Adaptive Submodularity:
A New Approach to Active Learning and Stochastic Optimization

Daniel Golovin
California Institute of Technology
Pasadena, CA 91125
dgolovin@caltech.edu

Andreas Krause
California Institute of Technology
Pasadena, CA 91125
krausea@caltech.edu

Abstract

Solving stochastic optimization problems under partial observability, where one needs to adaptively make decisions with uncertain outcomes, is a fundamental but notoriously difficult challenge. In this paper, we introduce the concept of *adaptive submodularity*, generalizing submodular set functions to adaptive policies. We prove that if a problem satisfies this property, a simple adaptive greedy algorithm is guaranteed to be competitive with the optimal policy. We illustrate the usefulness of the concept by giving several examples of adaptive submodular objectives arising in diverse applications including sensor placement, viral marketing and pool-based active learning. Proving adaptive submodularity for these problems allows us to recover existing results in these applications as special cases and leads to natural generalizations.

1 Introduction

In many natural optimization problems one needs to adaptively make a sequence of decisions, taking into account observations about the outcome of past decisions. Often, these outcomes are uncertain, and one may only know a probability distribution over them. Finding optimal policies for decision making in such partially observable stochastic optimization problems is notoriously intractable. In this paper, we analyze a particular class of partially observable stochastic optimization problems. We introduce the concept of *adaptive submodularity*, and prove that if a problem satisfies this property, a simple adaptive greedy algorithm is guaranteed to obtain near-optimal solutions. *Adaptive submodularity* generalizes the notion of submodularity[1], which has been successfully used to develop approximation algorithms for a variety of non-adaptive optimization problems. Submodularity, informally, is an intuitive notion of diminishing returns, which states that adding an element to a small set helps more than adding that same element to a larger (super-)set. A celebrated result of Nemhauser et al. (1978) guarantees that for such submodular functions, a simple greedy algorithm, which adds the element that maximally increases the objective value, selects a near optimal set of k elements. The challenge in generalizing submodularity to adaptive planning is that feasible solutions are now policies (decision trees) instead of subsets. We consider a natural analog of the diminishing returns property for adaptive problems, which reduces to the classical notion of submodular set functions for deterministic distributions. We show how the results of Nemhauser et al. generalize to the adaptive setting. We further demonstrate the usefulness of the concept by showing how it captures known results in stochastic optimization and active learning as special cases, and leads to natural generalizations.

As a first example, consider the problem of deploying a collection of sensors to monitor some spatial phenomenon. Each sensor can cover a region depending on its sensing range. Suppose we would like to find the best subset of k locations to place the sensors. In this application, intuitively, adding a sensor helps more if we have placed few sensors so far and helps less if we have already placed many sensors. We can formalize this diminishing returns property using the notion of submodularity – the total area covered by the sensors is a submodular function defined over all sets of locations. Krause and Guestrin (2007) show that many more realistic utility functions in sensor placement (such as the improvement in prediction accuracy w.r.t. some probabilistic model) are submodular as well. Now consider the following stochastic variant: Instead of deploying a fixed set of sensors, we deploy one sensor at a time. With a certain probability, deployed sensors can fail, and our goal is to maximize the area covered by the functioning sensors. Thus, when deploying the next sensor, we need to take into account which of the sensors we deployed in the past failed. This problem

[1] For an extensive treatment of submodularity, see the books of Fujishige (1991) and Schrijver (2003).

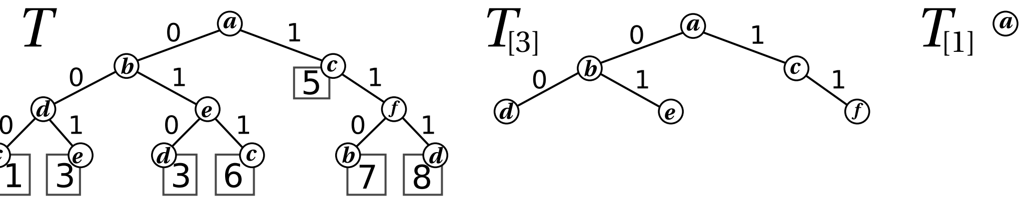

Figure 1: Left: Example policy tree, with edges labelled by state and rewards at (potential) terminal nodes in the rectangles. Middle and right: Prunings of policy trees T at layers 3 and 1.

has been studied by Asadpour et al. (2008) for the case where each sensor fails independently at random. In this paper, we show that the coverage objective is adaptive submodular, and use this concept to handle more general settings (where, e.g., rather than all-or-nothing failures there are different types of sensor failures of varying severity).

As another example, consider a viral marketing problem, where we are given a social network, and we want to influence as many people as possible in the network to buy some product. We do that by giving the product for free to a subset of the people, and hope that they convince their friends to buy the product as well. Formally, we have a graph, and each edge e is labeled by a number $0 \leq p_e \leq 1$. We "influence" a subset of nodes in the graph, and for each influenced node, their neighbors get randomly influenced according to the probability annotated on the edge connecting the nodes. This process repeats, until no further node gets influenced. Kempe et al. (2003) show that the set function which quantifies the expected number of nodes influenced is submodular. A natural stochastic variant of the problem is where we pick a node, get to see which nodes it influenced, then adaptively pick the next node based on these observations and so on. We show that a large class of such adaptive influence maximization problems satisfies adaptive submodularity.

Our third application is in pool-based active learning, where we are given an unlabeled data set, and we would like to adaptively pick a small set of examples whose labels imply all other labels. Thus, we want to pick examples to shrink the remaining version space (the set of consistent hypotheses) as quickly as possible. Here, we show that the reduction in version space mass is adaptive submodular, and use that observation to prove that the adaptive greedy algorithm is a near-optimal querying policy, recovering and generalizing results by Kosaraju et al. (1999) and Dasgupta (2004). Our results for active learning are also related to recent results of Guillory and Bilmes (2010) who study a generalization of submodular set cover to an interactive setting. In contrast to our approach however, Guillory and Bilmes (2010) analyze worst-case costs, and use rather different technical definitions and proof techniques.

In summary, our main contributions are:

- We consider a particular class of adaptive stochastic optimization problems, which we prove to be hard to approximate in general.
- We introduce the concept of *adaptive submodularity*, and prove that if a problem instance satisfies this property, a simple adaptive greedy policy performs near-optimally.
- We illustrate adaptive submodularity on several realistic problems, including Stochastic Maximum Coverage, Adaptive Viral Marketing and Active Learning. For these applications, adaptive submodularity allows us to recover known results and prove natural generalizations.

2 Adaptive Stochastic Optimization

Let E be a finite set of items. Each item $e \in E$ is in a particular state $\Phi(e) \in O$ from a set O of possible states. Hereby, $\Phi : E \to O$ is a (random) *realization* of the ground set, indicating which state each item is in. We take a Bayesian approach and assume that there is a (known) probability distribution $\mathbb{P}[\Phi]$ over realizations. We will consider the problem where we sequentially pick an item $e \in E$, get to see its state $\Phi(e)$, pick the next item, get to see its state, and so on. After each pick, our observations so far can be represented as a *partial realization* $\Psi \subseteq E \times O$, a function from some subset of E (i.e., the set of items that we already picked) to their states. A partial realization Ψ is *consistent* with a realization Φ if they are equal everywhere in the domain of Ψ. In this case we write $\Phi \sim \Psi$. If Ψ and Ψ' are both consistent with some Φ, and $\mathrm{dom}(\Psi) \subset \mathrm{dom}(\Psi')$, we say Ψ is a *subrealization* of Ψ'.

We encode our adaptive strategy for picking items as a *policy* π, which is a function from a set of partial realizations to E, specifying which item to pick next under a particular set of observations. If $\Psi \notin \mathrm{dom}(\pi)$, the policy terminates (stops picking items) upon observation of Ψ. Technically, we require that the domain of π is closed under subrealizations. That is, if $\Psi' \in \mathrm{dom}(\pi)$ and Ψ is a subrealization of Ψ' then $\Psi \in \mathrm{dom}(\pi)$. This condition simply ensures that the decision tree T^π associated with π as described below is connected. We define both $E(\pi, \Phi)$ and $E(T^\pi, \Phi)$ as the set of items picked by π conditioned on realization Φ. We also allow randomized policies that are functions from a set of partial realizations to distributions on E.

Each deterministic policy π can be associated with a tree T^π in a natural way (see Fig. 1 (left) for an illustration). We create the root of T^π, and label it with a tuple consisting of a partial realization $\emptyset$ and an item $\pi(\emptyset)$. Then inductively for each node, if its label is (Ψ, e), we construct a child for it for each state x such that $\Psi \cup \{(e,x)\} \in \text{dom}(\pi)$, labeled with $(\Psi \cup \{(e,x)\}, \pi(\Psi \cup \{(e,x)\}))$. A missing child for state x simply means that the policy terminates (stops picking items upon observing x). Thus, the first coordinate of the label at a node indicates what is known when the policy reaches that node, and the second coordinate indicates what it will do next. Similarly, randomized policies can be associated with distributions over trees in a natural way.

We wish to maximize, subject to some constraints, a utility function $f : 2^E \times O^E \to \mathbb{R}_{\geq 0}$ that depends on which items we pick and which state each item is in. Based on this notation, the expected utility of a policy π is $f_{\text{avg}}(\pi) := \mathbb{E}_\Phi[f(E(\pi, \Phi), \Phi)]$. The goal of the *Adaptive Stochastic Maximization* problem is to find a policy π^* such that

$$\pi^* \in \arg\max_\pi f_{\text{avg}}(\pi) \text{ subject to } |E(\pi, \Phi)| \leq k \text{ for all } \Phi, \tag{1}$$

where k is a budget on how many items can be picked.

Unfortunately, as we will show in §8, even for linear functions f, i.e., those where $f(A, \Phi) = \sum_{e \in A} w_{e,\Phi}$ is simply the sum of weights (depending on the realization Φ), Problems (1) is hard to approximate under reasonable complexity theoretic assumptions. Despite the hardness of the general problem, in the following sections we will identify conditions that are sufficient to allow us to approximately solve it.

Incorporating Item Costs. Instead of quantifying the cost of a set $E(\pi, \Phi)$ by the number of elements $|E(\pi, \Phi)|$, we can also consider the case where each item $e \in E$ has a cost $c(e)$. We show how to handle this extension and give many other results in the extended version of this paper (Golovin & Krause, 2010).

3 Adaptive Submodularity

We first review the classical notion of submodular set functions, and then introduce the novel notion of adaptive submodularity.

Submodularity. Let us first consider the simple special case where $\mathbb{P}[\Phi]$ is deterministic or, equivalently, $|O| = 1$. In this case, the realization Φ is known to the decision maker in advance, and thus there is no benefit in adaptive selection. Thus, Problem (1) is equivalent to finding a set $A^* \subseteq E$ such that

$$A^* \in \arg\max_{A \subseteq E} f(A, \Phi) \text{ such that } |A| \leq k. \tag{2}$$

For most interesting classes of utility functions f, this is an NP-hard optimization problem. However, in many practical problems, such as those mentioned in §1, $f(A) = f(A, \Phi)$ satisfies *submodularity*. A set function $f : 2^E \to \mathbb{R}$ is called submodular if, whenever $A \subseteq B \subseteq E$ and $e \in E \setminus B$ it holds that

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B), \tag{3}$$

i.e., adding e to the smaller set A increases f at least as much as adding e to the superset B. Furthermore, f is called *monotone*, if, whenever $A \subseteq B$ it holds that $f(A) \leq f(B)$. A celebrated result by Nemhauser et al. (1978) states that for monotone submodular functions with $f(\emptyset) = 0$, a simple greedy algorithm that starts with the empty set, $A_0 = \emptyset$ and chooses $A_{i+1} = A_i \cup \{\arg\max_{e \in E \setminus A_i} f(A_i \cup \{e\})$ guarantees that $f(A_k) \geq (1 - 1/e) \max_{|A| \leq k} f(A)$. Thus, the greedy set A_k obtains at least a $(1 - 1/e)$ fraction of the optimal value achievable using k elements. Furthermore, Feige (1998) shows that this result is tight if P $\neq$ NP; under this assumption no polynomial time algorithm can achieve a $(1 - 1/e + \epsilon)$-approximation for any constant $\epsilon > 0$, even for the special case of Maximum k-Cover where $f(A)$ is the cardinality of the union of sets indexed by A.

Now let us relax the assumption that $\mathbb{P}[\Phi]$ is deterministic. In this case, we may still want to find a non-adaptive solution (i.e., a constant policy π_A that always picks set A independently of Φ) maximizing $f_{\text{avg}}(\pi_A)$. If f is *pointwise* submodular, i.e., $f(A, \Phi)$ is submodular in A for any fixed Φ, the function $f(A) = f_{\text{avg}}(\pi_A)$ is submodular, since nonnegative linear combinations of submodular functions remain submodular. Thus, the greedy algorithm allows us to find a near-optimal *non-adaptive* policy.

However, in practice, we may be more interested in obtaining a non-constant policy π, that *adaptively* chooses items based on previous observations. Thus, the question is whether there is a natural extension of submodularity to policies. In the following, we will develop such a notion – *adaptive submodularity*.

Adaptive submodularity. The key challenge is to find an appropriate generalization of the diminishing returns condition (3). Informally, our generalization will require that playing a layer k of a policy tree T^π earlier in the policy cannot decrease its marginal contribution to the objective. Since there are many more nodes at layer k than at earlier layers, we consider playing an appropriate distribution at earlier layers to make the comparison formal.

We will now formalize the above intuition. Given a tree $T = T^{\pi}$ we define its *level-k-pruning* $T_{[k]}$ as the subtree of T induced on all nodes of depth k or less, i.e., those that can be reached from the root via a path of at most $k-1$ edges. Tree pruning is illustrated in Fig. 1. Given two policies π_1, π_2 associated with trees T_1 and T_2 we define $\pi_1 @ \pi_2$ as the policy obtained by running π_1 to completion, and then running policy π_2 as if from a fresh start, ignoring the information gathered during the running of π_1. We let $T_1 @ T_2$ denote the tree associated with policy $\pi_1 @ \pi_2$. This concept is illustrated in Fig. 2.

Fix any integers i and j so that $0 \le i < j$, and any policy π. Define $\mathcal{D}(T, \Psi, j)$ to be the distribution on E induced by executing T under a random realization which is consistent with Ψ, and then outputting the item selected at depth j in T. For any node u of $T = T^{\pi}$, let $e(u)$ denote the item selected by u, and let Ψ_u be the partial realization encoding all state observations known to T just as it reaches u. As illustrated in Fig. 3, let $T^{\pi}_{[i]\cup\{j\}}$ be the (random) tree obtained as follows: Start with $T_{[i]}$ and for each of its leaves u and every possible state o (i.e., those with $\mathbb{P}[\Phi(e(u)) = o \mid \Phi \sim \Psi_u] > 0$) connect u to a new node which plays a (random) item e drawn from $\mathcal{D}(T, \Psi_u \cup \{(e(u), o)\}, j)$. The new node's corresponding partial realization, indicating what is known when it is first reached, is $\Psi_u \cup \{(e(u), o)\}$. Note that if T terminates before selecting j items for some realizations consistent with Ψ, then $\mathcal{D}(T, \Psi, j)$ will select nothing at all with the total conditional probability mass of such realizations.

We now introduce our generalizations of monotonicity and submodularity to the adaptive setting:

Definition 1 (Adaptive Monotonicity) *A function $f : 2^E \times O^E \to \mathbb{R}_{\ge 0}$ is* adaptive monotone *with respect to distribution $\mathbb{P}[\Phi]$ if for all policies π, π' it holds that $f_{avg}(\pi) \le f_{avg}(\pi' @ \pi)$, where $f_{avg}(\pi) := \mathbb{E}_{\Phi}[f(E(\pi, \Phi), \Phi)]$ is defined w.r.t. $\mathbb{P}[\Phi]$.*

Definition 2 (Adaptive Submodularity) *A function $f : 2^E \times O^E \to \mathbb{R}_{\ge 0}$ is* adaptive submodular *with respect to distribution $\mathbb{P}[\Phi]$ if for all policies π and for all $0 \le i < j$*

$$f_{avg}(T^{\pi}_{[j]}) - f_{avg}(T^{\pi}_{[j-1]}) \le \mathbb{E}\left[f_{avg}(T^{\pi}_{[i]\cup\{j\}}) - f_{avg}(T^{\pi}_{[i]})\right] \tag{4}$$

where the expectation is over the random choice of $T^{\pi}_{[i]\cup\{j\}}$ and f_{avg} is defined w.r.t. $\mathbb{P}[\Phi]$.

We will give concrete examples of adaptive monotone and adaptive submodular functions that arise in the applications introduced in §1 in §5, §6 and §7. It turns out there is an equivalent characterization of adaptive submodular functions in terms of derivatives of the expected value with respect to each item e, conditioned on the states of the previously selected items. We denote this derivative by $\Delta_{\Psi}(e)$, where Ψ is the current partial realization. Formally,

$$\Delta_{\Psi}(e) := \mathbb{E}_{\Phi}[f(\mathrm{dom}(\Psi) \cup \{e\}, \Phi) - f(\mathrm{dom}(\Psi), \Phi) \mid \Phi \sim \Psi]. \tag{5}$$

Proposition 3 *A function $f : 2^E \times O^E \to \mathbb{R}_{\ge 0}$ is adaptive submodular if and only if for all Ψ and Ψ' such that Ψ is a subrealization of Ψ' (i.e., $\Psi \subseteq \Psi'$), and for all e, we have $\Delta_{\Psi'}(e) \le \Delta_{\Psi}(e)$.*

Proof: ($\Rightarrow$) To get from Eq. (4) to $\Delta_{\Psi'}(e) \le \Delta_{\Psi}(e)$, generate an order $\prec$ for $\mathrm{dom}(\Psi')$ such that each item in $\mathrm{dom}(\Psi)$ is less than each item in $\mathrm{dom}(\Psi') \setminus \mathrm{dom}(\Psi)$. Let $e_1, e_2, \ldots, e_m$ be the items of $\mathrm{dom}(\Psi')$ in order of $\prec$. Let $e_{m+1} := e$. Define a policy tree T that is a path $u_1, u_2, \ldots, u_m, u_{m+1}$ where each u_i is labeled with partial realization $\{(e_j, \Psi'(e_j)) : j < i\}$ and selects item e_i. Then applying Eq. (4) with T and $i = |\mathrm{dom}(\Psi)|$, $j = m+1$ yields $\mathbb{P}[\Psi'|\Psi] \cdot \Delta_{\Psi'}(e) \le \mathbb{P}[\Psi'|\Psi] \cdot \Delta_{\Psi}(e)$ and hence $\Delta_{\Psi'}(e) \le \Delta_{\Psi}(e)$.
($\Leftarrow$) Informally, if $\Delta_{\Psi'}(e) \le \Delta_{\Psi}(e)$ for all $\Psi \subseteq \Psi'$ and e, then in any tree T moving items from layer j up to layer i cannot decrease their marginal benefit. Since each item e in layer j of T is selected with the same probability in $T^{\pi}_{[j]}$ and in $T^{\pi}_{[i]\cup\{j\}}$, this implies Eq. (4). ■

Properties of adaptive submodular functions. It can be seen that adaptive monotonicity and adaptive submodularity enjoy similar closure properties as monotone submodular functions. In particular, if $w_1, \ldots, w_m \ge 0$ and $f_1, \ldots, f_m$ are adaptive monotone submodular w.r.t. distribution $\mathbb{P}[\Phi]$, then $f(A, \Phi) = \sum_{i=1}^{m} w_i f_i(A, \Phi)$ is adaptive monotone submodular w.r.t. $\mathbb{P}[\Phi]$. Similarly, for a fixed constant $c \ge 0$ and adaptive monotone submodular function f, the function $g(E, \Phi) = \min(f(E, \Phi), c)$ is adaptive monotone submodular. Thus, adaptive monotone submodularity is preserved by nonnegative linear combinations and by truncation.

4 Guarantee for the Greedy Policy

The greedy policy π_{greedy} at each time step tries to myopically increase the expected objective value, given its current observations. That is, suppose $f : 2^E \times O^E \to \mathbb{R}_{\ge 0}$ is the objective, and and Ψ is the partial realization indicating the states of items selected so far. Then the greedy policy will select the item e maximizing the

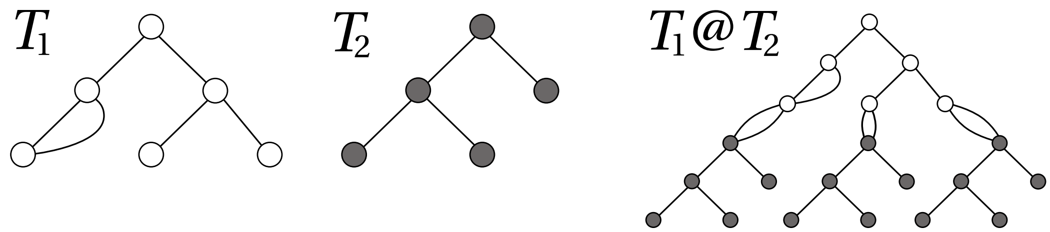

Figure 2: Concatenation of policy trees.

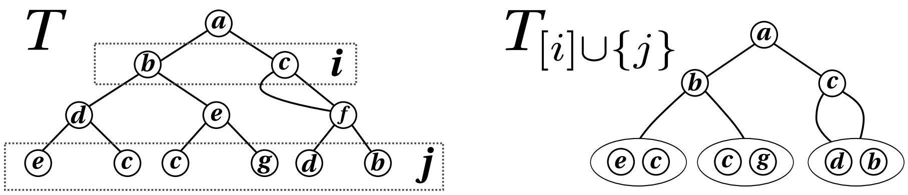

Figure 3: The "collapsed" tree $T_{[i]\cup\{j\}}$ which, speaking informally, plays layer j of T in its layer $i+1$. Each leaf of $T_{[i]\cup\{j\}}$ is depicted as containing the set of items it samples from. The diminishing returns condition for adaptive submodularity states that the marginal benefit of the j^{th} layer in T may not exceed the marginal benefit of the $(i+1)^{\text{th}}$ layer in $T_{[i]\cup\{j\}}$. The latter quantity is the marginal benefit layer $(i+1)$ would obtain if each layer $(i+1)$ node u selected its item from the distribution on items selected at layer j by executions of T that reach u. For example, if T has a 30% chance of picking e as its last item, conditioned on it reaching the layer $(i+1)$ node labeled d, then the left-most leaf of $T_{[i]\cup\{j\}}$ picks e with 30% probability and picks c with 70% probability. Alternately, if we use the convention that left edges correspond to the selected item being in state 0, and right edges correspond to the selected item being in state 1 (as depicted in Fig. 1), and let $\Psi = \{(a,0),(b,0)\}$, then we may say that the left-most leaf of $T_{[i]\cup\{j\}}$ picks an item from distribution $\mathcal{D}(T,\Psi,j)$, so that it picks e with probability $\mathbb{P}_\Phi\left[T \text{ picks } e \text{ in layer } j \mid \Phi\sim\Psi\right] = 0.3$, and picks c with probability $\mathbb{P}_\Phi\left[T \text{ picks } c \text{ in layer } j \mid \Phi\sim\Psi\right] = 0.7$.

expected increase in value, conditioned on the observed states of items it has already selected (i.e., conditioned on $\Phi\sim\Psi$). That is, it will select e to maximize the quantity $\Delta_\Psi(e)$ defined in Eq. (5).

In some applications, finding an item maximizing $\Delta_\Psi(e)$ may be computationally intractable, and the best we can do is find an α-approximation to the best greedy move. This means we find an e' such that $\Delta_\Psi(e') \geq \frac{1}{\alpha}\max_e \Delta_\Psi(e)$. We call a policy which always selects such an item an *α-approximate greedy policy*.

In this section we establish that if the objective function is adaptive submodular with respect to the distribution describing the environment in which we operate, then the greedy policy and any α-approximate greedy policy inherit precisely the performance guarantees of the greedy and α-approximate greedy algorithms for classic (nonadaptive) submodular maximization. We have the following result.

Theorem 4 *Fix any $\alpha\geq 1$. If f is adaptive monotone and adaptive submodular with respect to the distribution $\mathbb{P}[\Phi]$, and π is an α-approximate greedy policy, then for all policies π^* and positive integers ℓ, k*

$$f_{avg}(T^{\pi}_{[\ell]}) > \left(1 - e^{-\ell/\alpha k}\right) f_{avg}(T^{\pi^*}_{[k]}).$$

In particular, with $\ell = k$ this implies any α-approximate greedy policy achieves a $\left(1-e^{-1/\alpha}\right)$ approximation to the expected reward of the best policy, if both are terminated after running for an equal number of steps.

Proof: The proof goes along the lines of the performance analysis of the greedy algorithm for maximizing a submodular function subject to a cardinality constraint found in Nemhauser et al. (1978). An extension of that analysis to α-approximate greedy algorithms, which is analogous to ours but for the nonadaptive case, is shown by Goundan and Schulz (2007). Let $T = T^{\pi}_{[\ell]}, T^* = T^{\pi^*}_{[k]}$. Then for all i, $0 \leq i < \ell$

$$\begin{aligned}
f_{\text{avg}}(T^*) &\leq f_{\text{avg}}(T_{[i]} @ T^*) && (6)\\
&= f_{\text{avg}}(T_{[i]}) + \sum_{j=1}^{k}\left(f_{\text{avg}}(T_{[i]} @ T^*_{[j]}) - f_{\text{avg}}(T_{[i]} @ T^*_{[j-1]})\right) && (7)\\
&\leq f_{\text{avg}}(T_{[i]}) + \sum_{j=1}^{k}\mathbb{E}\left[f_{\text{avg}}\left((T_{[i]} @ T^*)_{[i]\cup\{i+j\}}\right) - f_{\text{avg}}(T_{[i]})\right] && (8)\\
&\leq f_{\text{avg}}(T_{[i]}) + \alpha\sum_{j=1}^{k}\left(f_{\text{avg}}(T_{[i+1]}) - f_{\text{avg}}(T_{[i]})\right) && (9)
\end{aligned}$$

The first inequality is due to the adaptive monotonicity of f, from which we may infer $f_{\text{avg}}(T_2) \leq f_{\text{avg}}(T_1 @ T_2)$ for any T_1 and T_2. The second is a simple telescoping sum. The third is a direct application of the adaptive submodularity guarantee of f with $T_{[i]} @ T^*_{[j]}$ at levels i and $i+j$, and the fourth is by the definition of an α-approximate greedy policy. Now define $\Delta_i := f_{\text{avg}}(T^*) - f_{\text{avg}}(T_{[i]})$, so that Eq. (9) implies $\Delta_i \leq \alpha k(\Delta_i - \Delta_{i+1})$, from which we infer $\Delta_{i+1} \leq \left(1 - \frac{1}{\alpha k}\right) \Delta_i$ and hence $\Delta_\ell \leq \left(1 - \frac{1}{\alpha k}\right)^\ell \Delta_0 < e^{-\ell/\alpha k}\Delta_0$, where for this last inequality we have used the fact that $1-x < e^{-x}$ for all $x > 0$. Thus $f_{\text{avg}}(T^*) - f_{\text{avg}}(T_{[\ell]}) < e^{-\ell/\alpha k}\left(f_{\text{avg}}(T^*) - f_{\text{avg}}(T_{[0]})\right) \leq e^{-\ell/\alpha k} f_{\text{avg}}(T^*)$ so $f_{\text{avg}}(T) > (1 - e^{-\ell/\alpha k}) f_{\text{avg}}(T^*)$. ■

Note that if the greedy rule can be implemented only with small *absolute* error rather than small *relative* error, i.e., $\Delta_\Psi(e') \geq \max_e \Delta_\Psi(e) - \varepsilon$, a similar argument shows that

$$f_{\text{avg}}(T^\pi_{[\ell]}) \geq \left(1 - e^{-\ell/k}\right) f_{\text{avg}}(T^{\pi^*}_{[k]}) - \ell\varepsilon.$$

This is important, since small absolute error can always be achieved (with high probability) whenever f can be evaluated efficiently, and sampling $P(\Phi \mid \Psi)$ is efficient. In this case, we can approximate

$$\Delta_\Psi(e) \approx \frac{1}{N}\sum_{i=1}^{N}\left[f(\text{dom}(\Psi) \cup \{e\}, \Phi_i) - f(\text{dom}(\Psi), \Phi_i)\right],$$

where Φ_i are sampled i.i.d. from $P(\Phi \mid \Psi)$. Note that the characterization of adaptive submodularity in Proposition 3 allows us to implement an "accelerated" version of the adaptive greedy algorithm using lazy evaluations of marginal benefits as originally suggested for the nonadaptive case by Minoux (1978).

5 Application: Stochastic Submodular Maximization

As our first application, consider the sensor placement problem introduced in §1. Suppose we would like to monitor a spatial phenomenon such as temperature in a building. We discretize the environment into a set E of locations. We would like to pick a subset $A \subseteq E$ of k locations that is most "informative", where we use a set function $\hat{f}(A)$ quantifying the informativeness of placement A. Krause and Guestrin (2007) show that many natural objective functions (such as reduction in predictive uncertainty measured in terms of Shannon entropy) are monotone submodular.

Now consider the problem, where sensors can fail or partially fail (e.g., be subject to some varying amount of noise) after deployment. We can model this extension by assigning a state $\Phi(e) \in O$ to each possible location, indicating the extent to which a sensor placed at location e is working. To quantify the value of a set of sensor deployments under a realization Φ indicating to what extent the various sensors are working, we first define (e, o) for each $e \in E$ and $o \in O$, which represents the placement of a sensor in state o at location e. We then suppose there is a function $\hat{f} : 2^{E\times O} \to \mathbb{R}_{\geq 0}$ which quantifies the informativeness of a set of sensor deployments in arbitrary states. The utility $f(A, \Phi)$ of placing sensors at the locations in A under realization Φ then is

$$f(A, \Phi) = \hat{f}(\{(e, \Phi(e)) : e \in A\}).$$

We aim to adaptively place k sensors to maximize our expected utility. We assume that sensor failures at each location are independent of each other, i.e., $\mathbb{P}[\Phi] = \prod_e \mathbb{P}[\Phi(e)]$, where $\mathbb{P}[\Phi(e) = o]$ is the probability that a sensor placed at location e will be in state o. Goemans and Vondrák (2006) studied a related problem called *Stochastic Covering* where the goal is to achieve the maximum attainable objective value at minimum cost, i.e., their problem generalizes Set Cover in the same way our problem generalizes Maximum k-Cover. Asadpour et al. (2008) studied a special case of our problem, in which sensors either fail completely (in which case they contribute no value at all) or work perfectly, under the name *Stochastic Submodular Maximization*. They proved that the adaptive greedy algorithm obtains a constant fraction $(1 - 1/e)$ approximation to the optimal adaptive policy, provided $\hat{f}$ is monotone submodular. We extend their result to multiple types of failures by showing that $f(A, \Phi)$ is adaptive submodular with respect to distribution $\mathbb{P}[\Phi]$ and then invoking Theorem 4.

Theorem 5 *Fix a prior such that $\mathbb{P}[\Phi] = \prod_{e\in E} \mathbb{P}[\Phi(e)]$, and integer k and let the objective function $\hat{f} : 2^{E\times O} \to \mathbb{R}_{\geq 0}$ be monotone submodular. Let π be the adaptive greedy policy attempting to maximize f, and let π^* be any policy. Then*

$$f_{avg}(T^\pi_{[k]}) \geq \left(1 - \frac{1}{e}\right) f_{avg}(T^{\pi^*}_{[k]}).$$

Proof: We prove Theorem 5 by first proving f is adaptive monotone and adaptive submodular in this model, and then applying Theorem 4. Adaptive monotonicity is readily proven after observing that $f(\cdot, \Phi)$ is monotone for each Φ, and noting that for any Φ, T_1 and T_2, we have $E(T_2, \Phi) \subseteq E(T_1 @ T_2, \Phi)$. Moving on to adaptive submodularity, fix any T and $i < j$. We prove Eq. (4) using the alternate characterization of Proposition 3. We use a coupled distribution over the realizations seen when running $T_{[j]}$ and $T_{[i]\cup\{j\}}$, such that the same realization is sampled for both. For any partial realization Ψ encoding the observations made immediately before reaching a level $i+1$ node, and any ground set item e such that e is in the support of $\mathcal{D}(T, \Psi, j)$, consider the expected marginal contribution of e to the objective conditioned on the fact that the policy has observed Ψ for trees T and $T_{[i]\cup\{j\}}$. In both cases e is equally likely to be selected by the policy, and is equally likely be in any given state, since $\Phi(e)$ is independent of $\{\Phi(e') : e' \in E \setminus \{e\}\}$. However, its marginal contribution under $T_{[j]}$ can be at most that under $T_{[i]\cup\{j\}}$ by the submodularity of $\hat{f}$, since in the former case there are potentially more items in the base set to which we add $(e, \Phi(e))$ (namely, the realized versions $(e', \Phi(e'))$ of those items e' selected in layers $i+1$ through $j-1$), but there are never fewer items in it. ■

6 Application: Adaptive Viral Marketing

For our next application, consider the following scenario. Suppose we would like to generate demand for a genuinely novel product. Potential customers do not realize how valuable the new product will be in their lives, and conventional advertisements are failing to induce them to try it. In this case, we may try to spur demand by offering a special promotional deal to a select few people, and hope that demand builds virally, propagating through the social network as people recommend the product to their friends and associates. Supposing we know something about the structure of the social networks people inhabit, and how ideas, innovation, and new product adoption diffuse through them, this begs the question: to which initial set of people should we offer the promotional deal, in order to spur maximum demand for our product? We imagine there is a fixed budget for the promotional campaign, which can be interpreted as a budget k indicating the maximum size of the initial set of people.

This, broadly, is the viral marketing problem. In the adaptive variant, we may select a person to offer the promotion to, make some observations about the resulting spread of demand for our product, and repeat. The same problem arises in the context of spreading technological, cultural, and intellectual innovations, broadly construed. In the interests of having unified terminology we follow Kempe et al. (2003) and talk of spreading *influence* through the social network, where we say people are *active* if they have adopted the idea or innovation in question, and *inactive* otherwise, and that *a influences b* if a convinces b to adopt the idea or innovation in question.

There are many ways to model the diffusion dynamics governing the spread of influence in a social network. We consider a basic and well-studied model, the *independent cascade model*, described in detail below. For this model Kempe et al. (2003) obtained a very interesting result; they showed that the eventual spread of the influence f (i.e., the ultimate number of customers that demand the product) is a monotone submodular function of the seed set S of initial people. This, in conjunction with the results of Nemhauser et al. (1978) implies that the greedy algorithm obtains at least $\left(1 - \frac{1}{e}\right)$ of the value of the best feasible seed set, $\arg\max_{S:|S|\leq k} f(S)$. In this section, we use the idea of adaptive submodularity to extend their results in two directions simultaneously. First, we extend the guarantees to the adaptive version of the problem, and show that the a greedy policy obtains at least $\left(1 - \frac{1}{e}\right)$ of the value of the best *policy*. Second, we achieve this guarantee not only for the case where our reward is simply the number of influenced people, but also for any (nonnegative) monotone submodular function of the *set* of people influenced.

Independent Cascade Model. In this model, the social network is a directed graph $G = (V, A)$ where each vertex in V is a person, and each edge $(u, v) \in A$ has an associated binary random variable X_{uv} indicating if u will influence v. That is, $X_{uv} = 1$ if u will influence v once it has been influenced, and $X_{uv} = 0$ otherwise. The random variables X_{uv} are independent, and have known means $p_{uv} := \mathbb{E}[X_{uv}]$. We will call an edge (u, v) with $X_{uv} = 1$ a *live edge* and an edge with $X_{uv} = 0$ a *dead edge*. When a node u is activated, the edges X_{uv} to each neighbor v of u are sampled, and v is activated if (u, v) is live. Influence can then spread from u's neighbors to their neighbors, and so on, according to the same process. Once active, nodes remain active throughout the process, however Kempe et al. (2003) show that this assumption is without loss of generality, and can be removed.

The Feedback Model. In the Adaptive Viral Marketing problem under the independent cascades model, the items correspond to people we can "activate" by, e.g., offering them the promotional deal. How we define the states $\Phi(u)$ depends on what information we obtain as a result of activating u. Given the nature of the diffusion process, activating u can have wide-ranging effects, so the state $\Phi(u)$ has more to do with the state of the social network on the whole than with u in particular. Specifically, we model $\Phi(u)$ as a function $\Phi_u : A \to \{0, 1, ?\}$, where $\Phi_u((u, v)) = 0$ means that activating u has revealed that (u, v) is dead, $\Phi_u((u, v)) = 1$ means that activating u has revealed that (u, v) is live, and $\Phi_u((u, v)) = ?$ means that activating u has not revealed the status of (u, v) (i.e., the value of X_{uv}). We require each realization to be *consistent* and *complete*. Consistency means that no edge should be declared both live and dead by any two states. That is, for all $u, v \in V$ and $a \in A$, $(\Phi_u(a), \Phi_v(a)) \notin \{(0,1),(1,0)\}$. Completeness means that the status of each edge is revealed by some activation. That is, for all $a \in A$ there exists $u \in V$ such that $\Phi_u(a) \in \{0,1\}$. A consistent and complete realization thus encodes X_{uv} for each edge (u,v). Let $A(\Phi)$ denote the live edges as encoded by Φ. There are several candidates for which edge sets we are allowed to observe when activating a node u. We consider the following two concrete feedback models:

Myopic Feedback: After activating u we get to see the status (live or dead) of all edges exiting u in the social network, i.e., $\partial_+(u) := \{(u,v) : v \in V\} \cap A$.

Full-Adoption Feedback: After activating u we get to see the status (live or dead) of all edges exiting v, for all nodes v reachable from u via live edges (i.e., reachable from u in $(V, A(\Phi))$, where Φ is the true realization.

The Objective Function. In the simplest case, the reward for influencing a set $U \subseteq V$ of nodes is $\hat{f}(U) := |U|$. Kempe et al. (2003) obtain an $\left(1 - \frac{1}{e}\right)$-approximation for the slightly more general case in which each node u has a weight w_u indicating its importance, and the reward is $\hat{f}(U) := \sum_{u \in U} w_u$. We generalize this result further, to include arbitrary nonnegative monotone submodular reward functions $\hat{f}$. This allows us, for example, to encode a value associated with the *diversity* of the set of nodes influenced, such as the notion that it is better to achieve 20% market penetration in five different (equally important) demographic segments than 100% market penetration in one and 0% in the others.

Comparison with Stochastic Submodular Maximization. It is worth contrasting the Adaptive Viral Marketing problem with the Stochastic Submodular Maximization problem of §5. In the latter problem, we can think of the items as being random independently distributed sets. In Adaptive Viral Marketing by contrast, the random sets (of nodes influenced when a fixed node is selected) depend on the random status of the edges, and hence may be correlated through them. Nevertheless, we can obtain the same $\left(1 - \frac{1}{e}\right)$ approximation factor for both problems.

We are now ready to formally state our result for this section.

Theorem 6 *The greedy policy obtains at least $\left(1 - \frac{1}{e}\right)$ of the value of the best* policy *for the Adaptive Viral Marketing problem with arbitrary monotone submodular reward functions, in the independent cascade model, in both feedback models discussed above. That is, if $\sigma(S, \Phi)$ is the set of all activated nodes when S is the seed set of activated nodes and Φ is the realization, $\hat{f} : 2^V \to \mathbb{R}_{\geq 0}$ is an arbitrary monotone submodular function indicating the reward for influencing a set, and the objective function is $f(S, \Phi) := \hat{f}(\sigma(S, \Phi))$, then*

$$f_{avg}(T^{greedy}_{[k]}) \geq \left(1 - \frac{1}{e}\right) f_{avg}(T_{[k]})$$

for all $k \in \mathbb{N}$, where T^{greedy} is the policy tree of the greedy policy, and T is any policy tree.

Proof: It suffices to prove that f is adaptive submodular with respect to the probability distribution on realizations $\mathbb{P}[\Phi]$, in both feedback models, because then we can invoke Theorem 4 to complete the proof.

We will say we have *observed* an edge (u,v) if we know its status, i.e., if it is live or dead. We will actually prove that f is adaptive submodular in any feedback model in which all observed edges (u,v) have u active (presuming the algorithm is aware of this fact). This includes the feedback models described above. Fix any policy tree T, and integers $i < j$. We aim to show Eq. (4) from the definition of adaptive submodularity holds, that is

$$f_{\text{avg}}(T^{\pi}_{[j]}) - f_{\text{avg}}(T^{\pi}_{[j-1]}) \leq \mathbb{E}\left[f_{\text{avg}}(T^{\pi}_{[i] \cup \{j\}}) - f_{\text{avg}}(T^{\pi}_{[i]})\right].$$

Fix a partial realization Ψ corresponding to the policy tree T's knowledge after making i selections, and sample a node $v \in V$ from the social network from $\mathcal{D}(T, \Psi, j)$, the distribution on nodes selected by T at layer j conditioned on the realization being consistent with Ψ (i.e., $\Phi \sim \Psi$), as described in §3.

We claim that the marginal contribution of v cannot be larger in $T_{[j]}$ than in $T_{[i] \cup \{j\}}$, despite the fact that when selecting v the former has observed more edges. We couple the distributions on the executions of $T_{[j]}$ and $T_{[i] \cup \{j\}}$ so that we can speak of a common Ψ between them. Let S be the random set of nodes activated by selecting v in $T_{[i] \cup \{j\}}$ conditioned on Ψ, and let S' be the analogous set for $T_{[j]}$. For two random subsets A, B of V, we say A *stochastically dominates* B if for all $U \subseteq V$ we have $\mathbb{P}[U \subseteq B] \leq \mathbb{P}[U \subseteq A]$. Now fix any $B, B' \subseteq V$ such that $B \subseteq B'$, and note that if S stochastically dominates S' then for all Φ

$$\mathbb{E}_{S'}[f(S' \cup B', \Phi) - f(B', \Phi)] \leq \mathbb{E}_S[f(S \cup B, \Phi) - f(B, \Phi)] \tag{10}$$

since $S \mapsto f(S, \Phi)$ is monotone submodular for all realizations Φ. Let B be the set of nodes activated by the first i nodes selected when executing T, and let B' to be the set of nodes activated by the first $j-1$ selected nodes. Then if we take the expectation of Eq. (10) with respect to sampling $\Phi \sim \Psi$, we get the adaptive submodularity condition for this i, j and T, conditioned on $\Phi \sim \Psi$. Taking an appropriate convex combination of these inequalities over valid choices for Ψ yields the adaptive submodularity condition for our arbitrary choices of i, j and T, and hence proves the overall adaptive submodularity of f.

We now show that S does in fact stochastically dominate S'. Intuitively, S stochastically dominates S' because if an edge (v_1, v_2) has been observed while executing layers in $[i+1, j-1]$ then v_1 is already active, and so activating v cannot result in the activation of v_1, i.e., $v_1 \notin S'$. Moreover if (v_1, v_2) is live, then v_2 is also already active, so $v_2 \notin S'$. On the other hand, if (v_1, v_2) is dead it makes it harder for v to spread influence than if (v_1, v_2) is merely unobserved as yet. More formally, consider any v in the support of $\mathcal{D}(T, \Psi, j)$; here v can depend on the partial realization seen by $T_{[j]}$ just before it makes a selection at layer j, which we denote by Ψ'. Next, fix $\Phi \sim \Psi'$ and consider the graph $(V, A(\Phi))$ of live edges. We argue that if we "remove" the elements of $\mathrm{dom}(\Psi') \setminus \mathrm{dom}(\Psi)$ and their effects (i.e., we deactivate the nodes they influenced), then the set of nodes influenced by playing v can only grow. Let $S(\Phi)$ denote the sets of nodes in influenced by playing v assuming Φ is the true realization and we have already selected $\mathrm{dom}(\Psi)$. Let $S'(\Phi)$ denote the analogous set if we have already selected $\mathrm{dom}(\Psi')$. We aim to prove $S'(\Phi) \subseteq S(\Phi)$. Note $S(\Phi)$ is the set of nodes reachable from v via the live edges $A(\Phi)$, excluding already active nodes (i.e., excluding those reachable from any node in $\mathrm{dom}(\Psi)$ via live edges). The analogous observation holds for $S'(\Phi)$, where the excluded nodes are those reachable from any node in $\mathrm{dom}(\Psi')$ via live edges. Since $\mathrm{dom}(\Psi) \subset \mathrm{dom}(\Psi')$ and the underlying graph $(V, A(\Phi))$ is the same in both cases, we infer $S'(\Phi) \subseteq S(\Phi)$. Hence conditioning on Ψ', for all $U \subseteq V$ we have

$$\mathbb{P}\left[U \subseteq S'(\Phi) | \Phi \sim \Psi'\right] \leq \mathbb{P}\left[U \subseteq S(\Phi) | \Phi \sim \Psi'\right].$$

Removing the conditioning on Ψ' by taking the expectation over all Ψ' consistent with Ψ, we infer S stochastically dominates S', which completes the proof. ■

7 Application: Active Learning

In pool-based active learning (McCallum & Nigam, 1998), we are given a set of hypotheses H, and a set of unlabeled data points X where each $x \in X$ is independently drawn from some distribution $\mathcal{D}$. Let L be the set of possible labels. The goal is to adaptively select points to query (i.e., to obtain labels for) until we can output a hypothesis h that will have expected error at most ε with probability $1-\delta$, for some fixed $\varepsilon, \delta > 0$. That is, if h^* is the target hypothesis (with zero error), and $\mathrm{error}(h) := \mathbb{P}_{x \sim \mathcal{D}}\left[h(x) \neq h^*(x)\right]$ is the error of h, we require $\mathbb{P}\left[\mathrm{error}(h) \leq \varepsilon\right] \geq 1 - \delta$. The latter probability is taken with respect to $\mathcal{D}(X)$; the learned hypothesis h and thus $\mathrm{error}(h)$ depend on it.

In the case of binary labels $L = \{-1, 1\}$, various authors have considered greedy policies which generalize binary search (Garey & Graham, 1974; Loveland, 1985; Arkin et al., 1993; Kosaraju et al., 1999; Dasgupta, 2004; Guillory & Bilmes, 2009; Nowak, 2009). The simplest of these, called *generalized binary search* (GBS) or the *splitting algorithm*, works as follows. Define the *version space* V to be the set of hypotheses consistent with the observed labels (here we assume that there is no label noise). In the worst-case setting, GBS selects a query $x \in X$ that minimizes $\left|\sum_{h \in V} h(x)\right|$. In the Bayesian setting we assume we are given a prior p_H over hypotheses; in this case GBS selects a query $x \in X$ that minimizes $\left|\sum_{h \in V} p_H(h) \cdot h(x)\right|$. Intuitively these policies myopically attempt to shrink a measure of the version space (i.e., cardinality or the probability mass) as quickly as possible. The former provides an $\mathcal{O}(\log |H|)$-approximation for the worst-case number of queries (Arkin et al., 1993), and the latter provides an $\mathcal{O}(\log \frac{1}{\min_h p_H(h)})$-approximation for the expected number of queries (Kosaraju et al., 1999; Dasgupta, 2004) and a natural generalization of GBS obtains the same guarantees with a larger set of labels (Guillory & Bilmes, 2009). Kosaraju *et al.* also point out that running GBS on a modified prior $p'_H(h) \propto \max\left\{p_H(h), 1/|H|^2 \log |H|\right\}$ is sufficient to obtain an $\mathcal{O}(\log |H|)$-approximation.

Viewed from this perspective of the previous sections, shrinking the version space amounts to "covering" all false hypotheses with stochastic sets (i.e., queries), where query x covers all hypotheses that disagree with the target hypothesis h^* at x. That is, x covers $\{h : h(x) \neq h^*(x)\}$. As in §6, these sets may be correlated in complex ways determined by the set of possible hypotheses. Hence the problem is an adaptive stochastic coverage problem in the same vein as Stochastic Submodular Maximization and Adaptive Viral Marketing, except that it is a variant of the Set Cover problem (where we wish to find the cheapest set achieving full coverage) rather than the Maximum Coverage problem (where we want to maximize coverage subject to a budget constraint). We give general results for this adaptive stochastic coverage variant in the extended version of this paper (Golovin & Krause, 2010). We prove the following result, which improves the constant in front of the $\ln(1/\min_h p_H(h))$ to 1. Chakaravarthy et al. (2007) provide a reduction to Set Cover that, with some parameters slightly tweaked, can be combined with a Set Cover inapproximability result due to Feige (1998) to yield a lower bound of $(1/2-\epsilon)\ln(1/\min_h p_H(h))$ assuming NP $\not\subseteq$ DTIME($n^{\mathcal{O}(\log\log n)}$).

Theorem 7 *In the Bayesian setting in which there is a prior p_H on a finite set of hypotheses H, the generalized binary search algorithm makes $\mathit{OPT}\cdot\left(\ln\left(\frac{1}{\min_h p_H(h)}\right)+1\right)$ queries in expectation to identify a hypothesis drawn from p_H, where OPT is the minimum expected number of queries made by any policy.*

Due to space limitations, we defer the proof of Theorem 7 to the longer version of this paper. The proof relies on the fact that the reduction in version space is adaptive monotone submodular, which we prove in Proposition 8 below. To define the objective function formally, we first need some notation. Define a realization Φ_h for each hypothesis $h \in H$. The ground set is $E = X$, and the outcomes are binary; we define $O = \{-1, 1\}$ instead of using $\{0, 1\}$ to be consistent with our earlier exposition. For all $h \in H$ we set $\Phi_h \equiv h$, meaning $\Phi_h(x) = h(x)$ for all $x \in X$. Given observed labels $\Psi \subset E \times O$, let $V(\Psi)$ denote the version space, i.e., the set of hypotheses for which $h(x) = \Psi(x)$ for all $x \in \mathrm{dom}(\Psi)$. For a set of hypotheses V, let $p_H(V) := \sum_{h\in V} p_H(h)$ denote their total prior probability. Finally, let $\Psi(S, h) = \{(x, h(x)) : x \in S\}$ be the function with domain S that agrees with h on S. We define the objective function by

$$f(S,\Phi_h) \;:=\; 1 - p_H(V(\Psi(S,h))) \;=\; p_H(\{\Phi \,:\, \exists x \in S, \Phi(x) \neq \Phi_h(x)\}) \qquad (11)$$

and use $\mathbb{P}[\Phi_h] = p_H(h)$ for all h. Note that identifying h^* as the target hypothesis corresponds to eliminating everything but h^* from the version space.

Proposition 8 *The version space reduction objective is adaptive monotone and adaptive submodular.*

Proof: Adaptive monotonicity is immediate, as additional queries only remove hypotheses from the version space, and never add to it. We establish the adaptive submodularity of f using the characterization in Proposition 3. Each query x eliminates some subset of hypotheses, and as more queries are performed, the subset of hypotheses eliminated by x cannot grow. More formally, consider the expected marginal contribution of x under two partial realizations Ψ, Ψ' where Ψ is a subrealization of Ψ' (i.e., $\Psi \subset \Psi'$), and $x \notin \mathrm{dom}(\Psi')$. Let $\Psi[e/o]$ be the partial realization with domain $\mathrm{dom}(\Psi) \cup \{e\}$ that agrees with Ψ on its domain, and maps e to o. For each $o \in O$, let $a_o := p(V(\Psi[x/o]))$, $b_o := p(V(\Psi'[x/o]))$. Since a hypotheses eliminated from the version space cannot later appear in the version space, we have $a_o \geq b_o$ for all o. Next, note the expected reduction in version space mass (and hence the expected marginal contribution) due to selecting x given partial realization Ψ is

$$\Delta_\Psi(x) = \sum_{o\in O} a_o \cdot \mathbb{P}[\Phi(x) \neq o \mid \Phi \sim \Psi] = \sum_o a_o\left(\frac{\sum_{o'\neq o} a_{o'}}{\sum_{o'} a_{o'}}\right) = \frac{\sum_{o\neq o'} a_o a_{o'}}{\sum_{o'} a_{o'}} \qquad (12)$$

The corresponding quantity for Ψ' has b_o substituted for a_o in Eq. (12), for each o. Proposition 3 states that proving adaptive submodularity amounts to showing $\Delta_\Psi(x) \geq \Delta_{\Psi'}(x)$. Using Eq. (12), it suffices to show that $\partial\phi/\partial z_o \geq 0$ for each o, where $\phi(\vec{z}) := \left(\sum_{o\neq o'} z_o z_{o'}\right) / \left(\sum_{o'} z_{o'}\right)$ and we assume each $z_o \geq 0$ and $z_o > 0$ for some o. This is because $\partial\phi/\partial z_o \geq 0$ for each o implies that growing the version space in any manner cannot decrease the marginal benefit of query x, and hence shrinking it in any manner cannot increase the marginal benefit of x. The fact $\partial\phi/\partial z_o \geq 0$ for each o can be shown by means of elementary calculus. ■

Extensions to Arbitrary Costs, Multiple Classes, and Approximate Greedy Policies.

This result easily generalizes to handle the multi-class setting (i.e., $|O| \geq 2$), and α-approximate greedy policies, where we lose a factor of α in the approximation factor. As we describe in the extended version of this paper, we can generalize adaptive submodularity to incorporate costs on items, which allows us to extend this result to handle query costs as well. We can therefore recover these extensions of Guillory and Bilmes (2009), while improving the approximation factor for GBS with item costs to $\ln\left(\frac{1}{\min_h p_H(h)}\right)+1$. Guillory and Bilmes also showed how to extend the technique of Kosaraju et al. (1999) to obtain an $\mathcal{O}\left(\log\left(|H|\frac{\max_x c(x)}{\min_x c(x)}\right)\right)$-approximation with costs using a greedy policy, which may be combined with our tighter analysis as well to give a similar result with an improved leading constant. Recently, Gupta et al. (2010) showed how to simultaneously remove the dependence on both costs and probabilities from the approximation ratio. Specifically, within the context of studying an adaptive travelling salesman problem they investigated the *Optimal Decision Tree* problem, which is equivalent to the active learning problem we consider here. Using a clever, more complex algorithm than adaptive greedy, they achieve an $\mathcal{O}(\log|H|)$-approximation in the case of non-uniform costs and general priors.

8 Hardness of Approximation

In this paper, we have developed the notion of adaptive submodularity, which characterizes when certain adaptive stochastic optimization problems are well-behaved in the sense that a simple greedy policy obtains a constant factor or logarithmic factor approximation to the best policy. However, without adaptive submodularity, the Adaptive Stochastic Maximization problem (1) is extremely inapproximable, even with (pointwise) *linear* objective functions (i.e., those where for each Φ, $f : 2^E \times O^E \to \mathbb{R}$ is linear in the first argument): We cannot hope to achieve an $\mathcal{O}(|E|^{1-\varepsilon})$ approximation ratio for this problem, unless the polynomial hierarchy collapses to Σ_2^P. Even worse, we can rule out good bicriteria results under the same assumption.

Theorem 9 *In general, for all (possibly non-constant) $\beta \geq 1$, no polynomial time algorithm for Adaptive Stochastic Maximization with a budget of βk items can approximate the reward of an optimal policy with a budget of only k items to within a multiplicative factor of $\mathcal{O}(|E|^{1-\varepsilon}/\beta)$ for any $\varepsilon > 0$, unless PH $= \Sigma_2^P$. This holds even for pointwise linear f.*

Proof: We construct a hard instance based on the following intuition. We make the algorithm go "treasure hunting". There is a set of t locations $\{0, 1, , \ldots, t-1\}$, there is a treasure at one of these locations, and the algorithm gets unit reward if it finds it, and zero reward otherwise. There are m "maps," each consisting of a cluster of s bits, and each purporting to indicate where the treasure is, and each map is stored in a (weak) secret-sharing way, so that querying few bits of a map reveals nothing about where it says the treasure is. Moreover, all but one of the maps are *fake*, and there is a puzzle indicating which map is the correct one indicating the treasure's location. Formally, a fake map is one which is probabilistically independent of the location of the treasure, conditioned on the puzzle.

Our instance will have three types of elements, $E = E_T \uplus E_M \uplus E_P$, where $|E_T| = t$ encodes where the treasure is, $|E_M| = ms$ encodes the maps, and $|E_P| = n^3$ encodes the puzzle, where m, t, s and n are specified below. All outcomes are binary, $O = \{0, 1\}$. For all $e \in E_M \cup E_P$, $\mathbb{P}[\Phi(e) = 1] = .5$ independently. The conditional distribution $\mathbb{P}[\Phi(E_T) \mid \Phi(E_M \cup E_P)]$ will be deterministic as specified below. Our objective function f is linear, and defined as follows:

$$f(E', \Phi) = |\{e \in E' \cap E_T : \Phi(e) = 1\}|.$$

We now describe the puzzle, which is to compute $i(P) := (\mathrm{perm}(P) \mod p) \mod 2^\ell$ for a suitably sampled random matrix P, and suitable prime p and integer ℓ, where $\mathrm{perm}(P) = \sum_{\sigma \in S_n} \prod_{i=1}^n P_{i\sigma(i)}$ is the permanent of P. We exploit Theorem 1.9 of Feige and Lund (1997) in which they show that if there exist constants $\eta, \delta > 0$ such that a randomized polynomial time algorithm can compute $(\mathrm{perm}(P) \mod p) \mod 2^\ell$ correctly with probability $2^{-\ell}(1 + 1/n^\eta)$, where P is drawn uniformly at random from $\{0, 1, 2, \ldots, p-1\}^{n \times n}$, p is any prime superpolynomial in n, and $\ell \leq p\left(\frac{1}{2} - \delta\right)$, then PH $=$ AM $= \Sigma_2^P$. To encode the puzzle, we fix a prime $p \in [2^{n-2}, 2^{n-1}]$ and use the n^3 bits of $\Phi(E_P)$ to sample $P = P(\Phi)$ (nearly) uniformly at random from $\{0, 1, 2, \ldots, p-1\}^{n \times n}$ as follows. For a matrix $P \in \mathbb{Z}^{n \times n}$, we let $\mathrm{rep}(P) := \sum_{ij} P_{ij} \cdot p^{(i-1)n+(j-1)}$ define a base p representation of P. Note $\mathrm{rep}(\cdot)$ is one-to-one for $n \times n$ matrices with entries in $\mathbb{Z}_p$, so we can define its inverse $\mathrm{rep}^{-1}(\cdot)$. The encoding $P(\Phi)$ interprets the bits $\Phi(E_P)$ as an integer x in $[2^{n^3}]$, and computes $y = x \mod (p^{n^2})$. If $x \leq \left\lfloor 2^{n^3}/p^{n^2} \right\rfloor p^{n^2}$, then $P = \mathrm{rep}^{-1}(y)$. Otherwise, P is the all zero matrix. This latter event occurs with probability at most $p^{n^2}/2^{n^3} \leq 2^{-n^2}$, and in this case we simply suppose the algorithm under consideration finds the treasure and so gets unit reward. This adds 2^{-n^2} to its expected reward. So let us assume from now on that P is drawn uniformly at random.

Next we consider the maps. Partition $E_M = \biguplus_{i=1}^m M_i$ into m maps M_i, each consisting of s items. For each map M_i, partition its items into $s/\log_2 t$ groups of $\log_2 t$ bits each, and let $v_i \in \{0, 1, \ldots, t-1\}$ be the XOR of these groups of bits. We say M_i *points to* v_i as the location of the treasure. A priori, each v_i is uniformly distributed in $\{0, ..., t-1\}$. For a particular realization of $\Phi(E_P \cup E_M)$, define $v(\Phi) := v_{i(P(\Phi))}$. We set $v(\Phi)$ to be the location of the treasure under realization Φ, i.e., we label $E_T = \{e_0, e_1, \ldots, e_{t-1}\}$ and ensure $\Phi(e_j) = 1$ if $j = v_{i(P(\Phi))}$, and $\Phi(e) = 0$ for all other $e \in E_T$. Note the random variable $v = v(\Phi)$ is distributed uniformly at random in $\{0, 1, \ldots, t-1\}$. Note that this still holds if we condition on the realizations of any set of $s/\log_2 t - 1$ items in a map.

Now consider the optimal policy with a budget of $k = n^3 + s + 1$ items to pick. Clearly, its reward can be at most 1. However, given a budget of k, a computationally unconstrained policy can exhaustively sample E_P, solve the puzzle (i.e., compute $i(P)$), read the correct map (i.e., exhaustively sample $M_{i(P)}$), decode the map (i.e., compute $v = v_{i(P)}$), and get the treasure (i.e., pick e_v) thereby obtaining a reward of one.

Now we give an upper bound on the expected reward R of any randomized polynomial time algorithm $\mathcal{A}$ with a budget of βk items, assuming $\Sigma_2^P \neq$ PH. Fix a small constant $\gamma > 0$, and set $s = n^3$ and $m = t = n^{1/\gamma}$. We suppose we give $\mathcal{A}$ the realizations $\Phi(E_M)$ for free. We also replace its budget of

βk items with a budget of βk specifically for map items in E_M and an additional budget of βk specifically for the treasure locations in E_T. Obviously, this can only help it. As noted, if it selects less than $s/\log_2 t$ bits from the map $M_{i(P)}$ indicated by P, the distribution over $v_{i(P)}$ conditioned on those realizations is still uniform. Of course, knowledge of v_i for $i \neq i(P)$ is useless for getting reward. Hence $\mathcal{A}$ can try at most $\beta k \log_2(t)/s = o(\beta k)$ maps in an attempt to find $M_{i(P)}$. Note that if we have a randomized algorithm which given a random P drawn from $\{0, 1, 2, \ldots, p-1\}^{n \times n}$ always outputs a set S of integers of size α such that $\mathbb{P}[i(P) \in S] \geq q$, then we can use it to construct a randomized algorithm that, given P, outputs an integer x such that $\mathbb{P}[i(P) = x] \geq q/\alpha$, simply by running the first algorithm and then selecting a random element of S. If $\mathcal{A}$ does not find $M_{i(P)}$, the distribution on the treasure's location is uniform given its knowledge. Hence it's budget of βk treasure locations can only earn it expected reward at most $\beta k/t$. Armed with these observations and Theorem 1.9 of Feige and Lund (1997) and our complexity theoretic assumptions, we infer $\mathbb{E}[R] \leq o(\beta k) \cdot 2^{-\ell}(1 + 1/n^{\eta}) + \beta k/t + 2^{-n^2}$. Since $s = n^3$ and $m = t = n^{1/\gamma}$ and $\gamma = \Theta(1)$ and $\eta = 1$ and $\ell = \log_2 m$ and $k = n^3 + s + 1 = 2n^3 + 1$, we have

$$\mathbb{E}[R] \leq \frac{\beta k}{t}(1 + o(1)) = 2\beta n^{3-1/\gamma}(1 + o(1)).$$

Next note that $|E| = t + ms + n^3 = n^{3+1/\gamma}(1 + o(1))$. Straightforward algebra shows that in order to ensure $\mathbb{E}[R] = o(\beta/|E|^{1-\varepsilon})$, it suffices to choose $\gamma \leq \varepsilon/6$. Thus, under our complexity theoretic assumptions, any polynomial time randomized algorithm $\mathcal{A}$ with budget βk achieves at most $o(\beta/|E|^{1-\varepsilon})$ of the value obtained by the optimal policy with budget k, so the approximation ratio is $\omega(|E|^{1-\varepsilon}/\beta)$. ■

9 Conclusions

In this paper, we introduced the concept of *adaptive submodularity*, generalizing submodular set functions to adaptive policies. Our generalization is based on a natural adaptive analog of the diminishing returns property well understood for set functions. In the special case of deterministic distributions, adaptive submodularity reduces to the classical notion of submodular set functions. We proved that guarantees carried by the non-adaptive greedy algorithm for submodular set functions generalize to a natural adaptive greedy algorithm in the case of adaptive submodular functions. We illustrated the usefulness of the concept by giving several examples of adaptive submodular objectives arising in diverse applications including sensor placement, viral marketing and pool-based active learning. Proving adaptive submodularity for these problems allowed us to recover existing results in these applications as special cases and lead to natural generalizations. We believe that our results provide an interesting step in the direction of exploiting structure to solve complex stochastic optimization problems under partial observability.

Acknowledgments

This research was partially supported by ONR grant N00014-09-1-1044, NSF grant CNS-0932392, NSF grant IIS-0953413, a gift by Microsoft Corporation, an Okawa Foundation Research Grant, and by the Caltech Center for the Mathematics of Information.

References

Arkin, E. M., Meijer, H., Mitchell, J. S. B., Rappaport, D., & Skiena, S. S. (1993). Decision trees for geometric models. *Proc. of Symposium on Computational Geometry* (pp. 369–378). New York, NY, USA: ACM.

Asadpour, A., Nazerzadeh, H., & Saberi, A. (2008). Stochastic submodular maximization. *WINE '08: Proc. of the 4th International Workshop on Internet and Network Economics* (pp. 477–489). Berlin, Heidelberg: Springer-Verlag.

Chakaravarthy, V. T., Pandit, V., Roy, S., Awasthi, P., & Mohania, M. (2007). Decision trees for entity identification: Approximation algorithms and hardness results. *In Proceedings of the ACM-SIGMOD Symposium on Principles of Database Systems*.

Dasgupta, S. (2004). Analysis of a greedy active learning strategy. *NIPS: Advances in Neural Information Processing Systems* (pp. 337–344). MIT Press.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM*, *45*, 634 – 652.

Feige, U., & Lund, C. (1997). On the hardness of computing the permanent of random matrices. *Computational Complexity*, *6*, 101–132.

Fujishige, S. (1991). *Submodular functions and optimization*. No. 47. North Holland, Amsterdam: Annals of Discrete. Mathematics. 2nd edition.

Garey, M. R., & Graham, R. L. (1974). Performance bounds on the splitting algorithm for binary testing. *Acta Inf.*, *3*, 347–355.

Goemans, M. X., & Vondrák, J. (2006). Stochastic covering and adaptivity. *LATIN* (pp. 532–543). Springer.

Golovin, D., & Krause, A. (2010). Adaptive submodularity: A new approach to active learning and stochastic optimization. *CoRR*, *abs/1003.3967*.

Goundan, P. R., & Schulz, A. S. (2007). *Revisiting the greedy approach to submodular set function maximization* (Technical Report). Massachusetts Institute of Technology.

Guillory, A., & Bilmes, J. (2009). Average-case active learning with costs. *The 20th International Conference on Algorithmic Learning Theory*. University of Porto, Portugal.

Guillory, A., & Bilmes, J. (2010). Interactive submodular set cover. *To appear in Proc. International Conference on Machine Learning (ICML)*.

Gupta, A., Krishnaswamy, R., Nagarajan, V., & Ravi, R. (2010). Approximation algorithms for optimal decision trees and adaptive TSP problems. *To appear in Proc. International Colloquium on Automata, Languages and Programming (ICALP)*.

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 137–146). New York, NY, USA: ACM.

Kosaraju, S. R., Przytycka, T. M., & Borgstrom, R. S. (1999). On an optimal split tree problem. *Proc. of the 6th Intl. Workshop on Algorithms and Data Structures* (pp. 157–168). London, UK: Springer-Verlag.

Krause, A., & Guestrin, C. (2007). Near-optimal observation selection using submodular functions. *AAAI Nectar track* (pp. 1650–1654).

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

McCallum, A., & Nigam, K. (1998). Employing EM and pool-based active learning for text classification. *ICML* (pp. 350–358). Morgan Kaufmann.

Minoux, M. (1978). Accelerated greedy algorithms for maximizing submodular set functions. *Proc. of the 8th IFIP Conference on Optimization Techniques* (pp. 234–243). Springer.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, *14*, 265–294.

Nowak, R. (2009). Noisy generalized binary search. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams and A. Culotta (Eds.), *Advances in neural information processing systems 22*, 1366–1374.

Schrijver, A. (2003). *Combinatorial optimization : polyhedra and efficiency*. Volume B, Part IV, Chapters 39-49. Springer.