Optimal Quantum Sample Complexity of Learning Algorithms

Srinivasan Arunachalam

(Joint work with Ronald de Wolf)

Centrum Wiskunde & Informatica

QuSoft
Research Center for Quantum Software

Machine learning

Classical machine learning

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves
- Practical goal: learn "something" from given data

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves
- Practical goal: learn "something" from given data
- Recent success: deep learning is extremely good at image recognition, natural language processing, even the game of Go

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves
- Practical goal: learn "something" from given data
- Recent success: deep learning is extremely good at image recognition, natural language processing, even the game of Go
- Why the recent interest? Flood of available data, increasing computational power, growing progress in algorithms

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves
- Practical goal: learn "something" from given data
- Recent success: deep learning is extremely good at image recognition, natural language processing, even the game of Go
- Why the recent interest? Flood of available data, increasing computational power, growing progress in algorithms

Quantum machine learning

- What can quantum computing do for machine learning?

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves
- Practical goal: learn "something" from given data
- Recent success: deep learning is extremely good at image recognition, natural language processing, even the game of Go
- Why the recent interest? Flood of available data, increasing computational power, growing progress in algorithms

Quantum machine learning

- What can quantum computing do for machine learning?
- The learner will be quantum, the data may be quantum

Machine learning

Classical machine learning

- Grand goal: enable AI systems to improve themselves
- Practical goal: learn "something" from given data
- Recent success: deep learning is extremely good at image recognition, natural language processing, even the game of Go
- Why the recent interest? Flood of available data, increasing computational power, growing progress in algorithms

Quantum machine learning

- What can quantum computing do for machine learning?
- The learner will be quantum, the data may be quantum
- Some examples are known of reduction in time complexity:
  - clustering (Aïmeur et al. '06)
  - principal component analysis (Lloyd et al. '13)
  - perceptron learning (Wiebe et al. '16)
  - recommendation systems (Kerenidis & Prakash '16)

Probably Approximately Correct (PAC) learning

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target concept c: some function $c \in \mathcal{C}$ (Unknown)

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target concept c: some function $c \in \mathcal{C}$ (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$ (Unknown)

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target concept c: some function $c \in \mathcal{C}$ (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$ (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$

$\mathcal{C}$

$\downarrow$

c
target concept

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$

$\mathcal{C}$

$\downarrow$

c
target concept $x_1 \sim D$ $\longrightarrow$ $(x_1, c(x_1))$ 

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$

$\mathcal{C}$

$\downarrow$

c
target concept

$x_1 \sim D \quad \longrightarrow \quad (x_1, c(x_1))$

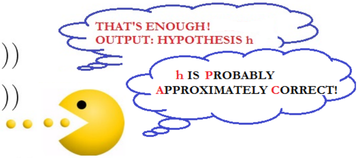
$x_2 \sim D \quad \longrightarrow \quad (x_2, c(x_2))$

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$

$$\mathcal{C} \downarrow c \text{ (target concept)} \qquad \begin{array}{ccc} x_1 \sim D & \longrightarrow & (x_1, c(x_1)) \\ x_2 \sim D & \longrightarrow & (x_2, c(x_2)) \\ \vdots & & \vdots \\ x_T \sim D & \longrightarrow & (x_T, c(x_T)) \end{array}$$



Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \rightarrow [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$.

Formally: A theory of the learnable (Valiant'84)

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$.

Formally: A theory of the learnable (Valiant'84)

- Using i.i.d. labeled examples, learner for $\mathcal{C}$ should output hypothesis h that is *Probably Approximately Correct*

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$.

Formally: A theory of the learnable (Valiant'84)

- Using i.i.d. labeled examples, learner for $\mathcal{C}$ should output hypothesis h that is *Probably Approximately Correct*
- Error of h w.r.t. target c: $err_D(c,h) = \Pr_{x \sim D}[c(x) \neq h(x)]$

Probably Approximately Correct (PAC) learning

Basic definitions

- Concept class $\mathcal{C}$: collection of Boolean functions on n bits (Known)
- Target Concept c: some function $c \in \mathcal{C}$. (Unknown)
- Distribution $D : \{0,1\}^n \to [0,1]$. (Unknown)
- Labeled example for $c \in \mathcal{C}$: $(x, c(x))$ where $x \sim D$.

Formally: A theory of the learnable (Valiant'84)

- Using i.i.d. labeled examples, learner for $\mathcal{C}$ should output hypothesis h that is *Probably Approximately Correct*
- Error of h w.r.t. target c: $err_D(c,h) = \Pr_{x \sim D}[c(x) \neq h(x)]$
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \;\; \forall D : \quad \Pr[\; \underbrace{err_D(c,h) \leq \varepsilon}_{\text{Approximately Correct}} \;] \geq \underbrace{1 - \delta}_{\text{Probably}}$$

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \quad \forall D : \quad \Pr[\ \underbrace{err_D(c,h) \leq \varepsilon}_{\text{Approximately Correct}}\ ] \geq \underbrace{1 - \delta}_{\text{Probably}}$$

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \ \ \forall D : \quad \Pr[\ \underbrace{err_D(c,h) \le \varepsilon}_{\text{Approximately Correct}}\ ] \ge \underbrace{1-\delta}_{\text{Probably}}$$

- How to measure the efficiency of the learning algorithm?

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \rightarrow \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \quad \forall D : \quad \Pr[\ \underbrace{err_D(c,h) \leq \varepsilon}_{\text{Approximately Correct}}\ ] \geq \underbrace{1-\delta}_{\text{Probably}}$$

- How to measure the efficiency of the learning algorithm?
  - Sample complexity: number of labeled examples used by learner

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \;\; \forall D : \quad \Pr[\underbrace{err_D(c,h) \leq \varepsilon}_{\text{Approximately Correct}}] \geq \underbrace{1-\delta}_{\text{Probably}}$$

- How to measure the efficiency of the learning algorithm?
  - Sample complexity: number of labeled examples used by learner
  - Time complexity: number of time-steps used by learner

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \;\; \forall D : \quad \Pr[\; \underbrace{err_D(c,h) \leq \varepsilon}_{\text{Approximately Correct}} \;] \geq \underbrace{1-\delta}_{\text{Probably}}$$

- How to measure the efficiency of the learning algorithm?
  - Sample complexity: number of labeled examples used by learner
  - Time complexity: number of time-steps used by learner
- This talk: focus on *sample complexity*

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \;\; \forall D: \quad \Pr[\; \underbrace{err_D(c,h) \le \varepsilon}_{\text{Approximately Correct}} \;] \ge \underbrace{1-\delta}_{\text{Probably}}$$

- How to measure the efficiency of the learning algorithm?
  - Sample complexity: number of labeled examples used by learner
  - Time complexity: number of time-steps used by learner
- This talk: focus on *sample complexity*
  - No need for complexity-theoretic assumptions

Complexity of learning

Recap

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \;\; \forall D : \quad \Pr[\underbrace{err_D(c,h) \leq \varepsilon}_{\text{Approximately Correct}}] \geq \underbrace{1-\delta}_{\text{Probably}}$$

- How to measure the efficiency of the learning algorithm?
  - Sample complexity: number of labeled examples used by learner
  - Time complexity: number of time-steps used by learner
- This talk: focus on *sample complexity*
  - No need for complexity-theoretic assumptions
  - No need to worry about the format of hypothesis h

Vapnik and Chervonenkis (VC) dimension

VC dimension of $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$

Vapnik and Chervonenkis (VC) dimension

VC dimension of $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$

Let M be the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of concept $c : \{0,1\}^n \to \{0,1\}$

Vapnik and Chervonenkis (VC) dimension

VC dimension of $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$

Let M be the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of concept $c : \{0,1\}^n \to \{0,1\}$

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$

Vapnik and Chervonenkis (VC) dimension

VC dimension of $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$

Let M be the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of concept $c : \{0,1\}^n \to \{0,1\}$

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$
These d column indices are shattered by $\mathcal{C}$

Vapnik and Chervonenkis (VC) dimension

VC dimension of $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$

M is the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of c

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$
These d column indices are shattered by $\mathcal{C}$

Table : VC-dim($\mathcal{C}$) $= 2$

Concepts	Truth table			
c_1	0	1	0	1
c_2	0	1	1	0
c_3	1	0	0	1
c_4	1	0	1	0
c_5	1	1	0	1
c_6	0	1	1	1
c_7	0	0	1	1
c_8	0	1	0	0
c_9	1	1	1	1

Vapnik and Chervonenkis (VC) dimension

VC dimension of $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$

M is the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of c

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$
These d column indices are shattered by $\mathcal{C}$

Table : VC-dim($\mathcal{C}$) $= 2$

Concepts	Truth table			
c_1	0	1	0	1
c_2	0	1	1	0
c_3	1	0	0	1
c_4	1	0	1	0
c_5	1	1	0	1
c_6	0	1	1	1
c_7	0	0	1	1
c_8	0	1	0	0
c_9	1	1	1	1

Table : VC-dim($\mathcal{C}$) $= 3$

Concepts	Truth table			
c_1	0	1	1	0
c_2	1	0	0	1
c_3	0	0	0	0
c_4	1	1	0	1
c_5	1	0	1	0
c_6	0	1	1	1
c_7	0	0	1	1
c_8	0	1	0	1
c_9	0	1	0	0

VC dimension characterizes PAC sample complexity

VC dimension of $\mathcal{C}$

M is the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of c

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$

These d column indices are shattered by $\mathcal{C}$

Fundamental theorem of PAC learning

VC dimension characterizes PAC sample complexity

VC dimension of $\mathcal{C}$

M is the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of c

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$
These d column indices are shattered by $\mathcal{C}$

Fundamental theorem of PAC learning

Suppose VC-dim($\mathcal{C}$) $= d$

VC dimension characterizes PAC sample complexity

VC dimension of $\mathcal{C}$

M is the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of c

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$
These d column indices are shattered by $\mathcal{C}$

Fundamental theorem of PAC learning

Suppose VC-dim$(\mathcal{C}) = d$

- Blumer-Ehrenfeucht-Haussler-Warmuth'86:
  every (ε, δ)-PAC learner for $\mathcal{C}$ needs $\Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ examples

VC dimension characterizes PAC sample complexity

VC dimension of $\mathcal{C}$

M is the $|\mathcal{C}| \times 2^n$ Boolean matrix whose c-th row is the truth table of c

VC-dim($\mathcal{C}$): largest d s.t. the $|\mathcal{C}| \times d$ rectangle in M contains $\{0,1\}^d$
These d column indices are shattered by $\mathcal{C}$

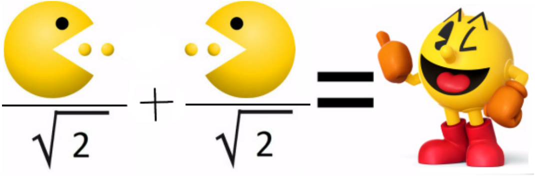
Fundamental theorem of PAC learning

Suppose VC-dim$(\mathcal{C}) = d$

- Blumer-Ehrenfeucht-Haussler-Warmuth'86:
  every (ε, δ)-PAC learner for $\mathcal{C}$ needs $\Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ examples
- Hanneke'16: there exists an (ε, δ)-PAC learner for $\mathcal{C}$ using $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ examples

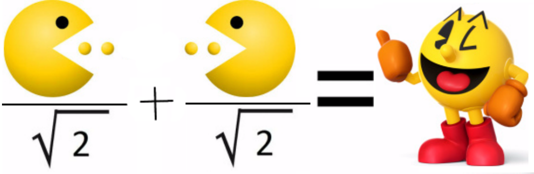
Quantum PAC learning

Do quantum computers provide an advantage for PAC learning?



Quantum PAC learning

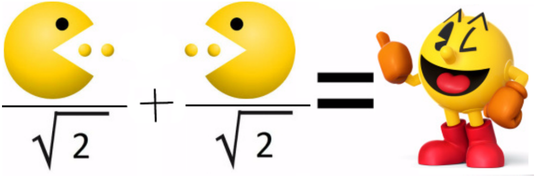
Do quantum computers provide an advantage for PAC learning?



Quantum data

Quantum PAC learning

Do quantum computers provide an advantage for PAC learning?



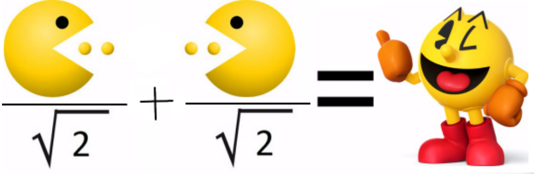
Quantum data

- Bshouty-Jackson'95: Quantum example is a superposition

$$|E_{c,D}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$$

Quantum PAC learning

Do quantum computers provide an advantage for PAC learning?



Quantum data

- Bshouty-Jackson'95: Quantum example is a superposition

$$|E_{c,D}\rangle = \sum_{x \in \{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$$

- Measuring this $(n+1)$-qubit state gives a classical example, so quantum examples are at least as powerful as classical

Quantum PAC learning

Quantum Data

- Quantum example: $|E_{c,D}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$
- Quantum examples are at least as powerful as classical examples

Quantum is indeed more powerful for learning! (for a fixed distribution)

Quantum PAC learning

Quantum Data

- Quantum example: $|E_{c,D}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$
- Quantum examples are at least as powerful as classical examples

Quantum is indeed more powerful for learning! (for a fixed distribution)

- Learning class of linear functions under uniform D:

Quantum PAC learning

Quantum Data

- Quantum example: $|E_{c,D}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$
- Quantum examples are at least as powerful as classical examples

Quantum is indeed more powerful for learning! (for a fixed distribution)

- Learning class of linear functions under uniform D:
  Classical: $\Omega(n)$ classical examples needed
  Quantum: $O(1)$ quantum examples suffice (Bernstein-Vazirani'93)

Quantum PAC learning

Quantum Data

- Quantum example: $|E_{c,D}\rangle = \sum_{x \in \{0,1\}^n} \sqrt{D(x)}\, |x, c(x)\rangle$
- Quantum examples are at least as powerful as classical examples

Quantum is indeed more powerful for learning! (for a fixed distribution)

- Learning class of linear functions under uniform D:
  Classical: $\Omega(n)$ classical examples needed
  Quantum: $O(1)$ quantum examples suffice (Bernstein-Vazirani'93)
- Learning DNF under uniform D:

Quantum PAC learning

Quantum Data

- Quantum example: $|E_{c,D}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$
- Quantum examples are at least as powerful as classical examples

Quantum is indeed more powerful for learning! (for a fixed distribution)

- Learning class of linear functions under uniform D:
  Classical: $\Omega(n)$ classical examples needed
  Quantum: $O(1)$ quantum examples suffice (Bernstein-Vazirani'93)
- Learning DNF under uniform D:
  Classical: Best known upper bound is quasi-poly. time (Verbeugt'90)
  Quantum: Polynomial-time (Bshouty-Jackson'95)

Quantum PAC learning

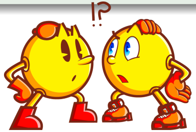
Quantum Data

- Quantum example: $|E_{c,D}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$
- Quantum examples are at least as powerful as classical examples

Quantum is indeed more powerful for learning! (for a fixed distribution)

- Learning class of linear functions under uniform D:
  Classical: $\Omega(n)$ classical examples needed
  Quantum: $O(1)$ quantum examples suffice (Bernstein-Vazirani'93)
- Learning DNF under uniform D:
  Classical: Best known upper bound is quasi-poly. time (Verbeugt'90)
  Quantum Polynomial-time (Bshouty-Jackson'95)

But in the PAC model,
learner has to succeed for all D!



Quantum sample complexity

Quantum sample complexity

Quantum upper bound

Classical upper bound $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ carries over to quantum

Quantum sample complexity

Quantum upper bound

Classical upper bound $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ carries over to quantum

Best known quantum lower bounds

Atici & Servedio'04: lower bound $\Omega\left(\frac{\sqrt{d}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\right)$

Zhang'10 improved first term to $\frac{d^{1-\eta}}{\varepsilon}$ for all $\eta > 0$

Quantum sample complexity = Classical sample complexity

Quantum upper bound

Classical upper bound $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ carries over to quantum

Best known quantum lower bounds

Atici & Servedio'04: lower bound $\Omega\left(\frac{\sqrt{d}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\right)$

Zhang'10 improved first term to $\frac{d^{1-\eta}}{\varepsilon}$ for all $\eta > 0$

Our result: Tight lower bound

We show: $\Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ quantum examples are necessary

Quantum sample complexity = Classical sample complexity

Quantum upper bound

Classical upper bound $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ carries over to quantum

Best known quantum lower bounds

Atici & Servedio'04: lower bound $\Omega\left(\frac{\sqrt{d}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\right)$

Zhang'10 improved first term to $\frac{d^{1-\eta}}{\varepsilon}$ for all $\eta > 0$

Our result: Tight lower bound

We show: $\Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ quantum examples are necessary

Two proof approaches

- Information theory: conceptually simple, nearly-tight bounds

Quantum sample complexity = Classical sample complexity

Quantum upper bound

Classical upper bound $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ carries over to quantum

Best known quantum lower bounds

Atici & Servedio'04: lower bound $\Omega\left(\frac{\sqrt{d}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\right)$

Zhang'10 improved first term to $\frac{d^{1-\eta}}{\varepsilon}$ for all $\eta > 0$

Our result: Tight lower bound

We show: $\Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ quantum examples are necessary

Two proof approaches

- Information theory: conceptually simple, nearly-tight bounds
- Optimal measurement: tight bounds, some messy calculations

Proof approach: Pretty Good Measurement

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z \in [m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated,

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated,
  but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement,

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated,
  but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm}$

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z \in [m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z \in [m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z \in [m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*
- Goal: show $T \geq d/\varepsilon$, where $d = \text{VC-dim}(\mathcal{C})$

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*
- Goal: show $T \geq d/\varepsilon$, where $d = \text{VC-dim}(\mathcal{C})$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$.

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*
- Goal: show $T \geq d/\varepsilon$, where $d = \text{VC-dim}(\mathcal{C})$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Fix a nasty distribution D: $D(s_0) = 1 - 16\varepsilon$, $D(s_i) = 16\varepsilon/d$ on $\{s_1, \ldots, s_d\}$

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*
- Goal: show $T \geq d/\varepsilon$, where $d = \text{VC-dim}(\mathcal{C})$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Fix a nasty distribution D: $D(s_0) = 1 - 16\varepsilon$, $D(s_i) = 16\varepsilon/d$ on $\{s_1, \ldots, s_d\}$
- Let $E : \{0,1\}^k \to \{0,1\}^d$ be a good error-correcting code s.t. $k \geq d/4$ and $d_H(E(y), E(z)) \geq d/8$

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*
- Goal: show $T \geq d/\varepsilon$, where $d = \text{VC-dim}(\mathcal{C})$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Fix a nasty distribution D: $D(s_0) = 1 - 16\varepsilon$, $D(s_i) = 16\varepsilon/d$ on $\{s_1, \ldots, s_d\}$
- Let $E : \{0,1\}^k \to \{0,1\}^d$ be a good error-correcting code s.t. $k \geq d/4$ and $d_H(E(y), E(z)) \geq d/8$
- Pick concepts $\{c^z\}_{z\in\{0,1\}^k} \subseteq \mathcal{C}$:

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: if P_{opt} is the success probability of the optimal measurement, then $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Quantum PAC: Given $|\psi_c\rangle = |E_{c,D}\rangle^{\otimes T}$, learn c *approximately*
- Goal: show $T \geq d/\varepsilon$, where $d = \text{VC-dim}(\mathcal{C})$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Fix a nasty distribution D: $D(s_0) = 1 - 16\varepsilon$, $D(s_i) = 16\varepsilon/d$ on $\{s_1, \ldots, s_d\}$
- Let $E : \{0,1\}^k \to \{0,1\}^d$ be a good error-correcting code s.t. $k \geq d/4$ and $d_H(E(y), E(z)) \geq d/8$
- Pick concepts $\{c^z\}_{z\in\{0,1\}^k} \subseteq \mathcal{C}$: $c^z(s_0) = 0$, $c^z(s_i) = E(z)_i$ $\forall$ i

Pick concepts $\{c^z\} \subseteq \mathcal{C}$: $c^z(s_0) = 0$, $c^z(s_i) = E(z)_i \ \forall \ i$

Suppose $VC(\mathcal{C}) = d + 1$ and $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$, i.e., $|\mathcal{C}| \times (d+1)$ rectangle of $\{s_0, \ldots, s_d\}$ contains $\{0,1\}^{d+1}$

Pick concepts $\{c^z\} \subseteq \mathcal{C}$: $c^z(s_0) = 0,\ c^z(s_i) = E(z)_i\ \forall\ i$

Suppose $VC(\mathcal{C}) = d + 1$ and $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$, i.e., $|\mathcal{C}| \times (d+1)$ rectangle of $\{s_0, \ldots, s_d\}$ contains $\{0,1\}^{d+1}$

Concepts	Truth table						
$c \in \mathcal{C}$	s_0	s_1	$\cdots$	s_{d-1}	s_d	$\cdots$	$\cdots$
c_1	0	0	$\cdots$	0	0	$\cdots$	$\cdots$
c_2	0	0	$\cdots$	1	0	$\cdots$	$\cdots$
c_3	0	0	$\cdots$	1	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$
c_{2^d-1}	0	1	$\cdots$	1	0	$\cdots$	$\cdots$
c_{2^d}	0	1	$\cdots$	1	1	$\cdots$	$\cdots$
c_{2^d+1}	1	0	$\cdots$	0	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$
$c_{2^{d+1}}$	1	1	$\cdots$	1	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$

$c(s_0) = 0$ (rows c_1 to c_{2^d})

Pick concepts $\{c^z\} \subseteq \mathcal{C}$: $c^z(s_0) = 0$, $c^z(s_i) = E(z)_i \ \forall \ i$

Suppose $VC(\mathcal{C}) = d + 1$ and $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$, i.e., $|\mathcal{C}| \times (d+1)$ rectangle of $\{s_0, \ldots, s_d\}$ contains $\{0,1\}^{d+1}$

Concepts $c \in \mathcal{C}$	Truth table						
	s_0	s_1	$\cdots$	s_{d-1}	s_d	$\cdots$	$\cdots$
c_1	0	0	$\cdots$	0	0	$\cdots$	$\cdots$
c_2	0	0	$\cdots$	1	0	$\cdots$	$\cdots$
c_3	0	0	$\cdots$	1	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$
c_{2^d-1}	0	1	$\cdots$	1	0	$\cdots$	$\cdots$
c_{2^d}	0	1	$\cdots$	1	1	$\cdots$	$\cdots$
c_{2^d+1}	1	0	$\cdots$	0	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$
$c_{2^{d+1}}$	1	1	$\cdots$	1	1	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$

$c(s_0) = 0$ (for $c_1, \ldots, c_{2^d}$)

Among $\{c_1, \ldots, c_{2^d}\}$, pick 2^k concepts that correspond to codewords of $E : \{0,1\}^k \to \{0,1\}^d$ on $\{s_1, \ldots, s_d\}$

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated,
  but we can always use the Pretty Good Measurement
- Crucial property: $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Given $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$, learn c^z *approximately*. Show $T \geq d/\varepsilon$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Fix a nasty distribution D:
  $D(s_0) = 1 - 16\varepsilon$, $D(s_i) = 16\varepsilon/d$ on $\{s_1, \ldots, s_d\}$
- Let $E : \{0,1\}^k \rightarrow \{0,1\}^d$ be a good error-correcting code
  s.t. $k \geq d/4$ and $d_H(E(y), E(z)) \geq d/8$
- Pick concepts $\{c^z\}_{z\in\{0,1\}^k} \subseteq \mathcal{C}$: $c^z(s_0) = 0$, $c^z(s_i) = E(z)_i$ $\forall$ i

Proof approach: Pretty Good Measurement

State identification: Ensemble $\mathcal{E} = \{(p_z, |\psi_z\rangle)\}_{z\in[m]}$

- Given state $|\psi_z\rangle \in \mathcal{E}$ with prob p_z Goal: identify z
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement
- Crucial property: $P_{opt} \geq P_{pgm} \geq P_{opt}^2$ (Barnum-Knill'02)

How does learning relate to identification?

- Given $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$, learn c^z *approximately*. Show $T \geq d/\varepsilon$
- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. Fix a nasty distribution D: $D(s_0) = 1 - 16\varepsilon$, $D(s_i) = 16\varepsilon/d$ on $\{s_1, \ldots, s_d\}$
- Let $E : \{0,1\}^k \to \{0,1\}^d$ be a good error-correcting code s.t. $k \geq d/4$ and $d_H(E(y), E(z)) \geq d/8$
- Pick concepts $\{c^z\}_{z\in\{0,1\}^k} \subseteq \mathcal{C}$: $c^z(s_0) = 0$, $c^z(s_i) = E(z)_i \ \forall\ i$
- *Learning* c^z approximately (wrt D) is equivalent to *identifying* z!

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$
- Goal: Show $T \geq d/\varepsilon$

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1-\delta$
- Goal: Show $T \geq d/\varepsilon$

Analysis of PGM

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$
- Goal: Show $T \geq d/\varepsilon$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have P_{pgm}

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$
- Goal: Show $T \geq d/\varepsilon$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$
- Goal: Show $T \geq d/\varepsilon$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$
- $P_{pgm} \leq$

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1-\delta$
- Goal: Show $T \geq d/\varepsilon$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$
- $P_{pgm} \leq \cdots$ 4-page calculation $\cdots \leq \exp(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)$

Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner (i.e., measurement) that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1-\delta$
- Goal: Show $T \geq d/\varepsilon$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$
- $P_{pgm} \leq \cdots$4-page calculation$\cdots \leq \exp(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)$
- This implies $T = \Omega(d/\varepsilon)$

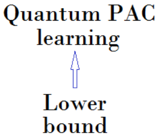
Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$
- $P_{pgm} \leq \cdots$ 4-page calculation $\cdots \leq \exp(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)$
- This implies $T = \Omega(d/\varepsilon)$



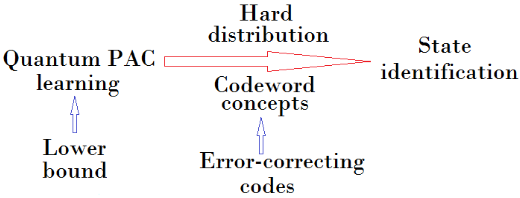
Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1 - \delta$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$
- $P_{pgm} \leq \cdots$ 4-page calculation $\cdots \leq \exp(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)$
- This implies $T = \Omega(d/\varepsilon)$



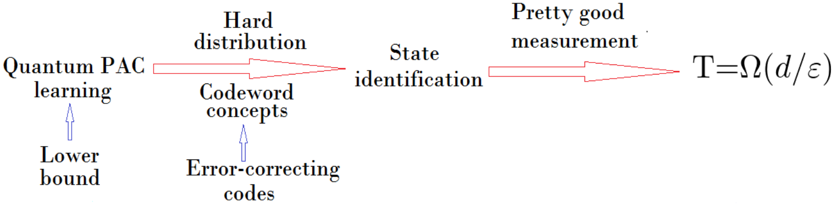
Sample complexity lower bound via PGM

Recap

- Learning c^z approximately (wrt D) is equivalent to identifying z!
- If sample complexity is T, then there is a good learner that *identifies* z from $|\psi_{c^z}\rangle = |E_{c^z,D}\rangle^{\otimes T}$ with probability $\geq 1-\delta$

Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{pgm} \geq P_{opt}^2 \geq (1-\delta)^2$
- $P_{pgm} \leq \cdots$ 4-page calculation $\cdots \leq \exp(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)$
- This implies $T = \Omega(d/\varepsilon)$



Agnostic learning

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$
- Goal of the agnostic learner: output $h \in \mathcal{C}$ with error

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$
- Goal of the agnostic learner: output $h \in \mathcal{C}$ with error $\leq \text{OPT} + \varepsilon$

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$
- Goal of the agnostic learner: output $h \in \mathcal{C}$ with error $\leq \text{OPT} + \varepsilon$

What about sample complexity?

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$
- Goal of the agnostic learner: output $h \in \mathcal{C}$ with error $\leq \text{OPT} + \varepsilon$

What about sample complexity?

- Classical sample complexity: $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$ [VC74,Tal94]

Agnostic learning

Lets get real!

- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$
- Goal of the agnostic learner: output $h \in \mathcal{C}$ with error $\leq \text{OPT} + \varepsilon$

What about sample complexity?

- Classical sample complexity: $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$ [VC74,Tal94]
- No quantum bounds known before (unlike PAC model)

Agnostic learning

Lets get real!

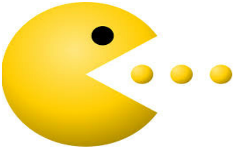
- So far, examples were generated according to a target concept $c \in \mathcal{C}$
- In realistic situations we could have *"noisy" examples* for the target concept, or maybe *no fixed target concept* even exists

How do we model this? Agnostic learning

- Unknown distribution D on (x, ℓ) generates examples
- Suppose "best" concept in $\mathcal{C}$ has error $\mathrm{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$
- Goal of the agnostic learner: output $h \in \mathcal{C}$ with error $\leq \mathrm{OPT} + \varepsilon$

What about sample complexity?

- Classical sample complexity: $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$ [VC74,Tal94]
- No quantum bounds known before (unlike PAC model)
- We show the quantum examples do not reduce sample complexity



Classical PAC

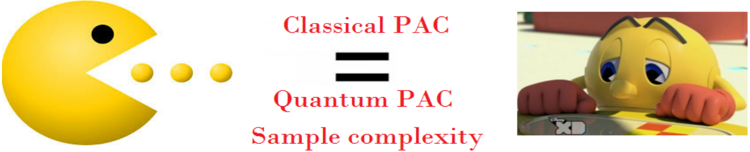
=

Quantum PAC

Sample complexity

Conclusion

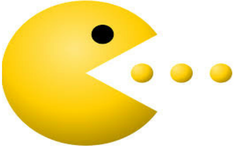
Conclusion and future work



Conclusion

- PAC and agnostic: Quantum examples are no better than classical

Conclusion and future work



Classical PAC

=

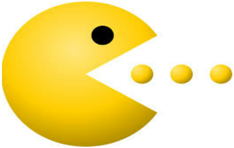
Quantum PAC

Sample complexity

Conclusion

- PAC and agnostic: Quantum examples are no better than classical
- We also studied the model with random classification noise and show that quantum examples are no better than classical

Conclusion and future work



Classical PAC

=

Quantum PAC

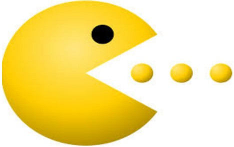
Sample complexity

Conclusion

- PAC and agnostic: Quantum examples are no better than classical
- We also studied the model with random classification noise and show that quantum examples are no better than classical

Future work

Conclusion and future work



Classical PAC

=

Quantum PAC

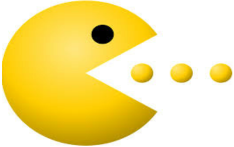
Sample complexity

Conclusion

- PAC and agnostic: Quantum examples are no better than classical
- We also studied the model with random classification noise and show that quantum examples are no better than classical

Future work

- Quantum machine learning is still young! Don't have convincing examples where quantum significantly improve machine learning



Classical PAC

=

Quantum PAC

Sample complexity

Conclusion

- PAC and agnostic: Quantum examples are no better than classical
- We also studied the model with random classification noise and show that quantum examples are no better than classical

Future work

- Quantum machine learning is still young! Don't have convincing examples where quantum significantly improve machine learning
- Theoretically, one could consider more optimistic PAC-like models where learner need not succeed $\forall c \in \mathcal{C}$ and $\forall D$

Buffer 1: Proof approach via Information theory

- Suppose $\{s_0, \ldots, s_d\}$ is shattered by $\mathcal{C}$. By definition: $\forall a \in \{0,1\}^d\ \exists c \in \mathcal{C}$ s.t. $c(s_0) = 0$, and $c(s_i) = a_i\ \forall\ i \in [d]$
- Fix a nasty distribution D:

  $D(s_0) = 1 - 4\varepsilon,\ D(s_i) = 4\varepsilon/d$ on $\{s_1, \ldots, s_d\}$.
- Good learner produces hypothesis h s.t. $h(s_i) = c(s_i) = a_i$ for $\geq \frac{3}{4}$ of is

  Think of c as uniform d-bit string A, approximated by $h \in \{0,1\}^d$ that depends on examples $B = (B_1, \ldots, B_T)$

  1. $I(A : B) \geq I(A : h(B)) \geq \Omega(d)$ [because $h \approx A$]
  2. $I(A : B) \leq \sum_{i=1}^{T} I(A : B_i) = T \cdot I(A : B_1)$ [subadditivity]
  3. $I(A : B_1) \leq 4\varepsilon$ [because prob of useful example is 4ε]

  This implies $\Omega(d) \leq I(A : B) \leq 4T\varepsilon$, hence $T = \Omega(\frac{d}{\varepsilon})$
- For analyzing quantum examples, only step 3 changes: $I(A : B_1) \leq O(\varepsilon \log(d/\varepsilon)) \Rightarrow T = \Omega(\frac{d}{\varepsilon} \frac{1}{\log(d/\varepsilon)})$

Buffer 2: Proof approach in detail

- Suppose we're given state $|\psi_i\rangle$ with prob p_i, $i = 1, \ldots, m$. Goal: learn i
- Optimal measurement could be quite complicated, but we can always use the Pretty Good Measurement.

  This has POVM operators

  $M_i = p_i \rho^{-1/2} |\psi_i\rangle\langle\psi_i| \rho^{-1/2}$, where $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$
- Success probability of PGM: $P_{PGM} = \sum_i p_i \mathrm{Tr}(M_i |\psi_i\rangle\langle\psi_i|)$
- Crucial property (BK'02): if P_{OPT} is the success probablity of the optimal POVM, then $P_{OPT} \geq P_{PGM} \geq P_{OPT}^2$
- Let G be the $m \times m$ Gram matrix of the vectors $\sqrt{p_i}\,|\psi_i\rangle$, then $P_{PGM} = \sum_i \sqrt{G}(i,i)^2$

Buffer 3: Analysis of PGM

- For the ensemble $\{|\psi_{c^z}\rangle : z \in \{0,1\}^k\}$ with uniform probabilities $p_z = 1/2^k$, we have $P_{PGM} \geq (1-\delta)^2$
- Let G be the $2^k \times 2^k$ Gram matrix of the vectors $\sqrt{p_z}\,|\psi_{c^z}\rangle$, then $P_{PGM} = \sum_z \sqrt{G}(z,z)^2$
- $G_{xy} = g(x \oplus y)$. Can diagonalize G using Hadamard transform, and its eigenvalues will be $2^k \widehat{g}(s)$. This gives $\sqrt{G}$
- $\sum_z \sqrt{G}(z,z)^2 \leq \cdots$ 4-page calculation $\cdots \leq$
$$\leq \exp(T^2\varepsilon^2/d + \sqrt{Td\varepsilon} - d - T\varepsilon)$$
- This implies $T = \Omega(d/\varepsilon)$