Ti (t ? 1) ? ‘i,k } i=1

?

i=1

?

t=1

k=1 L X

” max

?1 ,...,?n

L X K X

K X k=1

Gk

n X

# 1i,k,t (?t)1{Ti (t ? 1) ? ‘i,k }
t=1
+ 0 Gk ‘i,k ? max ‘ . i,k 0
i=1 k=1
k ¡k
(19)

The last inequality can be proved as follows. Our upper bound on the expected loss at step k, Gk , is monotonically decreasing with k, and therefore G1 ? G2 ? . . . ? GK . So for any given arm i, the highest cumulative regret subject to the constraint Ti (t ? 1) ? ‘i,k at step k is achieved as follows. + The first ‘i,1 mistakes are made at the first step, [‘i,2 ? ‘i,1] mistakes are made at the second step, + [‘i,3 ? max{‘i,1 , ‘i,2 }] mistakes are made at the third step, and so on. Specifically, the number of + mistakes at step k is [‘i,k ? maxk0 ¡k ‘i,k0] and the associated loss is Gk . Our main claim follows from combining the upper bounds in Equations 18 and 19. 3.4

Discussion of Theoretical Results

Algorithm OASM mimics the greedy policy ? g . Therefore, we decided to prove Theorem 1 based on counting how many times the policies ? t and ? g choose a different item. Our proof has three parts. First, we associate the regret in episode t with the first step where the policy ? t chooses a different item from ? g . Second, we bound the expected regret in each episode by the probability of deviating from the policy ? g at step k and an upper bound on the associated loss Gk , which depends only on k. Finally, we divide the expected cumulative regret into two terms, before and after item i at step k is selected a sufficient number of times ‘i,k , and then set ‘i,k such that both terms are O(log n). We would like to stress that our proof is relatively general. Our modeling assumptions (Section 3.1) are leveraged only in Lemma 4. In the rest of the proof, we only assume that f is adaptive submodular and adaptive monotonic. Our regret bound has several notable properties. First, it is logarithmic in the number of episodes n, through problem-specific constants ‘i,k . So we recover a classical result from the bandit literature. Second, the bound is polynomial in all constants of interest, such as the number of items L and the number of maximization steps K in each episode. We would like to stress that it is not linear in the number of contexts YK at step K, which is exponential in K. Finally, note that our bound captures the shape of the optimized function f . In particular, because the function f is adaptive submodular, the upper bound on the gain of the policy ? g from step k forward, Gk , decreases as k increases. As a result, earlier deviations from ? g are penalized more than later ones.

4

Experiments

Our algorithm is evaluated on a preference elicitation problem in a movie recommendation domain. This problem is cast as asking K yes-or-no movie-genre questions. The users and their preferences are extracted from the MovieLens dataset [11], a dataset of 6k users who rated one million movies. 6

30 Covered movies [%]

gi (0) g?i (0) P (?[i] = 1) 4.1% 13.0% 0.32 4.1% 9.2% 0.44 3.2% 6.6% 0.48 3.0% 8.0% 0.38 2.8% 23.0% 0.12 2.6% 6.0% 0.44 2.6% 5.8% 0.44 2.3% 19.6% 0.12

Genre Crime Children?s Animation Horror Sci-Fi Musical Fantasy Adventure

20 g

?

g d

10

Deterministic ? g f

Factored ? 0

2

4

6 8 10 12 14 Number of questions K

16

18

Figure 1: Left. Eight movie genres that cover the largest number of movies in expectation. Right. Comparison of three greedy policies for solving our preference elicitation problem. For each policy and K ? L, we report the expected percentage of covered movies after K questions. K=2

K=4

Covered movies [%]

10

K=8

15

25

8

20

10

6

15 4

5

10

2 0 1 10

2

10

3

4

10 10 Episode t

5

10

0 1 10

2

10

3

4

10 10 Episode t
5
10
5 1 10
2
10
3
4
10 10 Episode t
5
10

Figure 2: The expected return of the OASM policy ? t (cyan lines) in all episodes up to t = 105 . The return is compared to those of the greedy policies ? g (blue lines), ?fg (red lines), and ?dg (gray lines) in the offline setting (Figure 1) at the same operating point, the number of asked questions K. We choose 500 most rated movies from the dataset. Each movie l is represented by a feature vector xl such that xl [i] = 1 if the movie belongs to genre i and xl [i] = 0 if it does not. The preference of user j for genre i is measured by tf-idf, a popular importance score in information retrieval [12]. In

nu particular, it is defined as tf-idf(j, i) = #(j, i) log #(?,i) , where #(j, i) is the number of movies from genre i rated by user j, nu is the number of users, and #(?, i) is the number of users that rated at least one movie from genre i. Intuitively, this score prefers genres that are often rated by the user but rarely rated overall. Each user j is represented by a genre preference vector ? such that ?[i] = 1 when genre i is among five most favorite genres of the user. These genres cover on average 25% of our movies. In Figure 1, we show several popular genres from our dataset. The reward for asking user ? questions A is: P500 f (A, ?) = 15 l=1 maxi [xl [i]1{?[i] = 1} 1{i ? A}] ,

(20)

the percentage of movies that belong to at least one genre i that is preferred by the user and queried in A. The function f captures the notion that knowing more preferred genres is better than knowing less. It is submodular in A for any given preference vector ?, and therefore adaptive submodular in A when the preferences are distributed independently of each other (Equation 6). In this setting, the expected value of f can be maximized near optimally by a greedy policy (Equation 4). In the first experiment, we show that our assumption on P (?) (Equation 6) is not very restrictive in our domain. We compare three greedy policies for maximizing f that know P (?) and differ in how the expected gain of choosing items is estimated. The first policy ? g makes no assumption on P (?) and computes the gain according to Equation 5. The second policy ?fg assumes that the distribution P (?) is factored and computes the gain using Equation 7. Finally, the third policy ?dg computes the gain according to Equation 8, essentially ignoring the stochasticity of our problem. All policies are applied to all users in our dataset for all K ? L and their expected returns are reported in Figure 1. We observe two trends. First, the policy ?fg usually outperforms the policy ?dg by a large margin. So although our independence assumption may be incorrect, it is a better approximation than ignoring 7

the stochastic nature of the problem. Second, the expected return of ?fg is always within 84% of ? g . We conclude that ?fg is a good approximation to ? g . In the second experiment, we study how the OASM policy ? t improves over time. In each episode t, we randomly choose a new user ?t and then the policy ? t asks K questions. The expected return of ? t is compared to two offline baselines, ?fg and ?dg . The policies ?fg and ?dg can be viewed as upper and lower bounds on the expected return of ? t , respectively. Our results are shown in Figure 2. We observe two major trends. First, ? t easily outperforms the baseline ?dg that ignores the stochasticity of our problem. In two cases, this happens in less than ten episodes. Second, the expected return of ? t approaches that of ?fg , as is expected based on our analysis.

5

Related Work

Our paper is motivated by prior work in the areas of submodularity [14, 5] and bandits [1]. Similar problems to ours were studied by several authors. For instance, Yue and Guestrin [17], and Guillory and Bilmes [6], applied bandits to submodular problems in a non-adaptive setting. In our work, we focus on the adaptive setting. This setting is more challenging because we learn a K-step policy for choosing items, as opposing to a single set of items. Wen et al. [16] studied a variant of generalized binary search, sequential Bayesian search, where the policy for asking questions is learned on-thefly by interacting with the environment. A major observation of Wen et al. [16] is that this problem can be solved near optimally without exploring. As a result, its solution and analysis are completely different from those in our paper. Learning with trees was studied in machine learning in many settings, such as online learning with tree experts [3]. This work is similar to ours only in trying to learn a tree. The notions of regret and the assumptions on solved problems are completely different. Optimism in the face of uncertainty is a popular approach to designing learning algorithms, and it was previously applied to more general problems than ours, such as planning [13] and MDPs [8]. Both of these solutions are impractical in our setting. The former assumes that the model of the world is known and the latter is computationally intractable.

6

Conclusions

This is the first work that studies adaptive submodular maximization in the setting where the model of the world is initially unknown. We propose an efficient bandit algorithm for solving the problem and prove that its expected cumulative regret increases logarithmically with time. Our work can be viewed as reinforcement learning (RL) [15] for adaptive submodularity. The main difference in our setting is that we can learn near-optimal policies without estimating the value function. Learning of value functions is typically hard, even when the model of the problem is known. Fortunately, this is not necessary in our problem and therefore we can develop a very efficient learning algorithm. We assume that the states of items are distributed independently of each other. In our experiments, this assumption was less restrictive than we expected (Section 4). Nevertheless, we believe that our approach should be studied under less restrictive assumptions. In preference elicitation (Section 4), for instance, the answers to questions are likely to be correlated due to many factors, such as user?s preferences, user?s mood, and the similarity of the questions. Our current model cannot capture any of these dependencies. However, we believe that our approach is quite general and can be extended to more complex models. We think that any such generalization would comprise three major steps: choosing a model of P (?), deriving a corresponding upper confidence bound on the expected gain, and finally proving an equivalent of Lemma 4. We also assume that the expected gain of choosing an item (Equation 7) can be written as a product of some known gain function (Equation 8) and the probability of the item?s states. This assumption is quite natural in maximum coverage problems but may not be appropriate in other problems, such as generalized binary search [4]. Our upper bound on the expected regret at step k (Lemma 1) may be loose in practice because it is obtained by maximizing over all contexts y ? Yk . In general, it is difficult to prove a tighter bound. Such a bound would have to depend on the probability of making a mistake in a specific context at step k, which depends on the policy in that episode, and indirectly on the progress of learning in all earlier episodes. We leave this for future work. 8

2 References

[1] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47:235?256, 2002. [2] Sandilya Bhamidipati, Branislav Kveton, and S. Muthukrishnan. Minimal interaction search: Multi-way search with item categories. In Proceedings of AAAI Workshop on Intelligent Techniques for Web Personalization and Recommendation, 2013. [3] Nicolo Cesa-Bianchi and Gabor Lugosi. Prediction, Learning, and Games. Cambridge University Press, New York, NY, 2006. [4] Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In Advances in Neural Information Processing Systems 17, pages 337?344, 2005. [5] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. Journal of Artificial Intelligence Research, 42:427? 486, 2011. [6] Andrew Guillory and Jeff Bilmes. Online submodular set cover, ranking, and repeated active learning. In Advances in Neural Information Processing Systems 24, pages 1107?1115, 2011. [7] Andrew Guillory and Jeff Bilmes. Simultaneous learning and covering with adversarial noise. In Proceedings of the 28th International Conference on Machine Learning, pages 369?376, 2011. [8] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. Journal of Machine Learning Research, 11:1563?1600, 2010. ? Tardos. Maximizing the spread of influence through a [9] David Kempe, Jon Kleinberg, and Eva social network. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 137?146, 2003. [10] Andreas Krause, Ajit Paul Singh, and

Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. Journal of Machine Learning Research, 9:235?284, 2008. [11] Shyong Lam and Jon Herlocker. MovieLens 1M Dataset. http://www.grouplens.org/node/12, 2012. [12] Christopher Manning, Prabhakar Raghavan, and Hinrich Sch?utze. Introduction to Information Retrieval. Cambridge University Press, New York, NY, 2008. [13] R?emi Munos. The optimistic principle applied to games, optimization, and planning: Towards foundations of Monte-Carlo tree search. Foundations and Trends in Machine Learning, 2012. [14] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - I. Mathematical Programming, 14(1):265?294, 1978. [15] Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998. [16] Zheng Wen, Branislav Kveton, Brian Eriksson, and Sandilya Bhamidipati. Sequential Bayesian search. In Proceedings of the 30th International Conference on Machine Learning, pages 977? 983, 2013. [17] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In Advances in Neural Information Processing Systems 24, pages 2483?2491, 2011.

9