Submodular Learning and Covering with Response-Dependent Costs

Sivan Sabato

Ben-Gurion University of the Negev

Abstract. We consider interactive learning and covering problems, in a setting where actions may incur different costs, depending on the response to the action. We propose a natural greedy algorithm for response-dependent costs. We bound the approximation factor of this greedy algorithm in active learning settings as well as in the general setting. We show that a different property of the cost function controls the approximation factor in each of these scenarios. We further show that in both settings, the approximation factor of this greedy algorithm is near-optimal among all greedy algorithms. Experiments demonstrate the advantages of the proposed algorithm in the response-dependent cost setting.

Keywords: Interactive learning, submodular functions, outcome costs

1 Introduction

We consider interactive learning and covering problems, a term introduced in [7]. In these problems, there is an algorithm that interactively selects actions and receives a response for each action. Its goal is to achieve an objective, whose value depends on the actions it selected, their responses, and the state of the world. The state of the world, which is unknown to the algorithm, also determines the response to each action. The algorithm incurs a cost for every action it performs. The goal is to have the total cost incurred by the algorithm as low as possible.

Many real-world problems can be formulated as interactive learning and covering problems. These include pool-based active learning problems [12, 2], maximizing the influence of marketing in a social network [7], interactive sensor placement [4] and document summarization [11] with interactive user feedback. Interactive learning and covering problems cannot be solved efficiently in general [14, 19]. Nevertheless, many such problems can be solved near-optimally by efficient algorithms, when the functions that map the sets of actions to the total reward are *submodular*. It has been shown in several settings, that a simple greedy algorithm pays a near-optimal cost when the objective function is submodular (e.g., [7, 4, 1]). Many problems naturally lend themselves to a submodular formulation. These include covering objectives, objectives promoting diversity [13] and active learning [2, 4, 7, 6].

Interactive learning and covering problems have so far been studied mainly under the assumption that the cost of the action is known to the algorithm before the action is taken. In this work we study the setting in which the costs of actions depend on the outcome of the action, which is only revealed by the observed response. This is the case in many real-world scenarios. For instance, consider an active learning problem, where the goal is to learn a classifier that predicts which patients should be administered a specific drug. Each action in the process of learning involves administering the drug to a patient and observing the effect. In this case, the cost (poorer patient health) is higher if the patient suffers adverse effects. Similarly, when marketing in a social network, an action involves sending an ad to a user. If the user does not like the ad, this incurs a higher cost (user dissatisfaction) than if they like the ad.

We study the achievable approximation guarantees in the setting of response-dependence costs, and characterize the dependence of this approximation factor on the properties of the cost function. We propose a natural generalization of the greedy algorithm of [7] to the response-dependent setting, and provide two approximation guarantees. The first guarantee holds whenever the algorithm's objective describes an active learning problem. We term such objectives *learning objectives*. The second guarantee holds for general objectives, under a mild condition. In each case, the approximation guarantees depend on a property of the cost function, and we show that this dependence is necessary for any greedy algorithm. Thus, this fully characterizes the relationship between the cost function and the approximation guarantee achievable by a greedy algorithm. We further report experiments that demonstrate the achieved cost improvement.

Response-dependent costs has been previously studied in specific cases of active learning, assuming there are only two possible labels [15–18]. In [8] this setting is also mentioned in the context of active learning. Our work is more general: First, it addresses general objective functions and not only specific active learning settings. Our results indicate that the active learning setting and the general setting are inherently different. Second, it is not limited to settings with two possible responses. As we show below, previous guarantees for two responses do not generalize to tight guarantees for cases with more than two responses. We thus develop new proof techniques that allow deriving these tighter bounds.

2 Definitions and Preliminaries

For an integer n, denote $[n] := \{1, \dots, n\}$. A set function $f : 2^{\mathcal{Z}} \to \mathbb{R}$ is *monotone* (non-decreasing) if $\forall A \subseteq B \subseteq \mathcal{Z}$, $f(A) \leq f(B)$. Let $\mathcal{Z}$ be a domain, and let $f : 2^{\mathcal{Z}} \to \mathbb{R}_+$ be a set function. Define, for any $z \in \mathcal{Z}, A \subseteq \mathcal{Z}$, $\delta_f(z \mid A) := f(A \cup \{z\}) - f(A)$. f is *submodular* if $\forall z \in \mathcal{Z}, A \subseteq B \subseteq \mathcal{Z}$, $\delta_f(z \mid A) \geq \delta_f(z \mid B)$.

Assume a finite domain of actions $\mathcal{X}$ and a finite domain of responses $\mathcal{Y}$. For simplicity of presentation, we assume that there is a one-to-one mapping between world states and mappings from actions to their responses. Thus the states of the world are represented by the class of possible mappings $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. Let $h^* \in \mathcal{H}$ be the true, unknown, mapping from actions to responses. Let $S \subseteq \mathcal{X} \times \mathcal{Y}$ be a set of action-response pairs.

We consider algorithms that iteratively select a action $x \in \mathcal{X}$ and get the response $h^*(x)$, where $h^* \in \mathcal{H}$ is the true state of the world, which is unknown to the algorithm. For an algorithm $\mathcal{A}$, let $S^h[\mathcal{A}]$ be the set of pairs collected by $\mathcal{A}$ until termination if $h^* = h$. Let $S^h_t[\mathcal{A}]$ be the set of pairs collected by $\mathcal{A}$ in the first t iterations if $h^* = h$. In each iteration, $\mathcal{A}$ decides on the next action to select based on responses to previous actions, or it decides to terminate. $\mathcal{A}(S) \in \mathcal{X} \cup \{\perp\}$ denotes the action that $\mathcal{A}$ selects after observing the set of pairs S, where $\mathcal{A}(S) = \perp$ if $\mathcal{A}$ terminates after observing S.

Each time the algorithm selects an action and receives a response, it incurs a cost, captured by a cost function $\mathfrak{cost} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$. If $x \in \mathcal{X}$ is selected and the response $y \in \mathcal{Y}$ is received, the algorithm pays $\mathfrak{cost}(x, y)$. Denote $\mathfrak{cost}(S) = \sum_{(x,y)\in S} \mathfrak{cost}(x, y)$. The total cost of a run of the algorithm when the state of the world is h^*, is thus $\mathfrak{cost}(S^{h^*}[\mathcal{A}])$. For a given $\mathcal{H}$, define the *worst-case cost* of $\mathcal{A}$ by $\mathfrak{cost}(\mathcal{A}) := \max_{h\in\mathcal{H}} \mathfrak{cost}(S^h[\mathcal{A}])$. Let $Q > 0$ be a threshold, and let $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$ be a monotone non-decreasing submodular objective function. We assume that the goal of the interactive algorithm is to collect pairs S such that $f(S) \geq Q$, while minimizing $\mathfrak{cost}(\mathcal{A})$.

Guillory and Bilmes [7] consider a setting in which instead of a single global f, there is a set of monotone non-decreasing objective functions $\mathcal{F}_\mathcal{H} = \{f_h : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+ \mid h \in \mathcal{H}\}$, and the value $f_h(S)$, for $S \subseteq \mathcal{X} \times \mathcal{Y}$, represents the reward obtained by the algorithm if $h^* = h$. They show that obtaining $f_{h^*}(S) \geq Q$ is equivalent to obtaining $\bar{F}(S) \geq Q$, where $\bar{F} : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$ is defined by

$$\bar{F}(S) := \frac{1}{|\mathcal{H}|}\Big(Q|\mathcal{H} \setminus \mathrm{VS}(S)| + \sum_{h\in\mathrm{VS}(S)} \min(Q, f_h(S))\Big). \tag{1}$$

Here $\mathrm{VS}(S)$ is the *version space* induced by S on $\mathcal{H}$, defined by $\mathrm{VS}_\mathcal{H}(S) = \{h \in \mathcal{H} \mid \forall(x, y) \in S, y = h(x)\}$. It is shown in [7] that if all the functions in $\mathcal{F}_\mathcal{H}$ are monotone and submodular then so is $\bar{F}$. Thus our setting of a single objective function can be applied to the setting of [7] as well.

Let $\alpha \geq 1$. An interactive algorithm $\mathcal{A}$ is an α*-approximate greedy algorithm* for *utility function* $u : \mathcal{X} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$, if the following holds: For all $S \subseteq \mathcal{X} \times \mathcal{Y}$, if $f(S) \geq Q$ then $\mathcal{A}(S) = \perp$, and otherwise, $\mathcal{A}(S) \in \mathcal{X}$ and $u(\mathcal{A}(S), S) \geq \frac{1}{\alpha} \max_{x\in\mathcal{X}} u(x, S)$. As shown below, consistently with previous works, (e.g. [4]), competitive guarantees are better for α-approximate-greedy algorithms with $\alpha = 1$ or α close to 1. However, due to efficiency of computation or other practical considerations, it is not always feasible to implement a 1-greedy algorithm. Thus, for full generality, we analyze also α-greedy algorithms for $\alpha > 1$. Let $\mathrm{OPT} := \min_\mathcal{A} \mathfrak{cost}(\mathcal{A})$, where the minimum is taken over all interactive $\mathcal{A}$ that obtain $f(S) \geq Q$ at termination, for all for all possible $h^* \in \mathcal{H}$. If no such $\mathcal{A}$ exist, define $\mathrm{OPT} = \infty$.

In [7] it is assumed that costs are not response-dependent, thus $\mathfrak{cost}(x, y) \equiv \mathfrak{cost}(x)$, and a greedy algorithm is proposed, based on the following utility function:

$$u(x, S) := \frac{\delta_{\bar{F}}((x, h(x)) \mid S)}{\mathfrak{cost}(x)}. \tag{2}$$

It is shown that for integral functions , this algorithm obtains an integer Q with a worst-case cost of at most $\mathrm{GCC}(\ln(Q|\mathcal{H}|)+1)\cdot\mathrm{OPT}$, where GCC is a lower bound on OPT. In [4], a different greedy algorithm and analysis guarantees a worst-case cost of $\alpha(\ln(Q)+1)\cdot\mathrm{OPT}$ for adaptive submodular objectives and α-approximate greedy algorithms. It is well known that the factor of $\ln(Q)$ cannot be substantially improved by an efficient algorithm, even for non-interactive problems [19, 3].

The results of [7] can be trivially generalized to the response-dependent cost setting using the *cost ratio* of the problem:

$$r_{\mathtt{cost}} := \max_{x\in\mathcal{X}} \frac{\max_{y\in\mathcal{Y}} \mathtt{cost}(x,y)}{\min_{y\in\mathcal{Y}} \mathtt{cost}(x,y)}.$$

Consider a generalized version of u:

$$u(x,S) := \frac{\delta_{\bar{F}}((x,h(x)) \mid S)}{\min_{y\in\mathcal{Y}} \mathtt{cost}(x,y)}. \tag{3}$$

Setting $\overline{\mathtt{cost}}(x) := \min_{y\in\mathcal{Y}} \mathtt{cost}(x,y)$, we have $\mathtt{cost} \le r_{\mathtt{cost}} \cdot \overline{\mathtt{cost}}$. Using this fact, it is easy to derive an approximation guarantee of $r_{\mathtt{cost}}\cdot\mathrm{OPT}(\ln(Q|\mathcal{H}|)+1)$, for a greedy algorithm which uses the utility function in Eq. (3) with a response-dependent cost, or $r_{\mathtt{cost}}\cdot\alpha(\ln(Q)+1)\mathrm{OPT}$ when applied to the setting of [4]. However, in this work we show that this trivial derivation is loose, since our new approximation bounds can be finite even if $r_{\mathtt{cost}}$ is infinite.

3 A greedy algorithm for response-dependent costs

We provide approximation guarantee for two types of objective functions. The first type captures active learning settings, while the second type is more general. Our results show that objective functions for active learning have better approximation guarantees than general objective functions. For both types of objective functions, we analyze a greedy algorithm that selects an element maximizing (or approximately maximizing) the following utility function:

$$u^f(x,S) := \min_{h\in\mathrm{VS}(S)} \frac{\delta_{\min(f,Q)}((x,h(x)) \mid S)}{\mathtt{cost}(x,h(x))}.$$

Note that $u^{\bar{F}}$ is equal to the function u defined in Eq. (3). We employ the following standard assumption in our results (see e.g. [5]):

Assumption 1. *Let $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$, $Q>0$, $\eta>0$. Assume that f is submodular and monotone, $f(\emptyset)=0$, and that for any $S\subseteq \mathcal{X}\times\mathcal{Y}$, if $f(S)\ge Q-\eta$ then $f(S)\ge Q$.*

In Section 3.1 we show an approximation guarantee for objectives meant for active learning, which we term *learning objectives*. In Section 3.2 we consider general monotone submodular objective functions. Our guarantees hold for objective functions f that satisfy the following property, which we term *consistency-aware*. This property requires that the function gives at least Q to any set of action-response pairs that are inconsistent with $\mathcal{H}$.

Definition 1. *A function* $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$ *is* consistency-aware *for threshold* $Q > 0$ *if for all* $S \subseteq \mathcal{X} \times \mathcal{Y}$ *such that* $\mathrm{VS}_{\mathcal{H}}(S) = \emptyset$*,* $f(S) \geq Q$*.*

Note that the definition is concerned with the value of f only on inconsistent sets S, which the algorithm never encounters. Therefore, it suffices that there exist an extension of f to these sets that is consistent with all the other requirements from f. The function $\bar{F}$ defined in Eq. (1) is consistency-aware. In addition, a similar construction to $\bar{F}$ with non-uniform weights for mappings is also consistency-aware. Such a construction is sometimes more efficient to compute than the uniform-weight construction. For instance, as shown in [6], non-uniform weights allow a more efficient computation when the mappings represent linear classifiers with a margin. In general, any objective f can be made consistency aware using a simple transformation such as $\bar{F}$. Thus our results are relevant to a diverse class of problems.

3.1 Guarantees for learning objectives

Active learning is an important special case of interactive learning. In active learning, the only goal is to discover information on the identity of h^*. We term functions that represent such a goal *learning objectives*.

Definition 2. *A function* $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$ *is a* learning objective *for* $\mathcal{H}$ *if* $f(S) = g(\mathrm{VS}_{\mathcal{H}}(S))$ *where* g *is a monotone non-increasing function.*

It is easy to see that all learning objectives $S \mapsto f(S)$ are monotone non-decreasing in S. In many useful cases, they are also submodular. In noise-free active learning, where the objective is to exactly identify the correct mapping h^*, one can use the learning objective $f(S) := 1 - |\mathrm{VS}(S)|/|\mathcal{H}|$, with $Q = 1 - 1/|\mathcal{H}|$. This is the *version space reduction* objective function [4, 7]. In [5] noise-aware active learning and its generalization to the problem of Equivalence Class Determination is considered. In this generalization, there is some partition of $\mathcal{H}$, and the goal is to identify the class to which h^* belongs. The objective function proposed by [5] measures the weight of pairs in $\mathrm{VS}(S)$ in which each mapping belongs to a different class. This function is also a learning objective. In [1] the *total generalized version space reduction* function is proposed. This function is also a learning objective. More generally, consider a set of structures $\mathcal{G} \subseteq 2^{\mathcal{H}}$, where the goal is to disqualify these structures from the version space, by proving that at least one of the mappings in this structure cannot be the true h^*. In this case one can define the submodular learning objective $f(S) := w(\mathcal{G}) - w(\mathcal{G} \cap 2^{\mathrm{VS}(S)})$, where w is a modular weight function on $\mathcal{G}$, and $Q = w(\mathcal{G})$. For instance, if $\mathcal{G}$ is the set of pairs from different equivalence classes in $\mathcal{H}$, this is the Equivalence Class Determination objective. If $\mathcal{G}$ is a set of triplets from different equivalence classes, this encodes an objective of reducing the uncertainty on the identity of h^* to at most two equivalence classes.

We show that for learning objectives, the approximation factor for a greedy algorithm that uses u^f depends on a new property of the cost function, which we term the *second-smallest cost ratio*, denoted by $r^{[2]}_{\mathtt{cost}}$. For $x \in \mathcal{X}$, let $\phi(x)$ be the second-smallest value in the multiset $\{\mathtt{cost}(x, y) \mid y \in \mathcal{Y}\}$. Define

$$r^{[2]}_{\mathtt{cost}} := \max_{x\in\mathcal{X}, y\in\mathcal{Y}} \frac{\mathtt{cost}(x, y)}{\phi(x)}.$$

Theorem 1. *Let $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+, Q > 0, \eta > 0$ such that Assumption 1 holds. Let $\mathcal{A}$ be an α-approximate greedy algorithm for the utility function u^f. If f is a learning objective, then $\mathtt{cost}(\mathcal{A}) \leq r^{[2]}_{\mathtt{cost}} \cdot \alpha(\ln(Q/\eta) + 1)\mathrm{OPT}$.*

The ratio between the trivial bound that depends on the cost ratio $r_{\mathtt{cost}}$, mentioned in Section 2, and this new bound, is $r_{\mathtt{cost}}/r^{[2]}_{\mathtt{cost}}$, which is unbounded in the general case: for instance, if each action has one response which costs 1 and the other responses cost $M \gg 1$, then $r_{\mathtt{cost}} = M$ but $r^{[2]}_{\mathtt{cost}} = 1$. Whenever $|\mathcal{Y}| = 2$, $r^{[2]}_{\mathtt{cost}} = 1$. Thus, the approximation factor of the greedy algorithm for any binary active learning problem is independent of the cost function. This coincides with the results of [17, 18] for active learning with binary labels. If $|\mathcal{Y}| > 2$, then the bound is smallest when $r^{[2]}_{\mathtt{cost}} = 1$, which would be the case if for each action there is one preferred response which has a low cost, while all other responses have the same high cost. For instance, consider a marketing application, in which the action is to recommend a product to a user, and the response is either buying the product (a preferred response), or not buying it, in which case additional feedback could be provided from the user, but the cost (user dissatisfaction) remains the same regardlesss of that feedback.

To prove Theorem 1, we use the following property of learning objectives: For such objectives, there exists an optimal algorithm (that is, one that obtains OPT) that only selects actions for which at least two responses are possible given the action-response pairs observed so far. Formally, we define *bifurcating* algorithms. Denote the set of possible responses for x given the history S by $\mathcal{Y}_{\mathcal{H}}(x, S) := \{h(x) \mid h \in \mathrm{VS}_{\mathcal{H}}(S)\}$. We omit the subscript $\mathcal{H}$ when clear from context.

Definition 3. *An interactive algorithm $\mathcal{A}$ is* bifurcating *for $\mathcal{H}$ if for all t and $h \in \mathcal{H}$, $|\mathcal{Y}_{\mathcal{H}}(\mathcal{A}(S^h_t[\mathcal{A}]), S^h_t[\mathcal{A}])| \geq 2$.*

Lemma 1. *For any learning objective f for $\mathcal{H}$ with an optimal algorithm, then there exists a bifurcating optimal algorithm for $f, \mathcal{H}$.*

Proof. Let $\mathcal{A}$ be an optimal algorithm for f. Suppose there exists some t, h such that $\mathcal{Y}(x_0, S^h_{t-1}[\mathcal{A}]) = \{y_0\}$ for some $y_0 \in \mathcal{Y}$, where $x_0 := \mathcal{A}(S^h_{t-1}[\mathcal{A}])$. Let $\mathcal{A}'$ be an algorithm that selects the same actions as $\mathcal{A}$, except that it skips the action x_0 it if has collected the pairs $S^h_{t-1}[\mathcal{A}]$. That is, $\mathcal{A}'(S) = \mathcal{A}(S)$ for $S \not\supseteq S^h_{t-1}[\mathcal{A}]$, and $\mathcal{A}'(S) = \mathcal{A}(S \cup \{(x_0, y_0)\})$ for $S \supseteq S^h_{t-1}$. Since $\mathrm{VS}(S) = \mathrm{VS}(S \cup \{(x_0, y_0)\})$, and $\mathcal{A}$ is a learning objective, $\mathcal{A}'$ obtains Q as well, at the same cost of $\mathcal{A}$ or less. By repeating this process a finite number of steps, we can obtain an optimal algorithm for $\mathcal{H}$ which is bifurcating. $\square$

The following lemma is the crucial step in proving Theorem 1, and will also be used in the proof for the more general case below. The lemma applies to general consistency-aware functions. It can be used for learning objectives, because all learning objectives with a finite OPT are consistency-aware: Suppose that f is a learning objective, and let $S \subseteq \mathcal{X} \times \mathcal{Y}$ such that $\mathrm{VS}_{\mathcal{H}}(S) = \emptyset$. For any $h \in \mathcal{H}$, denote $S_*^h := \{(x, h(x)) \mid x \in \mathcal{X}\}$. We have $\mathrm{VS}(S_*^h) \supseteq \mathrm{VS}(S)$, therefore, since f is a learning objective, $f(S) \geq f(S_*^h)$. Since OPT is finite, $f(S_*^h) \geq Q$. Therefore $f(S) \geq Q$. Thus f is consistency-aware.

Lemma 2. *Let f, Q, η which satisfy Assumption 1 such that f is consistency-aware. Let $\mathcal{A}$ be an interactive algorithm that obtains $f(S) \geq Q$ at termination. Let $\gamma = r_{\mathfrak{cost}}^{[2]}$ if $\mathcal{A}$ is bifurcating, and let $\gamma = r_{\mathfrak{cost}}$ otherwise. then*

$$\exists x \in \mathcal{X} \text{ s.t. } u^f(x, \emptyset) \geq \frac{Q}{\gamma \cdot \mathfrak{cost}(\mathcal{A})}.$$

Proof. Denote for brevity $\delta \equiv \delta_{\min(f,Q)}$. Define $\bar{\mathcal{H}} := \mathcal{Y}^{\mathcal{X}}$. Consider an algorithm $\bar{\mathcal{A}}$ such that for any S that is consistent with some $h \in \mathcal{H}$ (that is $\mathrm{VS}_{\mathcal{H}}(S) \neq \emptyset$), $\bar{\mathcal{A}}(S) = \mathcal{A}(S)$, and $\bar{\mathcal{A}}(S) = \perp$ otherwise. Since f is consistency-aware, we have $f(S^h[\bar{\mathcal{A}}]) \geq Q$ for all $h \in \bar{\mathcal{H}}$.

Consider a run of $\bar{\mathcal{A}}$, and denote the pair in iteration t of this run by (x_t, y_t). Denote $S_t = \{(x_i, y_i) \mid i \leq t\}$. Choose the run such that in each iteration t, the response y_t is in $\mathrm{argmin}_{y \in \mathcal{Y}} \delta(x_t, y \mid S_{t-1})/\mathfrak{cost}(x_t, y)$. Let T be the length of the run until termination. Denote $\psi := \max_{h \in \bar{\mathcal{H}}} \mathfrak{cost}(S^h[\bar{\mathcal{A}}])$, the worst-case cost of $\bar{\mathcal{A}}$ over $\bar{\mathcal{H}}$. We have

$$\begin{aligned} Q/\psi \leq f(S_T)/\mathfrak{cost}(S_T) &= \frac{\sum_{t \in [T]}(f(S_t) - f(S_{t-1}))}{\sum_{t \in [T]} \mathfrak{cost}(x_t, y_t)} \\ &= \frac{\sum_{t \in [T]} \delta((x_t, y_t) \mid S_{t-1})}{\sum_{t \in [T]} \mathfrak{cost}(x_t, y_t)} \leq \max_{t \in [T]} \left(\delta((x_t, y_t) \mid S_{t-1})/\mathfrak{cost}(x_t, y_t)\right), \end{aligned}$$

where we used $f(\emptyset) = 0$ in the second line. Thus there exists some $t \in [T]$ such that $Q/\psi \leq \delta((x_t, y_t) \mid S_{t-1})/\mathfrak{cost}(x_t, y_t)$. Therefore

$$\begin{aligned} u(x_t, \emptyset) &= \min_{y \in \mathcal{Y}} \delta((x_t, y) \mid \emptyset)/\mathfrak{cost}(x_t, y) \geq \min_{y \in \mathcal{Y}} \delta((x_t, y) \mid S_{t-1})/\mathfrak{cost}(x_t, y) \\ &= \delta((x_t, y_t) \mid S_{t-1})/\mathfrak{cost}(x_t, y_t) \geq Q/\psi. \end{aligned} \tag{4}$$

The second line follows from the submodularity of f. The third line follows from the definition of y_t. To prove the claim, we have left to show that $\psi \leq r^{[2]} \cdot \mathfrak{cost}(\mathcal{A})$. Consider again a run of $\bar{\mathcal{A}}$. If all observed pairs are consistent with some $h \in \mathcal{H}$, $\bar{\mathcal{A}}$ and $\mathcal{A}$ behave the same. Hence $\mathfrak{cost}(S^h[\bar{\mathcal{A}}]) = \mathfrak{cost}(S^h[\mathcal{A}])$. Now, consider $h \in \bar{\mathcal{H}} \setminus \mathcal{H}$. By the definition of $\bar{\mathcal{A}}$, $S^h[\bar{\mathcal{A}}]$ is a prefix of $S^h[\mathcal{A}]$. Let $T = |S^h[\bar{\mathcal{A}}]|$ be the number of iterations until $\bar{\mathcal{A}}$ terminates. Then $S_{T-1}^h[\bar{\mathcal{A}}]$ is consistent with some $h' \in \mathcal{H}$.

Let x_T be the action that $\mathcal{A}$ and $\bar{\mathcal{A}}$ select at iteration T, and let $h' \in \mathcal{H}$ which is consistent with $S_{T-1}^h[\bar{\mathcal{A}}]$, and incurs the maximal possible cost in iteration T. Formally, h' satisfies $h'(x_T) \in \mathrm{argmax}_{y \in \mathcal{Y}_{\mathcal{H}}(x_T, S_{T-1}^h[\mathcal{A}])} \mathfrak{cost}(x_T, y)$.

Now, compare the run of $\bar{\mathcal{A}}$ on h to the run of $\mathcal{A}$ on h'. In the first $T-1$ iterations, the algorithms observe the same pairs. In iteration T, they both select x_T. $\bar{\mathcal{A}}$ observes $h(x_T)$, while $\mathcal{A}$ observes $h'(x_T)$. $\bar{\mathcal{A}}$ terminates after iteration T. Hence $\mathtt{cost}(S^h[\bar{\mathcal{A}}]) = \mathtt{cost}(S^h_{T-1}[\mathcal{A}]) + \mathtt{cost}(x_T, h(x_T)) = \mathtt{cost}(S^{h'}_T[\mathcal{A}]) - \mathtt{cost}(x_T, h'(x_T)) + \mathtt{cost}(x_T, h(x_T))$. Consider two cases: (a) $\mathcal{A}$ is not bifurcating. Then $\gamma = r$, and so $\mathtt{cost}(x_T, h(x_T)) \leq \gamma\mathtt{cost}(x_T, h'(x_T))$. (b) $\mathcal{A}$ is bifurcating. Then there are at least two possible responses in $\mathcal{Y}_{\mathcal{H}}(x_T, S^h_{T-1}[\mathcal{A}])$. Therefore $\mathtt{cost}(x_T, h'(x_T)) \geq \phi(x_T)$. By the definition of $r^{[2]}_{\mathtt{cost}}$, $\mathtt{cost}(x_T, h(x_T)) \leq r^{[2]}_{\mathtt{cost}} \cdot \phi(x_T)$. Therefore $\mathtt{cost}(x_T, h(x_T)) \leq r_{\mathtt{cost}}\mathtt{cost}(x_T, h'(x_T))) = \gamma\mathtt{cost}(x_T, h'(x_T)))$.

In both cases, $\mathtt{cost}(x_T, h(x_T)) - \mathtt{cost}(x_T, h'(x_T)) \leq (\gamma - 1)\mathtt{cost}(x_T, h'(x_T))$. Therefore $\mathtt{cost}(S^h[\bar{\mathcal{A}}]) \leq \mathtt{cost}(S^{h'}_T[\mathcal{A}]) + (\gamma-1)\mathtt{cost}(x_T, h'(x_T)) \leq \gamma\mathtt{cost}(S^{h'}_T[\mathcal{A}])$, where the last inequality follows since $\mathtt{cost}(S^{h'}_T[\mathcal{A}]) \leq \mathtt{cost}(S^{h'}_T[\mathcal{A}])$. Thus for all $h \in \bar{\mathcal{H}}$, $\mathtt{cost}(S^h[\bar{\mathcal{A}}]) \leq \gamma \cdot \mathtt{cost}(\mathcal{A})$, hence $\psi \leq \gamma \cdot \mathtt{cost}(\mathcal{A})$. Combining this with Eq. (4), the proof is concluded.

□

In the proof of Theorem 1 we further use the following lemmas, which can be proved using standard techniques (see e.g. [7, 4]). The proofs are omitted due to lack of space,

Lemma 3. *Let $\beta, \alpha \geq 1$. Let f, Q, η such that Assumption 1 holds. If for all $S \subseteq \mathcal{X} \times \mathcal{Y}$, $\max_{x\in\mathcal{X}} u^f(x, S) \geq \frac{Q - f(S)}{\beta\mathrm{OPT}}$, then for any α-approximate greedy algorithm with u^f, $\mathtt{cost}(\mathcal{A}) \leq \alpha\beta(\log(Q/\eta) + 1)\mathrm{OPT}$.*

Lemma 4. *Let f, Q, η such that Assumption 1 holds and f is consistency-aware. Let $S \subseteq \mathcal{X} \times \mathcal{Y}$. Define $f' : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$ by $f'(T) := f(T \cup S) - f(S)$. Let $Q' = Q - f(S)$. Then*

1. *f' is submodular, monotone and consistency-aware, with $f'(\emptyset) = 0$.*
2. *Let $\mathcal{A}$ be an interactive algorithm for f', Q'. Let $\beta \geq 1$. If $\max_{x\in\mathcal{X}} u^{f'}(x, \emptyset) \geq \frac{Q'}{\beta\mathrm{OPT}'}$, where OPT' is the optimal cost for f', Q', then for any $S \subseteq \mathcal{X} \times \mathcal{Y}$, $\max_{x\in\mathcal{X}} u^f(x, S) \geq \frac{Q - f(S)}{\beta\mathrm{OPT}}$.*

Proof (of Theorem 1). Let f', Q', OPT' as in Lemma 4, and let $\mathcal{A}^*$ be an optimal algorithm for f', Q'. Let $\mathcal{A}^*$ be an optimal algorithm for f', Q'. Since f is a learning objective, then so is f', and by Lemma 1 we can choose $\mathcal{A}^*$ to be bifurcating. Combining this with the first part of Lemma 4, the conditions of Lemma 2 hold. Therefore $u^{f'}(x, \emptyset) \geq Q'/\mathtt{cost}(\mathcal{A}^*) \geq Q'/(r^{[2]}_{\mathtt{cost}} \cdot \mathrm{OPT}')$. By the second part of Lemma 4, $\forall S \subseteq \mathcal{X} \times \mathcal{Y}, u^f(x, S) \geq \frac{Q - f(S)}{r^{[2]}_{\mathtt{cost}} \cdot \mathrm{OPT}}$. Therefore, by Lemma 3, $\mathtt{cost}(\mathcal{A}) \leq \alpha(\log(Q/\eta) + 1) \cdot r^{[2]}_{\mathtt{cost}} \cdot \mathrm{OPT}$. □

Next, we show that a linear dependence of the approximation guarantee on $r^{[2]}_{\mathtt{cost}}$ is necessary for any greedy algorithm. To show the lower bound, we must exclude greedy algorithms that choose the utility function according to the set of available actions $\mathcal{X}$. Formally, define *local* greedy algorithms as follows. Assume there is a super-domain of all possible actions $\bar{\mathcal{X}}$, and consider an algorithm which receives as input a subset $\mathcal{X} \subseteq \bar{\mathcal{X}}$ of available actions. We say that such an algorithm is *local* greedy if it greedily selects the next action out of $\mathcal{X}$ using a fixed utility function $u : \bar{\mathcal{X}} \times 2^{\bar{\mathcal{X}} \times \mathcal{Y}} \to \mathbb{R}_+$, which does not depend on $\mathcal{X}$. The following lower bound shows that there exists a learning objective such that the approximation guarantee of any local greedy algorithm grows with $r^{[2]}_{\mathtt{cost}}$ or is trivially bad.

Theorem 2. *Let f be the version-space reduction objective function with the corresponding $Q = 1 - 1/|\mathcal{H}|$ and $\eta = 1/|\mathcal{H}|$. For any value of* OPT*, $r^{[2]}_{\mathtt{cost}} > 1$, and integer size of Q/η, there exist $\bar{\mathcal{X}}, \mathcal{H}$, and $\mathtt{cost}$ such that $\mathtt{cost}(x, y)$ depends only on y, and such that for any local greedy algorithm $\mathcal{A}$, there exists an input domain $\mathcal{X} \subseteq \bar{\mathcal{X}}$ such that, for η as in Theorem 1,*

$$\mathtt{cost}(\mathcal{A}) \geq \min\left(\frac{r^{[2]}_{\mathtt{cost}}}{\log_2(Q/\eta)}, \frac{Q/\eta}{\log_2(Q/\eta)}\right) \cdot \mathrm{OPT}.$$

Here $\mathtt{cost}(\mathcal{A})$ and OPT *refer to the costs for the domain $\mathcal{X}$.*

Proof. Define $\mathcal{Y} = \{1, 2, 3\}$. Let $\bar{\mathcal{X}} = \{a_i \mid i \in [k]\} \cup \{b^t_j \mid j \in [k], t \in [\lceil \log_2(k-2) \rceil]\}$. Let $\mathtt{cost}(x, y) = c_y$ for all $x \in \mathcal{X}$, where $c_3 \geq c_2 > c_1 = 0$. Set c_2, c_3 such that $r^{[2]}_{\mathtt{cost}} = c_3/c_2$. Let $\mathcal{H} = \{h_i \mid i \in [k]\}$, where h_i is defined as follows: for a_j,

$$h_i(a_j) := \begin{cases} 1 & i = j \\ 2 & i \neq j. \end{cases}$$

For b^t_j and $i \neq j$, let $l_{i,j}$ be the location of i in $(1, \ldots, j-1, j+1, \ldots, k)$, where the locations range from 0 to $k-2$. Denote by $l^t_{i,j}$ the t'th most significant bit in the binary expansion of $l_{i,j}$ to $\lceil \log_2(k-2) \rceil$ bits. Define

$$h_i(b^t_j) := \begin{cases} 1 & i \neq j \wedge l^t_{i,j} = 0 \\ 2 & i \neq j \wedge l^t_{i,j} = 1 \\ 3 & i = j \end{cases}$$

Fix an index $n \in [k]$. Let $\mathcal{X}_n = \{a_i \mid i \in [k]\} \cup \{b^t_n \mid t \in [\lceil \log_2(k-2) \rceil]\}$. We now show an interactive algorithm for $\mathcal{X}_n$ and bound its worst-case cost. On the first iteration, the algorithm selects action a_n. If the result is 1, then $\mathrm{VS}(S) = \{h_n\}$, hence $f(S) \geq Q$. In this case the cost is $c_1 = 0$. Otherwise, the algorithm selects all actions in $\{b^t_n \mid t \in [\lceil \log_2(k-2) \rceil]\}$. The responses reveal the binary expansion of $l_{j,n}$, thus limiting the version space to the a single h_i, hence $f(S) \geq Q$. In this case the total cost is at most $c_2 \lceil \log_2(k-2) \rceil$.

Now, consider a local greedy algorithm. Let $\sigma : [k] \to [k]$ be a permutation that represents the order in which $a_1, \ldots, a_k$ would be selected by the utility function if only a_i were available, and their response was always 2. Formally, $\sigma(i) = \mathrm{argmax}_{i \in [k]} u(a_{\sigma(i)}, \{(a_{\sigma(i')}, 2) \mid i' \in [i-1]\})$.[1]

[1] We may assume without loss of generality that $u(x, S) = 0$ whenever $(x, y) \in S$

Suppose the input to the algorithm is $\mathcal{X}_{\sigma(k)}$. Denote $S_i = \{(a_{\sigma(i')}, 2) \mid i' \in [i-1]\}$, and suppose $h^* = h_{\sigma(k)}$. First, assume that $\max_t u(b^t_{\sigma(k)}, S_{i'-1}) < u(a_{\sigma(k)}, S_{k-1})$. Then all of $a_{\sigma(1)}, \ldots, a_{\sigma(k-1)}$ are selected before any of $b^t_{\sigma(k)}$, and the version space is reduced to a singleton only after these $k-1$ actions. Therefore the cost of the run is at least $c_2(k-1)$. Second, assume that this assumption does not hold. Then there exists an integer i' such that $\max_t u(b^t_{\sigma(k)}, S_{i'-1}) > u(a_{\sigma(i)}, S_{i'-1})$. Let i' be the smallest such integer. Then, the algorithm receives 2 on each of the actions $a_{\sigma(1)}, \ldots, a_{\sigma(i'-1)}$, and its next action is $b^t_{\sigma(k)}$ for some t. Hence the cost of the run is at least c_3.

To summarize, the worst-case cost of every local greedy algorithm is at least $\min\{c_3, c_2(k-1)\}$ for at least one of the inputs $\mathcal{X}_n$. The worst-case cost of the optimal algorithm for each $\mathcal{X}_n$ is at most $c_2\lceil\log_2(k-2)\rceil$. The statement of the theorem follows. □

3.2 Guarantees for general objectives

In the previous section we showed that for learning objectives, the achievable approximation guarantee for greedy algorithms is characterized by $r^{[2]}_{\mathtt{cost}}$. We now turn to general consistency-aware objective functions. We show that the factor of approximation for this class depends on a different property of the cost function, which is lower bounded by $r^{[2]}_{\mathtt{cost}}$. Define $\mathtt{cost}_{\max} := \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathtt{cost}(x, y)$, and let

$$\phi_{\min} := \min_{x\in\mathcal{X}} \phi(x), \qquad gr^{[2]}_{\mathtt{cost}} := \frac{\mathtt{cost}_{\max}}{\phi_{\min}}.$$

We term the ratio $gr^{[2]}_{\mathtt{cost}}$ the *Global second smallest cost ratio*. As we show below, the approximation factor is best when $gr^{[2]}_{\mathtt{cost}}$ is equal to 1. This is the case if there is at most one preferred response for every action, and in addition, all the non-preferred responses for all actions have the same cost.

Theorem 3. *Let $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+, Q > 0, \eta > 0$ such that Assumption 1 holds and f is consistency-aware. Let $\mathcal{A}$ be an α-approximate greedy algorithm for the utility function u^f. Then* $\mathtt{cost}(\mathcal{A}) \leq 2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}}) \cdot \alpha \cdot (\log(Q/\eta) + 1) \cdot \mathrm{OPT}$.

Like Theorem 1 for learning objectives, this result for general objectives is a significant improvement over the trivial bound, mentioned in Section 2, which depends on the cost ratio, since the ratio $gr^{[2]}_{\mathtt{cost}}/r_{\mathtt{cost}}$ can be unbounded. For instance, consider a case where each action has one response with a cost of 1 and all other responses have a cost of $M \gg 1$. Then $r_{\mathtt{cost}} = M$ but $gr^{[2]}_{\mathtt{cost}} = 1$.

The proof of Theorem 3 hinges on two main observations: First, any interactive algorithm may be "reordered" without increasing its cost, so that all actions with only one possible response (given the history so far) are last. Second, there are two distinct cases for the optimal algorithm: In one case, for all $h \in \mathcal{H}$, the optimal algorithm obtains a value of at least $Q/2$ before performing actions with a single possible response. In the other case, there exists at least one mapping h for which actions with a single possible response obtain at least $Q/2$ of the value. We start with the following lemma, which handles the case where $\mathrm{OPT} < \phi_{\min}$.

Lemma 5. *Let $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$, $Q > 0$. Suppose that f is submodular, and $f(0) = \emptyset$. If $\mathrm{OPT} < \phi_{\min}$, then $\max_{x\in\mathcal{X}} u^f(x, \emptyset) \geq Q/OPT$.*

Proof. For every action $x \in \mathcal{X}$ there is at most a single y with $\mathtt{cost}(x, y) < \phi_{\min}$. Denote this response by $y(x)$. Let $\mathcal{A}$ be an optimal algorithm for f, Q. For any value of $h^* \in \mathcal{H}$, $\mathcal{A}$ only receives responses with costs less than $\phi_{\min}$. Therefore for any x that $\mathcal{A}$ selects, it receives the response $y(x)$, regardless of the identity of h^*. In other words, for all $h \in \mathcal{H}$, in every iteration t, $\mathcal{A}$ selects an action x such that $\mathcal{Y}(x, S^h_{t-1}[\mathcal{A}]) = \{y(x)\}$. It follows that for all t, $S^h_t[\mathcal{A}]$ is the same for all $h \in \mathcal{H}$. Therefore, there is a fixed set of actions that $\mathcal{A}$ selects during its run, regardless of h^*. Let $\mathcal{X}' \subseteq \mathcal{X}$ be that set. Then for all $h \in \mathcal{H}, x \in X'$, $h(x) = y(x)$. For a set $A \subseteq \mathcal{X}$, denote $A^{[y(x)]} = \{(x, y(x)) \mid x \in A\}$. We have $f(\mathcal{X}'^{[y(x)]}) \geq Q$ and $\mathtt{cost}(\mathcal{X}'^{[y(x)]}) = \mathrm{OPT}$. By the submodulrity of f, and since $f(\emptyset) = 0$, we have $Q/\mathrm{OPT} \leq f(\mathcal{X}'^{[y(x)]})/\mathrm{OPT} \leq \sum_{x\in\mathcal{X}'} f((x, y(x)))/\sum_{x\in\mathcal{X}'} \mathtt{cost}(x, y(x))$. Therefore there exists some $x \in \mathcal{X}'$ with $f((x, y(x)))/\mathtt{cost}(x, y(x)) \geq Q/\mathrm{OPT}$. Moreover, for this x we have $\mathcal{Y}(x, \emptyset) = \{y(x)\}$. Therefore $u^f(x, \emptyset) = f((x, y(x)))/\mathtt{cost}(x, y(x)) \geq Q/\mathrm{OPT}$. □

We now turn to the main lemma, to address the two cases described above.

Lemma 6. *Let $f : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}_+$, $Q > 0$. Suppose that f is submodular, and $f(0) = \emptyset$. Assume that f is consistency-aware. There exists $x \in \mathcal{X}$ such that $u^f(x, \emptyset) \geq \frac{Q}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}}$.*

Proof. If $\mathrm{OPT} < \phi_{\min}$, the statement holds by Lemma 5. Suppose that $\mathrm{OPT} \geq \phi_{\min}$. Let $\mathcal{A}^*$ be an optimal algorithm for f, Q. We may assume without loss of generality, that for any $h^* \in \mathcal{H}$, if $\mathcal{A}^*$ selects an action that has only one possible response (given the current version space) at some iteration t, then all actions selected after iteration t also have only one possible response. This does not lose generality: let t be the first iteration such that the action at iteration t has one possible response, and the action at iteration $t + 1$ has two possible responses. Consider an algorithm which behaves the same as $\mathcal{A}^*$, except that at iteration t it selects the second action, and at iteration $t + 1$ it selects the first action (regardless of the response to the first action). This algorithm has the same cost as $\mathcal{A}^*$.

For $h \in \mathcal{H}$, define $\mathrm{val}(h) := f(S^h_{t_h}[\mathcal{A}^*])$, where t_h is the last iteration in which an action with more than one possible response (given the current version space) is selected, if $h^* = h$. Consider two cases: (a) $\min_{h\in\mathcal{H}} \mathrm{val}(h) \geq Q/2$ and (b) $\exists h \in \mathcal{H}, \mathrm{val}(h) < Q/2$. In case (a), there is a bifurcating algorithm that obtains $f(S) \geq Q/2$ at cost at most OPT: This is the algorithm that selects the same actions as $\mathcal{A}^*$, but terminates before selecting the first action that has a single response given the current version space. We also have $r^{[2]}_{\mathtt{cost}} \leq \min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})$. By Lemma 2, there exists some $x \in \mathcal{X}$ such that $u^f(x, \emptyset) \geq \frac{Q}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}}$.

In case (b), let $h \in \mathcal{H}$ such that $\mathrm{val}(h) < Q/2$. Denote $S_t := S^h_t[\mathcal{A}^*]$. Let $(x_t, h(x_t))$ be the action and the response received in iteration t if $h^* = h$. Then $f(S_{t_h}) < Q/2$. Let $S' = \{(x_t, h(x_t)) \mid t > t_h\}$. Then $f(S_{t_h} \cup S') \geq Q$.

Since $f(\emptyset) = 0$ and f is submodular, $f(S') = f(S') - f(\emptyset) \geq f(S_{t_h} \cup S') - f(S_{t_h}) \geq Q - \mathrm{val}(h) \geq Q/2$. In addition, $f(S') \leq \sum_{t > t_h} f(\{(x_t, h(x_t))\})$. Hence $\frac{Q}{2\mathrm{OPT}} \leq \frac{f(S')}{\mathrm{OPT}} \leq \frac{\sum_{t>t_h} f(\{(x_t, h(x_t))\})}{\sum_{t>t_h} \mathtt{cost}(x_t, y_t)}$. Therefore there is some t' such that $\frac{f(\{(x_{t'}, h(x_{t'}))\})}{\mathtt{cost}(x_{t'}, h(x_{t'}))} \geq \frac{Q}{2\mathrm{OPT}}$. Therefore,

$$\begin{aligned}
u^f(x_{t'}, \emptyset) &= \min_{y \in \mathcal{Y}(x_{t'}, \emptyset)} \min\{f(\{(x_{t'}, y)\}), Q\}/\mathtt{cost}(x_{t'}, y) \\
&\geq \min\{Q/\mathtt{cost}_{\max}, \min_{y \in \mathcal{Y}} f(\{(x_{t'}, y)\})/\mathtt{cost}(x_{t'}, y)\} \\
&\geq \min\{Q/\mathtt{cost}_{\max}, \frac{Q}{2\mathrm{OPT}}, \min_{y \in \mathcal{Y} \setminus \{h(x_{t'})\}} f(\{(x_{t'}, y)\})/\mathtt{cost}(x_{t'}, y)\}.
\end{aligned}$$

Now, $\mathtt{cost}_{\max} \leq gr^{[2]}_{\mathtt{cost}} \cdot \phi_{\min} \leq gr^{[2]}_{\mathtt{cost}} \cdot \mathrm{OPT}$, from our assumption that $\mathrm{OPT} \geq \phi_{\min}$, and $\mathtt{cost}_{\max} \leq r_{\mathtt{cost}} \mathtt{cost}(x_{t'}, h(x_{t'})) \leq r_{\mathtt{cost}} \cdot \mathrm{OPT}$. Therefore

$$u^f(x_{t'}, \emptyset) \geq \min\left\{\frac{Q}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}}, \min_{y \in \mathcal{Y} \setminus \{h(x_{t'})\}} \frac{f(\{(x_{t'}, y)\})}{\mathtt{cost}(x_{t'}, y)}\right\}.$$

We have left to show a lower bound on $\min_{y \in \mathcal{Y} \setminus \{h(x_{t'})\}} \frac{f(\{(x_{t'}, y)\})}{\mathtt{cost}(x_{t'}, y)}$. By the choice of t', $x_{t'}$ has only one possible response given the current version space, that is $|\mathcal{Y}(x_{t'}, S_{t'-1})| = 1$. Since the same holds for all $t > t_h$, we have $\mathrm{VS}(S_{t'-1}) = \mathrm{VS}(S_{t_h})$, hence also $\mathcal{Y}(x_{t'}, S_{t_h}) = \{h(x_{t'})\}$. It follows that for $y \in \mathcal{Y} \setminus \{h(x_{t'})\}$, the set $S_{t_h} \cup \{(x_{t'}, y)\}$ is not consistent with any $h \in \mathcal{H}$. Since f is consistency-aware, it follows that $f(S_{t_h} \cup \{(x_{t'}, y)\}) \geq Q$. Therefore $f(\{(x_{t'}, y)\}) = f(\{(x_{t'}, y)\}) - f(\emptyset) \geq f(S_{t_h} \cup \{(x_{t'}, y)\}) - f(S_{t_h}) \geq Q - \mathrm{val}(h) \geq Q/2$. Hence $\frac{f(\{(x_{t'}, y)\})}{\mathtt{cost}(x_{t'}, y)} \geq \frac{Q}{2c_{\max}} \geq \frac{Q}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}}$. It follows that $u^f(x_t, \emptyset) \geq \frac{Q}{2c_{\max}} \geq \frac{Q}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}}$ also in case (b). □

Using the lemmas above, the proof of Theorem 3 is straight forward.

Proof (of Theorem 3). Let f', Q', OPT' as in Lemma 4, and let $\mathcal{A}^*$ be an optimal algorithm for f', Q'. Let $\mathcal{A}^*$ be an optimal algorithm for f', Q'. From the first part of Lemma 4, the conditions of Lemma 6 hold for f', Q'. Therefore $u^{f'}(x, \emptyset) \geq \frac{Q'}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}'}$. By the second part of Lemma 4, $\forall S \subseteq \mathcal{X} \times \mathcal{Y}$, $\quad u^f(x, S) \geq \frac{Q - f(S)}{2\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})\mathrm{OPT}}$. Therefore, by Lemma 3, $\mathtt{cost}(\mathcal{A}) \leq 2\alpha \min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})(\log(Q/\eta) + 1) \cdot \mathrm{OPT}$. □

The guarantee of Theorem 3 for general objectives is weaker than the guarantee for learning objectives given in Theorem 1: The ratio $\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})/r^{[2]}_{\mathtt{cost}}$ is always at least 1, and can be unbounded. For instance, if there are two actions that have two responses each, and all action-response pairs cost 1, except for one action-response pair which costs $M \gg 1$, then $r_{\mathtt{cost}} = 1$ but $gr^{[2]}_{\mathtt{cost}} = M$. Nonetheless, the following theorem shows that for general functions, a dependence on $\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}})$ is essential in any greedy algorithm.

Theorem 4. *For any values of* $gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}} > 0$*, there exist* $\bar{\mathcal{X}}, \mathcal{Y}, \mathcal{H}, \mathtt{cost}$ *with* $|\mathcal{Y}| = 2$ *and* $r^{[2]}_{\mathtt{cost}} = 1$*, and a submodular monotone* f *which is consistency-aware, with* $Q/\eta = 1$*, such that for any local greedy algorithm* $\mathcal{A}$*, there exists an input domain* $\mathcal{X} \subseteq \bar{\mathcal{X}}$ *such that* $\mathtt{cost}(\mathcal{A}) \geq \frac{1}{2}\min(gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}}) \cdot \mathrm{OPT}$*, where* $\mathtt{cost}(\mathcal{A})$ *and* OPT *refer to the costs of an algorithm running on the domain* $\mathcal{X}$*.*

Proof. Define $\mathcal{Y} := \{0, 1\}$. Let $g, r > 0$ be the desired values for $gr^{[2]}_{\mathtt{cost}}, r_{\mathtt{cost}}$. Let $c_1 > 0$, $c_2 := c_1 \min(g, r)$. If $g < r$, define $c_3 := c_1/r$, $c_4 := c_1$. Otherwise, set $c_4 := c_3 := c_2/g$. Define $k := \lceil c_2/c_1 \rceil + 1$. Let $\bar{\mathcal{X}} = \{a_i \mid i \in [k]\} \cup \{b_i \mid i \in [k]\} \cup \{c\}$. Let $\bar{\mathcal{H}} = \{h_i \mid i \in [k]\}$ where h_i is defined as follows: $\forall i, j \in [k], h_i(a_j) = h_i(b_j)$, and equal to 1 if and only if $i = j$, and zero otherwise, and $\forall i \in [k], h_i(c) = i \bmod 2$. Let the cost function be as follows, where $c_2 \geq c_1 > 0$, and $c_3, c_4 > 0$: $\mathtt{cost}(a_i, y) = c_1$, $\mathtt{cost}(b_i, y) = c_{y+1}$, and $\mathtt{cost}(c, y) = c_{y+3}$. Then $gr^{[2]}_{\mathtt{cost}} = g$, $r_{\mathtt{cost}} = r$ as desired.

Define f such that $\forall S \subseteq \mathcal{X} \times \mathcal{Y}$, $f(S) = Q$ if there exists in S at least one of $(a_i, 1)$ for some $i \in [k]$ or (b_i, y) for some $i \in [k], y \in \mathcal{Y}$. Otherwise, $f(S) = 0$. Note that (f, Q) is consistency-aware. Fix an index $n \in [k]$. Let $\mathcal{X}_n = \{a_i \mid i \in [k]\} \cup \{b_n\}$. We have $\mathrm{OPT} = 2c_1$: An interactive algorithm can first select a_n, and then, only if the response is $y = 0$, select b_n. Now, consider a local greedy algorithm with a utility function u. Let $\sigma : [k] \to [k]$ be a permutation that represents the order in which $a_1, \ldots, a_k$ would be selected by the utility function if only a_i were considered, and their response was always $y = 0$. Formally, $\sigma(i) = \mathrm{argmax}_{i \in [k]}\, u(a_{\sigma(i)}, \{(a_{\sigma(i')}, 0) \mid i' \in [i-1]\})$.[2]

Now, suppose the input to the algorithm is $\mathcal{X}_{\sigma(k)}$. Denote $S_i = \{(a_{\sigma(i')}, 0) \mid i' \in [i-1]\}$, and suppose that there exists an integer i' such that $u(b_{\sigma(k)}, S_{i'-1}) > u(a_{\sigma(i)}, S_{i'-1})$, and let i' be the smallest such integer. Then, if the algorithm receives 0 on each of the actions $a_{\sigma(1)}, \ldots, a_{\sigma(i'-1)}$, its next action will be $b_{\sigma(k)}$. In this case, if $h^* = h_{\sigma(k)}$, then $b_{\sigma(k)}$ is queried before $a_{\sigma(k)}$ is queried and the response $y = 1$ is received. Thus the algorithm pays at least c_2 in the worst-case. On the other hand, if such an integer i' does not exist, then if $h^* = h_{\sigma(k)}$, the algorithm selects actions $a_{\sigma(1)}, \ldots, a_{\sigma(k-1)}$ before terminating. In this case the algorithm receives $k - 1$ responses 0, thus its cost is at least $c_1(k-1)$. To summarize, every local greedy algorithm pays at least $\min\{c_2, c_1(k-1)\}$ for at least one of the inputs $\mathcal{X}_n$, while $\mathrm{OPT} = 2c_1$. By the definition of k, $\min\{c_2, c_1(k-1)\} \geq c_2$. Hence the cost of the local greedy algorithm is at least $\frac{c_2}{2c_1}\mathrm{OPT}$. □

To summarize, for both learning objectives and general objectives, we have shown that the factors $r^{[2]}_{\mathtt{cost}}$ and $gr^{[2]}_{\mathtt{cost}}$, respectively, characterize the approximation factors obtainable by a greedy algorithm.

Test parameters			Results: $\mathtt{cost}(\mathcal{A})$		
Test	$r^{[2]}_{\mathtt{cost}}$	$gr^{[2]}_{\mathtt{cost}}$	u^f	u^f_2	u^f_3
f=edge users, 3 communities	5	5	**52**	255	157
	100	100	**148**	5100	2722
	1	100	**49**	52	2821
f=edge users, 10 communities	5	5	**231**	256	242
	100	100	**4601**	5101	4802
	1	100	**50**	52	2915
f=v. reduction, 3 communities	5	5	**13**	20	15
	100	100	**203**	400	300
	1	100	**3**	4	201
f=v. reduction, 10 communities	5	5	**8**	20	15
	100	100	**105**	400	300
	1	100	**101**	103	201

Test parameters			Results: $\mathtt{cost}(\mathcal{A})$		
Test	$r^{[2]}_{\mathtt{cost}}$	$gr^{[2]}_{\mathtt{cost}}$	u^f	u^f_2	u^f_3
f=edge users, 3 communities	5	5	**51**	181	123
	100	100	**147**	3503	1833
	1	100	**51**	53	2526
f=edge users, 10 communities	5	5	246	260	**245**
	100	100	4901	5200	**4900**
	1	100	**49**	52	3217
f=v. reduction, 3 communities	5	5	**10**	20	15
	100	100	**106**	400	300
	1	100	**3**	400	300
f=v. reduction, 10 communities	5	5	**15**	16	**15**
	100	100	**300**	301	**300**
	1	100	**3**	201	300

Table 1. Results of experiments. Left: Facebook dataset, Right: GR-QC dataset

4 Experiments

We performed experiments to compare the worst-case costs of a greedy algorithm that uses the proposed u^f, to a greedy algorithm that ignores response-dependent costs, and uses instead variant of u^f, notated u^f_2, that assumes that responses for the same action have the same cost, which was set to be the maximal response cost for this action. We also compared to u^f_3, a utility function which gives the same approximation guarantees as given in Theorem 3 for u^f. Formally, $u^f_2(x,S) := \min_{h\in \mathrm{VS}(S)} \frac{\delta_{\min(f,Q)}((x,h(x))|S)}{\max_{y\in\mathcal{Y}} \mathtt{cost}(x,y)}$ and $u^f_3(x,S) := \min_{h\in \mathrm{VS}(S)} \frac{\delta_{\min(f,Q)}((x,h(x))|S)}{\min\{\mathtt{cost}(x,h(x)),\phi_{\min}\}}$. We tested these algorithms on a social network marketing objective, where users in a social network are partitioned into communities. Actions are users, and a response identifies the community the user belongs to. We tested two objective functions: "edge users" counts how many of the actions are users who have friends not from their community, assuming that these users can be valuable promoters across communities. The target value Q was set to 50.The second objective function was the version-reduction objective function, and the goal was to identify the true partition into communities out of the set of possible partitions, which was generated by considering several possible sets of "center users", which were selected randomly. We compared the worst-case costs of the algorithms under several configurations of number of communities and the values of $r^{[2]}_{\mathtt{cost}}$, $gr^{[2]}_{\mathtt{cost}}$. The cost ratio $r_{\mathtt{cost}}$ was infinity in all experiments, obtained by always setting a single response to have a cost of zero for each action. Social network graphs were taken from a friend graph from Facebook[3] [10], and a collaboration graph from Arxiv GR-QC community[4] [9]. The results are reported in Table 1. We had $|\mathcal{H}| = 100$ for all tests with 3 communities, and $|\mathcal{H}| = 500$ for all tests with 10 communities. The results show an overall preference to the proposed u^f.

[2] We may assume without loss of generality that $u(x,S) = 0$ whenever $(x,y) \in S$

[3] `http://snap.stanford.edu/data/egonets-Facebook.html`

[4] `http://snap.stanford.edu/data/ca-GrQc.html`

Acknowledgements This work was supported in part by the Israel Science Foundation (grant No. 555/15).

References

1. Cuong, N., Lee, W., Ye, N.: Near-optimal adaptive pool-based active learning with general loss. In: 30th conference on Uncertainty in Artificial Intelligence (2014)
2. Dasgupta, S.: Analysis of a greedy active learning strategy. NIPS 17, 337–344 (2004)
3. Feige, U.: A threshold of ln(n) for approximating set cover. Journal of the ACM (JACM) 45(4), 634–652 (1998)
4. Golovin, D., Krause, A.: Adaptive submodularity: Theory and applications in active learning and stochastic optimization. Journal of Artificial Intelligence Research 42, 427–486 (2011)
5. Golovin, D., Krause, A., Ray, D.: Near-optimal bayesian active learning with noisy observations. In: NIPS. pp. 766–774 (2010)
6. Gonen, A., Sabato, S., Shalev-Shwartz, S.: Efficient active learning of halfspaces: an aggressive approach. Journal of Machine Learning Research 14, 2487–2519 (2013)
7. Guillory, A., Bilmes, J.A.: Interactive submodular set cover. In: Proceedings of the 27th International Conference on Machine Learning (ICML). pp. 415–422 (2010)
8. Kapoor, A., Horvitz, E., Basu, S.: Selective supervision: Guiding supervised learning with decision-theoretic active learning. In: Proceedings of IJCAI (2007)
9. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (TKDD) 1(1), 2 (2007)
10. Leskovec, J., Mcauley, J.J.: Learning to discover social circles in ego networks. In: Advances in neural information processing systems. pp. 539–547 (2012)
11. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 510–520. HLT '11 (2011)
12. McCallum, A.K., Nigam, K.: Employing em and pool-based active learning for text classification. In: ICML (1998)
13. Mirzasoleiman, B., Karbasi, A., Sarkar, R., Krause, A.: Distributed submodular maximization: Identifying representative elements in massive data. In: NIPS. pp. 2049–2057 (2013)
14. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functionsi. Mathematical Programming 14(1), 265–294 (1978)
15. Sabato, S., Sarwate, A.D., Srebro, N.: Auditing: Active learning with outcome-dependent query costs. In: Advances in Neural Information Processing Systems 26 (NIPS) (2013)
16. Sabato, S., Sarwate, A.D., Srebro, N.: Auditing: Active learning with outcome-dependent query costs. arxiv preprint ArXiv:1306.2347v4 (2015)
17. Saettler, A., Laber, E., Cicalese, F.: Trading off worst and expected cost in decision tree problems and a value dependent model. arXiv preprint arXiv:1406.3655 (2014)
18. Saettler, A., Laber, E., Cicalese, F.: Approximating decision trees with value dependent testing costs. Information Processing Letters 115(68), 594 – 599 (2015)
19. Wolsey, L.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica 2(4), 385–393 (1982)