A Survey of Quantum Learning Theory

Asim and Bhavishya

April 13, 2019

Table of contents

Introduction

- In recent years Machine Learning(ML) and Quantum computation(QC) have emerged as really successful technologies.
- A Lot of ML is based on *Heuristics*(i.e. not very mathematically rigorous) but this talk will focus on the theoretical side called "Quantum Learning Theory"(QML) and describe main results involving QC in three main learning models.

Introduction

- In recent years Machine Learning(ML) and Quantum computation(QC) have emerged as really successful technologies.
- A Lot of ML is based on *Heuristics*(i.e. not very mathematically rigorous) but this talk will focus on the theoretical side called "Quantum Learning Theory"(QML) and describe main results involving QC in three main learning models.
- The learner will be quantum in QML, the data may be quantum

	Classical learner	Quantum learner
Classical data	Classical ML	QML
Quantum data	?	QML

Introduction

- In recent years Machine Learning(ML) and Quantum computation(QC) have emerged as really successful technologies.
- A Lot of ML is based on *Heuristics*(i.e. not very mathematically rigorous) but this talk will focus on the theoretical side called “Quantum Learning Theory”(QML) and describe main results involving QC in three main learning models.
- The learner will be quantum in QML, the data may be quantum

	Classical learner	Quantum learner
Classical data	Classical ML	QML
Quantum data	?	QML

How can QC help ML?

- **Main Idea:** Inputs to ML problems are mostly high-dimensional vectors(imagine pixels for images, frequencies for audio etc.) i.e. $v \in \mathbb{R}^d$. We can represent them using log(d) qubits,
  $\Rightarrow |v\rangle = \frac{\sum_{j=1}^{d} v_i |j\rangle}{||v||}$.
- Then we can apply Quantum algorithms to learn from this efficient representation.
- **Caveat:** Quantum Machine Learning algorithms provide an exponential speedup only under certain assumptions like existence of quantum RAM, robustly invertible matrices etc. which make them difficult to implement in practice.

Tools from QC

- Grover Search: Provides sqrt speedup while searching unstructured databases
- Fourier Sampling: Exponentially faster than classical Fourier Sampling.

Learning Models

- **Concept:** A function $c : \{0{,}1\}^n \rightarrow \{0{,}1\}$.
- **Goal:** Assume $\mathrm{x} \in \{0{,}1\}^n$ as a n "feature" vector, then our goal is to learn c from small number of examples: (x, c(x))

	grey	brown	teeth	huge	$c(x)$
	1	0	1	0	1
	0	1	1	1	0
	0	1	1	0	1
	0	0	1	0	0

Output hypothesis could be: $(x_1 \text{ OR } x_2) \text{ AND } \neg x_4$

Figure: Taken from Ronald de Wolf's talk

Formal definitions

- **Concept:** some function $c : \{0,1\}^n \rightarrow \{0,1\}$
- **Concept class** $\mathcal{C}$: set of all concepts(eg: DNFs with small number of terms)
- Different types of learning
  1. Exact learning: $\forall c \in \mathcal{C}$, given access to MQ(c) oracle: w.p. $\geq 2/3$, output h s.t. $h(x) = c(x) \; \forall \; x$
  2. PAC learning: $\forall c \in \mathcal{C}$ and distribution D,given access to PEX(c, D) orcale: w.p $\geq 1 - \delta$, outputs h s.t. $\Pr_{x \sim D} \left[h(x) \neq c(x)\right] \leq \varepsilon$
  3. Agnostic learning: $\forall$ distributions D on $\{0,1\}^{n+1}$, given access to AEX(D) oracle: w.p. $\geq 1 - \delta$, outputs $h \in \mathcal{C}$ s.t. $\mathrm{err}_D(h) \leq opt_D(\mathcal{C}) + \varepsilon$

Framework for measuring complexity of Learning

- **Sample Complexity:** Number of examples used
- **Time Complexity:** Number of time-steps used

A good learner has small time and sample complexity, Next we compare Quantum and Classical algorithms on the basis of their sample and time complexity.

VC dimension determines Sample Complexity

- It characterizes the power of hypothesis class. Eg: Linear classifiers ¡ Polynomial classifiers.
- Measured by ability to *shatter* n points. Shatter means to classify n points in all possible labels.
- The max of n is called the VC dimension of the hypothesis class.
- Hanneke'16 showed that for every concept class $\mathcal{C}$ there exists an (ε, δ)-PAC-learner using $O(\frac{VC}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon})$ examples

Using Quantum Data

- We can put the classical data in a quantum superposition $\sum_{x \in \{0,1\}^n} \sqrt{D(x)} \, |x, c(x)\rangle$
- Under uniform D it is $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, c(x)\rangle$
- Hadamard transformation changes this into $\sum_{s \in \{0,1\}^n} \hat{c}(s) \, |s\rangle$ where $\hat{c}(s) = \frac{1}{\sqrt{2^n}} \sum_x c(x)(-1)^{s.x}$ are the Fourier coefficients of c. This allows us to sample s from distribution $\hat{c}(s)^2$.
- Eg: If c is linear mod 2(c(x) = s.x for some s) then distribution peaks at s. Thus we can learn c from one example.(Think Simon's algorithm)
- But in general PAC learning where D can be any distribution quantum examples are not significantly better than classical examples [Arunachalam & dW'17]

Proof sketch of lower bound

-

Similar results for Agnostic Learning

- Examples from unknown distribution D on (x, ℓ). Predict ℓ from x, this allows us to model the case when target concept might not even exist.
- Best concept from $\mathcal{C}$ has error $\texttt{opt} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D} [c(x) \neq \ell]$
- Therefor find $h \in \mathcal{C}$ with error $\leq \texttt{opt} + \varepsilon$
- Classical Sample complexity: $T = \Theta(\frac{VC}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2})$
- By methods similar to PAC lower bound it can be shown that Qunatum Sample complexity is same as the classical case.

Quantum improvements in time complexity

- Kearns & Vazirani'94 gave a concept class that is not efficiently PAC-learnable if factoring is hard
- But factoring is easy using Shor's algorithm. Therefore these classes can be learned efficiently[Servedio & Gortler'04]
- Servedio & Gortler'04: If classical one-way functions exist, then $\exists \mathcal{C}$ that is efficiently exactly learnable from membership queries by quantum but not by classical computers. Proof Idea: Use pseudo-random function to generate instances of Simon's problem (special 2-to-1 functions). Simon's algorithm can solve this efficiently, but classical learner would have to distinguish random from pseudo-random

Quantum improvements in time complexity

- We can quadratic speedup for some ML problems and exponential speedup under *very* strong assumptions.
- No improvements in Sample Complexity.
- Time complexity is exponentially improved for some concept classes like factoring.
- Various pragmatic issues like how to put big classical data in superposition, issues with quantum memory.