Probably Approximately Correct Greedy Maximization (Extended Abstract)

Yash Satsangi
University of Amsterdam
y.satsangi@uva.nl

Shimon Whiteson
University of Oxford
shimon.whiteson@cs.ox.ac.uk

Frans A. Oliehoek
University of Liverpool
University of Amsterdam
frans.oliehoek@liverpool.ac.uk

ABSTRACT

Submodular function maximization finds application in a variety of real-world decision-making problems. However, most existing methods, based on greedy maximization, assume it is computationally feasible to evaluate F, the function being maximized. Unfortunately, in many realistic settings F is too expensive to evaluate exactly even once. We present *probably approximately correct greedy maximization*, which requires access only to cheap anytime confidence bounds on F and uses them to prune elements. We show that, with high probability, our method returns an approximately optimal set. We propose novel, cheap confidence bounds for *conditional entropy*, which appears in many common choices of F and for which it is difficult to find unbiased or bounded estimates. Finally, results on a real-world dataset from a multi-camera tracking system in a shopping mall demonstrate that our approach performs comparably to existing methods, but at a fraction of the computational cost.

Keywords

Submodularity; Greedy maximization; Sensor scheduling; Planning.

1. INTRODUCTION

Submodularity is a property of set functions that formalizes the notion of *diminishing returns*. Many real-world problems involve maximizing submodular functions, e.g., summarizing text, image collection summarization, or selecting sensors to minimize uncertainty about a hidden variable. Formally, given a ground set $\mathcal{X} = \{1, 2 \dots n\}$, a set function $F : 2^{\mathcal{X}} \to \mathbb{R}$, is submodular if for every $\mathcal{A}_M \subseteq \mathcal{A}_N \subseteq \mathcal{X}$ and $i \in \mathcal{X} \setminus \mathcal{A}_N$, $\Delta_F(i|\mathcal{A}_M) \geq \Delta_F(i|\mathcal{A}_N)$, where $\Delta_F(i|\mathcal{A}) = F(\mathcal{A} \cup i) - F(\mathcal{A})$ is the *marginal gain* of adding i to A. Typically, the aim is to find an $\mathcal{A}^*$ that maximizes F subject to certain constraints. Here, we consider a constraint on $\mathcal{A}^*$'s size: $\mathcal{A}^* = \arg\max_{\mathcal{A} \in \mathcal{A}^+} F(\mathcal{A})$, where $\mathcal{A}^+ = \{\mathcal{A} \subseteq \mathcal{X} : |\mathcal{A}| \leq k\}$.

As n increases, the $\binom{n}{k}$ possibilities for $\mathcal{A}^*$ grow rapidly, rendering naive maximization intractable. Instead, *greedy maximization* finds an approximate solution $\mathcal{A}^G$ faster by iteratively adding to a partial solution the element that maximizes the marginal gain. Nemhauser et. al. [4] showed that the value obtained by greedy maximization is close to that of full maximization, i.e., $F(\mathcal{A}^G) \geq (1 - e^{-1})F(\mathcal{A}^*)$, if F is submodular, non-negative and monotone.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

Most of the existing methods [3] assume it is computationally feasible to exactly compute F, and thus the marginal gain. In many settings, this is not the case. For example, consider a surveillance task [6] in which an agent aims to minimise uncertainty about a hidden state by selecting a subset of sensors that maximise *information gain*. Computing information gain is computationally expensive, especially when the hidden variable can take many values, as it involves an expectation over the entropy of posterior beliefs about the hidden variable. When surveilling large areas like shopping malls, exactly computing the entropy of a single posterior belief becomes infeasible, let alone an expectation over them.

In this paper, we present a new algorithm called *probably approximately correct greedy maximization*. Rather than assuming access to F itself, we assume access only to confidence bounds on F that are cheaper to compute than F and are *anytime*, i.e., we can tighten them by spending more computation time, e.g., by generating additional samples. Our method uses confidence bounds to prune elements, thereby avoiding the need to further tighten their bounds. We provide a PAC analysis that shows that, with high probability, our method returns an approximately optimal set.

Given an unbiased estimator of F, it is possible to use concentration inequalities like Hoeffding's inequality to obtain the confidence bounds needed by PAC greedy maximization. Unfortunately, many applications, such as sensor placement and decision tree induction require information-theoretic definitions of F such as information gain that depend on computing entropy over posterior beliefs, which are impossible to estimate in unbiased way [5]. The absence of an unbiased estimator makes it hard to obtain computationally *cheap* confidence bounds on conditional entropy [2]. Therefore, in this paper, we propose novel, cheap confidence bounds on conditional entropy.

Finally, we apply PAC greedy maximization with these new confidence bounds to a real-life dataset collected a multi-camera tracking system employed in a shopping mall. Our empirical results demonstrate that our approach performs comparably to greedy and *lazier* greedy maximization, but at a fraction of the computational cost, leading to much better scalability.

2. METHOD & ANALYSIS

In this section, we propose *probably approximately correct greedy maximization*. We assume U and L be set functions such that for $\mathcal{A} \in \mathcal{A}^+$, with probability $1 - \delta_u$, $U(\mathcal{A}) \geq F(\mathcal{A})$ and for each $\mathcal{A} \in \mathcal{A}^+$, with probability $1 - \delta_l$, $F(\mathcal{A}) \geq L(\mathcal{A})$. The main idea is to use U and L to prune elements that with high probability do not maximize marginal gain.

PAC greedy maximization works by initializing a partial solution $\mathcal{A}^P$ as $\emptyset$ and adding at each iteration the element i^P, which maximizes the marginal gain for that iteration to $\mathcal{A}^P$ until $|\mathcal{A}^P|$

equals k. Algorithm 1 shows the main subroutine for finding i^P. Algorithm 1 maintains a queue of unpruned elements prioritized by their upper bound. In each iteration of the outer while loop, the algorithm examines each of these elements and prunes it if its upper bound is not at least ϵ_1 greater than the max lower bound found so far. In addition, the element with the max lower bound is never pruned. If an element is not pruned, then its bounds are tightened.

Algorithm 1 `pac-max`$(U, L, \mathcal{A}^P, \epsilon_1, t)$

```
i^P ← 0 % element with max lower bound
for i ∈ X \ A^P do
    ρ.enqueue(i, U(A^P ∪ i)) % initialize priority queue
    i^P ← arg max_{j∈{i,i^P}} L(A^P ∪ j)}
end for
while ρ.length() > 1 ∨ change in U & L is > t do
    ρ' ← empty queue
    while ¬ρ.empty() do
        i ← ρ.dequeue()
        if i = i^P ∨ U(A^P ∪ i) >= L(A^P ∪ i^P) + ε_1 then
            tighten(A^P ∪ i)
            i^P ← arg max_{j∈{i,i^P}} L(A^P ∪ j)
            ρ'.enqueue(i, U(A^P ∪ i))
        end if
    end while
    ρ ← ρ'
end while
return i^P.
```

Assumption 1. `pac-max` always terminates with $\rho.length() = 1$.

In other words, we assume that U and L can be tightened enough to disambiguate i^P. Using this assumption we show that PAC greedy maximization with high probability achieves the same guarantees as regular greedy maximization minus a constant term.

Theorem 1. *If F is non-negative, monotone and submodular in $\mathcal{X}$ and if Assumption 1 holds then, with probability $1-\delta$,*

$$F(\mathcal{A}^P) \geq (1-e^{-1})F(\mathcal{A}^*) - \epsilon, \quad (1)$$

where $\mathcal{A}^P$ is the solution returned by PAC greedy maximization, $\mathcal{A}^* = \arg\max_{\mathcal{A}\in\mathcal{A}^+} F(\mathcal{A})$, $\delta = k(\delta_u + \delta_l)$, and $\epsilon = k\epsilon_1$.

Proof. See [7]. □

Our main focus is on settings where F has information-theoretic definitions like *information gain, entropy* or *conditional entropy*, for which computationally cheap upper and lower bounds are not easy to find in absence of an unbiased estimator. In the sensor selection task, given a set of n sensors $\mathcal{X}$, the aim is to find $\mathcal{A}^* = \arg\max_{\mathcal{A}\in\mathcal{A}^+}[-H_b^{\mathcal{A}}(s|\mathbf{z})]$. Here s denotes a hidden variable, $b(s)$ denotes the prior probability distribution over s, $\mathbf{z} = \langle z_1, z_2 \dots z_n\rangle$ denotes the observation vector, z_i stands for the observation about s from sensor i and $H_b^{\mathcal{A}}(s|\mathbf{z}) = \mathbb{E}_{\mathbf{z}|b,\mathcal{A}}[H_{b_{\mathbf{z}}^{\mathcal{A}}}(s)]$ is the conditional entropy [5] of s given $\mathbf{z}$, which is the *expected* value for the *entropy* $H_{b_{\mathbf{z}}^{\mathcal{A}}}(s)$ [5] of the posterior probability distribution $b_{\mathbf{z}}^{\mathcal{A}}(s)$.

The upper bound on $F(\mathcal{A}) = -H_b^{\mathcal{A}}(s|\mathbf{z})$ can be obtained by simply approximating the entropy of $b_{\mathbf{z}}^{\mathcal{A}}$, $H_{b_{\mathbf{z}}^{\mathcal{A}}}(s)$ with its *maximum likelihood estimates* $H_{\hat{b}_{\mathbf{z}}^{\mathcal{A}}}(s)$, that is, $U(\mathcal{A}) = -\mathbb{E}_{\mathbf{z}|b,\mathcal{A}}[H_{\hat{b}_{\mathbf{z}}^{\mathcal{A}}}(s)] + \eta$ [7]. The lower bound, which is rather difficult to find, is obtained by using $\mathbf{r} = \langle r_1, r_2 \dots, r_n\rangle$ in place of $\mathbf{z}$, which is a crude approximation of $\mathbf{z}$, e.g., obtained by clustering z_i into d clusters deterministically and r_i denotes the cluster z_i belongs to. $L(\mathcal{A})$ is defined as $L(\mathcal{A}) = -[H_{\hat{b}}^{\mathcal{A}}(s|\mathbf{r}) + \eta + \log(1 + \frac{1}{M}(\psi_b(s) - 1))]$, where $\psi_b(s)$ is the support of $b(s)$. [7] provides the proofs and details that show, with probability $1-\delta_u$, $U(\mathcal{A}) \geq F(\mathcal{A})$ and with probability $1-\delta_l$, $L(\mathcal{A}) \leq F(\mathcal{A})$ with exact definitions of δ_u, δ_l, η and M. The following theorem ties our results together to show that for $F(\mathcal{A}) = -H_b^{\mathcal{A}}(s|\mathbf{z})$, PAC greedy maximization with above definitions of U and L, computes $\mathcal{A}^P$ such that $F(\mathcal{A}^P)$ has bounded error with respect to $F(\mathcal{A}^*)$, and achieves the same guarantees as greedy maximization minus a constant term.

Theorem 2. *For the above described definitions of F, U and L, if Assumption 1 holds and if $\mathbf{z}$ is conditionally independent given s, then with probability $1-\delta$, (1) is true.*

Proof. See [7]. □

3. EXPERIMENTS & RESULTS

We evaluated PAC greedy maximization on the problem of tracking multiple people using a multi-camera system. The problem was extracted from a real-world dataset [1] collected in a shopping mall using 13 CCTV cameras for over 4 hours. To evaluate a given algorithm, a trajectory was sampled randomly. The goal is to select a subset of cameras at each time step that induce a low-entropy belief about the person's location that can be used to predict those locations. As a baseline, we tested against greedy maximization which simply uses an approximation based on the MLE estimates of posterior beliefs and lazier greedy maximization [3] which, in each iteration, samples a subset of size $\mathcal{R}$ from $\mathcal{X}$ and selects i from $\mathcal{R}$ that maximizes marginal gain.

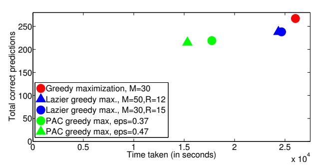

Figure 1: Multi-person tracking for $n = 20$ and $k = 3$.

Figure 1 shows the number of correct predictions (y-axis) against the runtime (x-axis) of each method. Thus, the top left is the most desirable region. PAC greedy maximization performs nearly as well as the greedy maximization and lazier greedy maximization but does so at lower computational cost leading to much better scalability.

Acknowledgments

This research is supported by the Dutch Technology Foundation STW (project #12622) and NWO Innovational Research Incentives Scheme Veni #639.021.336. We thank Henri Bouma and TNO for providing us with the dataset used in our experiments and the STW User Committee for its useful input.

REFERENCES

[1] H. Bouma, J. Baan, S. Landsmeer, C. Kruszynski, G. van Antwerpen, and J. Dijk. Real-time tracking and fast retrieval of persons in multiple surveillance cameras of a shopping mall. 2013.

[2] P.-L. Loh and S. Nowozin. Faster hoeffding racing: Bernstein races via jackknife estimates. In *Algorithmic Learning Theory*, 2013.

[3] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *AAAI*, 2015.

[4] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 1978.

[5] L. Paninski. Estimation of entropy and mutual information. *Neural computation*, 2003.

[6] Y. Satsangi, S. Whiteson, and F. Oliehoek. Exploiting submodular value functions for faster dynamic sensor selection. In *AAAI*, 2015.

[7] Y. Satsangi, S. Whiteson, and F. Oliehoek. Probably approximately correct greedy maximization. *ArXiv e-prints*, Feb. 2016. `http://arxiv.org/abs/1602.07860`.