Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Submodular learning and covering with response-dependent costs

Sivan Sabato

Ben-Gurion University of the Negev, Beer Sheva 8499000, Israel

ARTICLE INFO

Article history:
Received 29 September 2016
Received in revised form 14 June 2017
Accepted 26 June 2017
Available online xxxx

Keywords:
Interactive learning
Submodular functions
Outcome costs

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.

1. Introduction

We consider interactive learning and covering problems, a term introduced in [1]. 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, 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. For instance, in pool-based active learning problems [2,3], each possible action is a query of the label of an example, and the goal is to identify the correct mapping from examples to labels out of a given set of possible mappings. Another example is maximizing the influence of marketing in a social network [1]. In this problem, an action is a promotion sent to specific user, and the goal is to make sure all users of a certain community are affected by the promotion, either directly or via their friends. There are many other applications for interactive algorithms. As additional examples, consider interactive sensor placement [4] and document summarization [5] with interactive user feedback.

Interactive learning and covering problems cannot be solved efficiently in general [6,7]. 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., [1,4,8]). Many problems naturally lend themselves to a submodular formulation. For instance, a pure covering objective is usually submodular, and so is an objective in which diversity is a priority, such as finding representative items in a massive data set [9]. Active learning can also be formalized as a submodular interactive covering objective, leading to efficient algorithms [3,4,1,10].

E-mail address: sabatos@cs.bgu.ac.il.

https://doi.org/10.1016/j.tcs.2017.12.033

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 [1] 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 [11–14]. In [15] 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, our analysis is not limited to settings with two possible responses. As we show below, a straightforward generalization of previous guarantees for two responses to more than two responses results in loose bounds. We thus develop new proof techniques that allow deriving tighter bounds.

The paper is structured as follows. Definitions and preliminaries are given in Section 2. We show a natural generalization of the greedy algorithm to response-dependent costs in Section 3. We provide tight approximation bounds for the greedy algorithm, and matching lower bounds, in Section 4. Experiments are reported in Section 5. We conclude in Section 6.

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

$$\forall A \subseteq B \subseteq \mathcal{Z}, \quad f(A) \le f(B).$$

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

$$\forall z \in \mathcal{Z}, A \subseteq B \subseteq \mathcal{Z}, \quad \delta_f(z \mid A) \ge \delta_f(z \mid B).$$

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 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. $\mathcal{A}(S) \in \mathcal{X} \cup \{\bot\}$ denotes the action that $\mathcal{A}$ selects after observing the set of pairs S, where $\mathcal{A}(S) = \bot$ 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, if 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. The goal of the interactive algorithm is to collect pairs S such that $f(S) \ge Q$, while minimizing $\mathfrak{cost}(\mathcal{A})$.

Please cite this article in press as: S. Sabato, Submodular learning and covering with response-dependent costs, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2017.12.033