Quantum Learning Linear Functions

arXiv:1902.08753
Matthias C. Caro
Technical University of Munich

Abstract

Learning Boolean linear functions from quantum examples w.r.t. the uniform distribution is known to be "easy," both with and without noise [2, 3, 4]. Based on a recent extension of the quantum Fourier transform to product distributions [6], we develop efficient quantum algorithms for learning Boolean linear functions from quantum examples w.r.t. a biased product distribution. We also prove lower bounds on the number of quantum examples required by any quantum learner for this problem to discuss how close to optimal our strategies are.

Ingredients for quantum learning linear functions

Learning à la Bernstein-Vazirani

Bernstein-Vazirani algorithm [2]:
Given $|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, \underbrace{a\cdot x}_{f_a(x)}\rangle$, for $a \in \{0,1\}^n$, apply the $(n+1)$-qubit quantum Fourier transform $H^{\otimes(n+1)}$. This produces the state

$$H^{\otimes(n+1)}|\psi_a\rangle = \frac{1}{\sqrt{2}}|0^n, 0\rangle + \frac{1}{\sqrt{2}} \sum_{j\in\{0,1\}^n} \hat{f}_a(j)|j, 1\rangle = \frac{1}{\sqrt{2}}|0^n, 0\rangle + \frac{1}{\sqrt{2}}|a, 1\rangle.$$

Measure the last qubit in the computational basis. If 0 is oberved, abort. If 1 is observed, measure the first n qubits in the computational basis and output the observed string.
$\rightarrow$ With success probability $\frac{1}{2}$, observe a, because all other Fourier coefficients of f_a vanish!

What does this have to do with learning?
Change of perspective: $a \in \{0,1\}^n$ unknown, given only quantum training data $|\psi_a\rangle^{\otimes m}$
$\rightarrow$ Bernstein-Vazirani learns a exactly with success probability $1 - \frac{1}{2^m}$.
$\rightarrow$ Success probability $\geq 1-\delta$ with $m = \mathcal{O}\left(\log\frac{1}{\delta}\right)$ copies, independent of n!

Biased Quantum Fourier Transform

So far: Quantum Fourier transform applied to a uniform superposition for Fourier sampling
What if the training data arises from a different distribution?

We use a more general notion of quantum Fourier transform for product distributions [6]:
Definition 1: For $\mu \in (-1,1)^n$, the n-qubit μ-biased quantum Fourier transform $H_\mu^{\otimes n}$ is

$$H_\mu^{\otimes n}|x\rangle := \sum_{j\in\{0,1\}^n} \sqrt{D_\mu(x)}\phi_{\mu,j}(x)|j\rangle, \quad x \in \{-1,1\}^n,$$

where $D_\mu(x)$ is a product distribution with bias μ and $\{\phi_{\mu,j}\}_{j\in\{0,1\}^n}$ is an orthonormal basis for $\mathbb{R}^{\{-1,1\}^n}$ w.r.t. the inner product $\langle f, g\rangle_\mu := \mathbb{E}_{D_\mu}[fg]$.

Now we can do the analogue of quantum Fourier sampling, but for biased Fourier coefficients:
Lemma 2 [6]: Denote $g : \{-1,1\}^n \rightarrow \{-1,1\}$, $g(x) = (-1)^{f(x)}$. There exists a quantum algorithm that, upon input of one copy of $|\psi_f\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f(x)\rangle$, outputs $j \in \{0,1\}^n$ with probability $\frac{(\hat{g}_\mu(j))^2}{2} := \frac{\mathbb{E}_{D_\mu}[g\phi_{\mu,j}]^2}{2}$, with $\hat{g}_\mu(j)$ the μ-biased Fourier coefficient.

How easy is quantum learning linear functions?

So we can estimate Fourier coefficients! Known classical methods achieving the same goal need more than just examples, namely *query* access to the unknown function.

Biased Quantum Fourier Sampling Linear Functions

We need the biased Fourier coefficients of Boolean linear functions:
Theorem 3: The function $g^{(a)} = (-1)^{f^{(a)}}$ with $f^{(a)}(x) := \sum_{i=1}^n a_i \frac{1-x_i}{2} \pmod 2$ has μ-biased Fourier coefficients $\hat{g}_\mu^{(a_1\ldots a_n)}(j_1\ldots j_n) = \prod_{l:a_l=0}(1-j_l)\cdot \prod_{l:a_l=1}\left((1-j_l)\mu_l + j_l\sqrt{1-\mu_l^2}\right)$.

We can now understand the result of biased quantum Fourier sampling of linear functions:

Corollary 4: Biased quantum Fourier sampling with $|\psi_a\rangle = \sum_{x\in\{-1,1\}^n} \sqrt{D_\mu(x)}|x, f^{(a)}(x)\rangle$ yields an output $|j_1\ldots j_{n+1}\rangle$ with:
(i) $\mathbb{P}[j_{n+1}=0] = \frac{1}{2} = \mathbb{P}[j_{n+1}=1]$, and
(ii) $\mathbb{P}[j_1\ldots j_n = \omega | j_{n+1}=1] = \prod_{l:a_l=0}(1-\omega_l)\cdot\prod_{l:a_l=1}\left((1-\omega_l)\mu_l^2 + \omega_l(1-\mu_l^2)\right)$.
We can now use component-wise amplification to increase the probability of finding a.

Quantum Algorithms for Learning Linear Functions

Learning the unknown string a completely from a fully biased distribution is clearly not possible. We assume the bias μ to be bounded and call D_μ c-bounded if $\mu \in [-1+c, 1-c]^n$.

Without further assumptions on the bias, we can learn the function with a number of quantum examples logarithmic in the number of qubits:
Theorem 5: Exact learning of Boolean linear n-bit functions w.r.t. a c-bounded D_μ can be done with $\mathcal{O}\left(\left(2\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1}\left(\ln(n) + \ln(\frac{2}{\delta})\right)\right)$ quantum examples.

If the bias is "small," the dependence on the number of qubits can be removed:
Theorem 6: Exact learning of Boolean linear n-bit functions w.r.t. a c-bounded distribution with $c > 1 - \frac{1}{\sqrt{2n}}$ can be done with $\mathcal{O}\left(\frac{1}{(1-2n(1-c)^2)^2}\ln\left(\frac{1}{\delta}\right)\right) = \mathcal{O}\left(\ln\left(\frac{1}{\delta}\right)\right)$ quantum examples.

How hard is quantum learning linear functions?

This is stable w.r.t. (certain types of) small noise in the quantum training data and in the quantum computation. This also helps to learn if μ is not known in advance.

Hardness of Learning Linear Functions

So far: How many copies of quantum training data are **sufficient** for learning?
Now: How many copies of quantum training data are **necessary** for learning?

Classical: At most 1 bit per example $\rightarrow \Omega(n)$ training examples to learn n bits
$\rightarrow$ By the above: Quantum learning does better! But by how much?

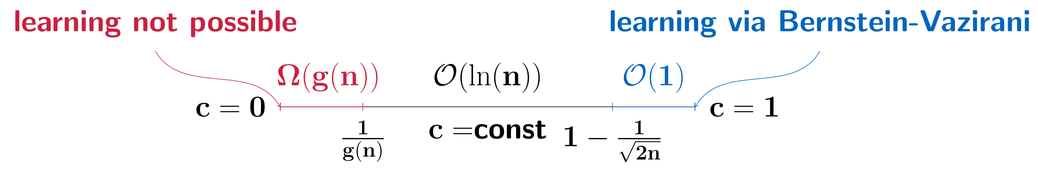
Quantum: Quantum exact learning $\approx$ State identification for training data ensemble
$\rightarrow$ Bound on success probability via Pretty Good Measurement [5, 1]
$\rightarrow$ Bound on PGM success probability using the square root of the Gram matrix of the ensemble, which can be computed inductively:
$$P^{PGM}(\mathcal{E}) \leq \left(\frac{1}{2}\left(\sqrt{1+\mu} + \sqrt{1-\mu^m}\right)\right)^{2n},$$
with m the number of quantum examples.

This gives a quantum sample complexity lower bound for distributions with "large" bias:
Theorem 7: Exact learning of Boolean linear n-bit functions w.r.t. a distribution with bias $\mu_i \geq 1 - \frac{1}{g(n)}\ \forall i$ requires $\Omega(g(n))$ quantum examples.

Using Theorem 5 for bias $\mu \geq 1 - \frac{1}{\ln(n)}$, the comparison between our lower and upper bounds on the sample size m becomes $\Omega\left(\ln(n)\right) \leq m \leq \mathcal{O}\left(\ln^2(n)\right)$, so $m = \Theta(\mathrm{polylog}(n))$. Thus we have an almost tight characterization for "large" bias.

How far from optimal are our results?

We can summarize our results on the quantum sample complexity of learning linear functions depending on the bias as follows:



There's still a discrepancy between our "small" and "large" bias assumptions. But our analysis already provides some insight into the prospects and limitations of quantum learning beyond the uniform distribution. Can we close the gap? Do we observe a similar behaviour for other function classes? Does learning become easier in a "smoothed-analysis" version of the problem?

References

[1] H. Barnum and E. Knill. *Reversing quantum dynamics with near-optimal quantum and classical fidelity*. 2000.
[2] E. Bernstein and U. Vazirani. *Quantum complexity theory*. 1993.
[3] A. W. Cross, G. Smith, and J. A. Smolin. *Quantum learning robust against noise*. 2015.
[4] A. B. Grilo, I. Kerenidis, and T. Zijlstra. *Learning with Errors is easy with quantum samples*. 2017.
[5] P. Hausladen and W. K. Wootters. *A 'Pretty Good' Measurement for Distinguishing Quantum States*. 1994.
[6] V. Kanade, A. Rocchetto, and S. Severini. *Learning DNFs under product distributions via μ-biased quantum Fourier sampling*. 2018.