Learning to Optimize Combinatorial Functions: Supplementary Material

Nir Rosenfeld [1] **Eric Balkanski** [1] **Amir Globerson** [2] **Yaron Singer** [1]

Theorem 1. *For all $\alpha > 1$ and $\epsilon, \delta > 0$, for m sufficiently large, there exists a family of functions $\mathcal{F}$ and a function $M_{\texttt{PMAC}}(\cdot)$ such that*

- *for all $\epsilon', \delta' > 0$: $\mathcal{F}$ is α-PMAC-learnable with sample complexity $M_{\texttt{PMAC}}(n, \delta', \epsilon', \alpha)$, and*

- *given strictly less than $M_{\texttt{PMAC}}(n, \delta, 1-(1-\epsilon)^{1/m}, \alpha)$ samples, $\mathcal{F}$ is not α-DOPS, i.e.,*

$$M_{\texttt{DOPS}}(n, m, \delta, \epsilon, \alpha) \geq M_{\texttt{PMAC}}(n, \delta, 1-(1-\epsilon)^{1/m}, \alpha).$$

Proof. Fix $\alpha > 1$ and $\epsilon > 0$. Define $p := 1-(1-\epsilon)^{1/m}+\epsilon_s$, for some small constant $\epsilon_s > 0$, and let $S_1, \dots, S_{1/p}$ be $1/p$ arbitrary distinct sets. The hard class of functions is $\mathcal{F} = \{f_i\}_{i\in[1/p]}$ where

$$f_i(S) = \begin{cases} \alpha & \text{if } S = S_i \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

Consider the distribution $\mathcal{D}$ which is the uniform distribution over sets $S_1, \dots, S_{1/p}$, so S_j is drawn with probability p for all $j \in [1/p]$. We first argue that the sample complexity for PMAC-learning f over $\mathcal{D}$ is at most

$$M_{\texttt{PMAC}}(n, \delta', \epsilon', \alpha) = \begin{cases} 0 & \text{if } \epsilon' \geq p \\ \frac{\log(1/\delta')}{\log(1/(1-p))} & \text{if } \epsilon' < p \end{cases}$$

Note that if $\epsilon' \geq p$, $\tilde{f}(S) = 1/2$ for all S is correct with probability $1-p \geq 1-\epsilon'$ over $S \sim \mathcal{D}$ and with probability 1 over the samples. If $\epsilon' < p$, if there exists sample S_i such that $f(S_i) = \alpha$, then $\tilde{f}(S_i) = \alpha$, and $\tilde{f}(S) = 1/2$ for all other S. Note that that this is correct with probability 1 over $S \sim \mathcal{D}$ if S_i is in the samples. The probability that S_i is in the samples

$$\begin{aligned} 1-(1-p)^m &= 1-(1-p)^{\frac{\log(1/\delta')}{\log(1/(1-p))}} \\ &= 1-e^{\frac{\log(\delta')}{\log(1-p)}\log(1-p)} \\ &= 1-\delta'. \end{aligned}$$

Thus, $\tilde{f}$ is correct with probability $1-\delta'$ over the samples. Next, we argue that for all $\delta > 0$ and m sufficiently large, the sample complexity for DOPS is at least

$$\begin{aligned} &M_{\texttt{PMAC}}(n, \delta, 1-(1-\epsilon)^{1/m}, \alpha) = \\ &M_{\texttt{PMAC}}(n, \delta, p-\epsilon_s, \alpha) = \frac{\log(1/\delta)}{\log(1/(1-p))}. \end{aligned}$$

Consider the random function f_i where $i \in [1/p]$ is uniformly random. Let $\mathcal{F}'$ be the randomized collection of functions f_i such that S_i is in the testing set but not in the training set. Since S_i is not in the testing set, we have that for all $f_i \in \mathcal{F}'$ and for all sets S in the testing set,

$$f_i(S) = 1.$$

Thus, the functions in $\mathcal{F}'$ are *indistinguishable* from the samples in the training set. This implies that the decisions of the algorithm are *independent* of the random variable i, *conditioned* on $f_i \in \mathcal{F}'$. Let S be the set in the testing set that is returned by the algorithm, we obtain that

$$\begin{aligned} \mathop{\mathbb{E}}_{i:f_i\in\mathcal{F}'}[f_i(S)] &= \Pr_{i:f_i\in\mathcal{F}'}[S = S_i]\cdot\alpha + \Pr_{i:f_i\in\mathcal{F}'}[S \neq S_i]\cdot\frac{1}{2} \\ &\leq \frac{\alpha}{|\mathcal{F}'|} + \frac{1}{2} \end{aligned}$$

since S is independent of i conditioned on $f_i \in \mathcal{F}'$. Consider the case where S_i is not in the training set with probability strictly greater than δ. The probability that S_i is in the testing set is $1-(1-p)^m = \epsilon + \epsilon_s$. Thus a function is in $\mathcal{F}'$ with probability at least $\delta(\epsilon + \epsilon_s)$. Note that $1/p$ is arbitrarily large if m is arbitrarily large. Thus, $|\mathcal{F}'| > 2\alpha$ with arbitrarily large probability if m is arbitrarily large for fixed ϵ, δ, and α. Combining with the previous inequality,

[1]Harvard University [2]Tel Aviv University. Correspondence to: Nir Rosenfeld <nir.rosenfeld@g.harvard.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

this implies that

$$\begin{aligned}\mathbb{E}_{i:f_i\in\mathcal{F}'}[f_i(S)] < 1 &= \frac{1}{\alpha}\cdot f_i(S_i)\\ &= \frac{1}{\alpha}\cdot \mathbb{E}_{i:f_i\in\mathcal{F}'}[\max_{S\in\mathcal{S}_{te}} f_i(S)]\end{aligned}$$

where the last equality is since $S_i \in \mathcal{S}_{te}$ for all $i \in \mathcal{F}'$. Thus, there exists at least one function $f_i \in \mathcal{F}$ such that the algorithm does not obtain an α-approximation when S_i is in the testing set and not in the training set.

The probability that S_i is in the testing set is $1-(1-p)^m = \epsilon + \epsilon_s$. Thus, S_i needs to be in the training set with probability at least $1-\delta$, otherwise we don't get an α-apx with probability $1-\epsilon$. The probability that S_i is not in the training set is $(1-p)^m$. Thus, we need $\delta > (1-p)^m$, or

$$\begin{aligned}m &> \frac{\log(1/\delta)}{\log(1/(1-p))}\\ &= m_{\texttt{PMAC}}(n,\delta,p-\epsilon_s,\alpha)\\ &= m_{\texttt{PMAC}}(n,\delta,1-(1-\epsilon)^{1/m},\alpha).\end{aligned}$$

□