Quantum Learning Theory

Ronald de Wolf

CWI

QuSoft

Universiteit van Amsterdam

Machine learning

- Algorithmically finding patterns and generalizations of given data. For prediction, understanding, theorizing,...
- Recently great successes in image recognition, natural language processing, playing Go, ...

- Different settings for machine learning:
  - Supervised learning: labeled examples
  - Unsupervised learning: unlabeled examples
  - Reinforcement learning: interaction with environment

Quantum machine learning?

- No need to stick to classical learning algorithms —
  What can quantum computers do for machine learning?

- The learner will be quantum, the data may be quantum

	Classical learner	*Quantum learner*
Classical data	Classical ML	This talk
Quantum data	?	This talk

Won't cover: classical ML to help quantum

- Many-body quantum state tomography with classical neural networks (Carleo & Troyer'16, Torlai et al.'17)
- In quantum error correction: learn to predict the best correction operations from the error syndrome measurement outcomes (Torlai & Melko'16, Baireuther et al.'17)
- Learning to create new quantum experiments & to control quantum systems (Melnikov et al.'17)
- Classical heuristics beating quantum annealing (Katzgraber et al.'17)

How can quantum computing help machine learning?

- Core idea: inputs to learning problem are often high-dimensional vectors of numbers (texts, images, ...). These can be viewed as amplitudes in a quantum state.

  Required number of qubits is only logarithmic in dimension!

  Vector $v \in \mathbb{R}^d \Rightarrow \log_2(d)$-qubit state $|v\rangle = \frac{1}{\|v\|}\sum_{i=1}^{d} v_i |i\rangle$

- So we want to efficiently represent our data as quantum states, and apply quantum algorithms on them to learn.

  Easier said than done...

- This talk focuses on provable, non-heuristic parts of QML:
  1. Some cases where quantum helps for specific ML problems
  2. Some more general quantum learning theory

Part 1:

Some cases where quantum helps ML

Example 1: Principal Component Analysis

- Data: classical vectors $v_1, \ldots, v_N \in \mathbb{R}^d$. For example:
  - jth entry of v_i counts # times document i contains keyword j
  - jth entry of v_i indicates whether buyer i bought product j
- PCA: find the principal components of

  $$\text{"correlation matrix"} \quad A = \sum_{i=1}^{N} v_i v_i^T$$

  Main eigenvectors describe patterns in the data.

  Can be used to summarize data, for prediction, etc.
- Idea for quantum speed-up (Lloyd, Mohseni, Rebentrost'13):

  IF we can efficiently prepare the $|v_i\rangle$ as $\log_2(d)$-qubit states, then doing this for random i gives mixed state $\rho = \frac{1}{N}A$.

  We want to sample (eigenvector,eigenvalue)-pairs from ρ

Example 1 (cntd): PCA via self-analysis

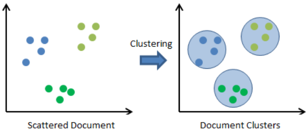
- Using few copies of ρ, we want to run $U = e^{-i\rho}$ on some σ
- Idea: start with $\sigma \otimes \rho$, apply $\mathrm{SWAP}^{\varepsilon}$, throw away 2nd register.
  1st register now has $U^{\varepsilon}\sigma(U^{\dagger})^{\varepsilon}$, up to error $O(\varepsilon^2)$.
  Repeat this $1/\varepsilon$ times, using a fresh copy of ρ each time.
  First register now contains $U\sigma U^{\dagger}$, up to error $\frac{1}{\varepsilon}O(\varepsilon^2) = O(\varepsilon)$
- Suppose ρ has eigendecomposition $\rho = \sum_i \lambda_i |w_i\rangle\langle w_i|$.
  Phase estimation maps $|w_i\rangle|0\rangle \mapsto |w_i\rangle|\tilde{\lambda}_i\rangle$,
  where $|\lambda_i - \tilde{\lambda}_i| \leq \delta$, using $O(1/\delta)$ applications of U
- Phase estimation on another fresh copy of ρ maps

$$\rho \otimes |0\rangle\langle 0| \mapsto \sum_i \lambda_i |w_i\rangle\langle w_i| \otimes |\tilde{\lambda}_i\rangle\langle\tilde{\lambda}_i|$$

  Measuring 2nd register samples $|w_i\rangle|\tilde{\lambda}_i\rangle$ with probability λ_i

Example 2: clustering based on PCA

- Data: classical vectors $v_1, \ldots, v_N \in \mathbb{R}^d$
  Goal: group these into $k \ll N$ clusters



- Good method: let the k clusters correspond to top-k eigenvectors of the correlation matrix $A = \sum_{i=1}^{N} v_i v_i^T$

- Idea for quantum speed-up (Lloyd et al.):

  IF we can efficiently prepare the $|v_i\rangle$ as $\log_2(d)$-qubit states, then we can use PCA to sample from top eigenvectors of A.

  Can build a database of several copies of each of the k top eigenvectors of A, thus learning the centers of the k clusters (as quantum states!)

Example 3: nearest-neighbor classification

- Data: classical vectors $w_1, \ldots, w_k \in \mathbb{R}^d$, representing k "typical" categories (clusters)
- Input: a new vector $v \in \mathbb{R}^d$ that we want to classify, by finding its nearest neighbor among $w_1, \ldots, w_k$
- Idea for quantum speed-up (Aïmeur et al.'07; Wiebe et al.'14):

  IF we can efficiently prepare $|v\rangle$ and $|w_i\rangle$ as $\log_2(d)$-qubit states, say in time P, then we can use the SWAP test to estimate distance $\| v - w_i \|$ up to small error, say in time P

  Then use Grover's algorithm on top of this to find $i \in \{1, \ldots, k\}$ minimizing $\| v - w_i \|$. Assign v to cluster i
- Complexity: $O(P\sqrt{k})$

How to put classical data in superposition?

- Given vector $v \in \mathbb{R}^d$: how to prepare $|v\rangle = \frac{1}{\|v\|} \sum_{i=1}^{d} v_i |i\rangle$?
- Assume quantum-addressable memory: $O_v : |i, 0\rangle \mapsto |i, v_i\rangle$

1. Find $\mu = \max_i |v_i|$ in $O(\sqrt{d})$ steps (Dürr-Høyer min-finding)
2. $\frac{1}{\sqrt{d}} \sum_i |i\rangle \overset{O_v}{\mapsto} \frac{1}{\sqrt{d}} \sum |i, v_i\rangle \mapsto \frac{1}{\sqrt{d}} \sum |i, v_i\rangle (\frac{v_i}{\mu}|0\rangle + \sqrt{1 - \frac{v_i^2}{\mu^2}}|1\rangle)$
   $\overset{O_v^{-1}}{\mapsto} \frac{1}{\sqrt{d}} \sum_i |i\rangle (\frac{v_i}{\mu}|0\rangle + \sqrt{1 - \frac{v_i^2}{\mu^2}}|1\rangle) = \frac{\|v\|}{\mu\sqrt{d}}|v\rangle|0\rangle + |w\rangle|1\rangle$
3. Boost $|0\rangle$ by $O\left(\frac{\mu\sqrt{d}}{\| v \|}\right)$ rounds of amplitude amplification

- Expensive for “peaked” v; cheap for “uniform” or “sparse” v (but there you can efficiently compute many things classically!)

Example 4: Recommendation systems

- m users, n products (movies), unknown $m \times n$ preference matrix $P = \begin{pmatrix} \ddots & & \ddots \\ & P_{ij} & \\ \ddots & & \ddots \end{pmatrix}$

  Assume $\exists$ rank-k approximation $P_k \approx P$, for some $k \ll m, n$
- Information about P comes in online: user i likes movie j. System can only access partial matrix $\widehat{P}$ with this information
- **Goal**: provide new recommendation to user i by sampling from ith row of P (normalized)

- Classical methods: construct rank-k completion $\widehat{P}_k$ from $\widehat{P}$, hope that $\widehat{P}_k \approx P$. Time poly(k, m, n)
- Kerenidis & Prakash'16: quantum recommendation system polylog(mn) update & poly$(k, \log(mn))$ recommendation time

Example 4: Quantum recommendation system (sketch)

- "Subsample matrix": $\widehat{P}_{ij} = \begin{cases} P_{ij}/p & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$
- $\widehat{P}_k$: projection of $\widehat{P}$ on its top-k singular vectors.
  Achlioptas & McSherry'01: $\widehat{P}_k \approx P$ in Frobenius distance.
  Hence for most i: ith row of $\widehat{P}_k$ is close to ith row of P
- For most i, sampling from ith row of $\widehat{P}_k$ is similar to sampling from ith row of P, so gives a good recommendation for user i
- Non-zero entries of $\widehat{P}$ come in one-by-one. Kerenidis & Prakash create data structure (polylog(mn) update time) that can generate $|i\text{th row of } \widehat{P}\rangle$ in polylog(mn) time
- When asked for a recommendation for user i:
  generate $|i\text{th row of } \widehat{P}\rangle$, project onto largest singular vectors of $\widehat{P}$ (via phase estimation), measure resulting quantum state

Many other attempts at using quantum for ML

- k-means clustering
- Support Vector Machines
- Training perceptrons (≈depth-1 neural networks)
- Quantum deep learning (=deep neural networks)
- Training Boltzmann machines for sampling
- ...

Problems:

- How to efficiently put classical data in superposition?
- How to use reasonable assumptions about the data (also in classical ML; much work is heuristic rather than rigorous)
- We don't have a large quantum computer yet...

Part 2:

Some more general quantum learning theory

Supervised learning

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$.
  Think of $x \in \{0,1\}^n$ as an object described by n "features", and concept c as describing a set of related objects
- **Goal**: learn c from a 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: $(x_1 \text{ OR } x_2) \text{ AND } \neg x_4$

Making this precise: Valiant's "theory of the learnable"

- Concept: some function $c : \{0,1\}^n \to \{0,1\}$
  Concept class $\mathcal{C}$: set of concepts (small circuits, DNFs,...)
- Example for an unknown target concept $c \in \mathcal{C}$:
  $(x, c(x))$, where $x \sim$ unknown distribution D on $\{0,1\}^n$
- Goal: using some i.i.d. examples, learner for $\mathcal{C}$ should output hypothesis h that is probably approximately correct (PAC).

  h is a function of examples and of learner's randomness.

  Error of h w.r.t. target c: $\text{err}_D(c, h) = \Pr_{x \sim D}[c(x) \neq h(x)]$
- An algorithm (ε, δ)-PAC-learns $\mathcal{C}$ if:

$$\forall c \in \mathcal{C} \;\; \forall D : \quad \Pr[\quad \underbrace{\text{err}_D(c, h) \leq \varepsilon}_{h \text{ is approximately correct}} \quad] \geq 1 - \delta$$

Complexity of learning

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

$$\forall c \in \mathcal{C} \ \ \forall D : \quad \Pr[\quad \underbrace{\mathrm{err}_D(c,h) \leq \varepsilon}_{h \text{ is approximately correct}} \quad] \geq 1 - \delta$$

- How to measure the efficiency of the learning algorithm?
  - Sample complexity: number of examples used
  - Time complexity: number of time-steps used
- A good learner has small time & sample complexity

VC-dimension determines sample complexity

- Cornerstone of classical sample complexity: VC-dimension

  Set $S = \{s_1, \ldots, s_d\} \subseteq \{0,1\}^n$ is shattered by $\mathcal{C}$ if
  for all $a \in \{0,1\}^d$, there is $c \in \mathcal{C}$ s.t. $\forall i \in [d] : c(s_i) = a_i$

  VC-dim($\mathcal{C}$) $= \max\{d : \exists S$ of size d shattered by $\mathcal{C}\}$

- Equivalently, let M be the $|\mathcal{C}| \times 2^n$ matrix whose c-row is the truth-table of c. Then M contains complete $2^d \times d$ rectangle

- 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: for every concept class $\mathcal{C}$, there exists an (ε, δ)-PAC-learner using $O\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ examples

Quantum data

- Let's try to circumvent the problem of putting classical data in superposition, by assuming we start from quantum data: one or more copies of some quantum state, generated by natural process or experiment
- Bshouty-Jackson'95: suppose example is a superposition

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

  Measuring this $(n+1)$-qubit state gives a classical example, so quantum examples are at least as powerful as classical
- Next slide: some cases where quantum examples are more powerful than classical for a fixed distribution D

Uniform quantum examples help some learning problems

- Quantum example under uniform D: $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, c(x)\rangle$
- Hadamard transform can turn this into $\sum_{s \in \{0,1\}^n} \widehat{c}(s)|s\rangle$

  $\hat{c}(s) = \frac{1}{2^n} \sum_x c(x)(-1)^{s \cdot x}$ are the Fourier coefficients of c.
  This allows us to sample s from distribution $\hat{c}(s)^2$!
- If c is linear mod 2 $(c(x) = s \cdot x$ for one $s)$, then distribution is peaked at s. We can learn c from one quantum example!
- Bshouty-Jackson'95: efficiently learn Disjunctive Normal Form (DNF) formulas. Fourier sampling + classical "boosting"
- Reduced sample complexity for juntas, sparse c's, LWE, ...
- But in the PAC model, learner has to succeed for all D

Quantum sample complexity

Could quantum sample complexity be significantly smaller than classical sample complexity in the PAC model?

- Classical sample complexity is $\Theta\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$
- Classical upper bound carries over to quantum examples
- Atici & Servedio'04: lower bound $\Omega\left(\frac{\sqrt{d}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\right)$
- Arunachalam & dW'17: tight bounds: $\Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$ quantum examples are necessary to learn $\mathcal{C}$

Hence in distribution-independent learning:
quantum examples are not significantly better than classical examples

Proof sketch of the lower bound

- Let $S = \{s_0, s_1, \ldots, s_d\}$ be shattered by $\mathcal{C}$.
  Define distribution D with $1 - 8\varepsilon$ probability on s_0,
  and $8\varepsilon/d$ probability on each of $\{s_1, \ldots, s_d\}$.
- ε-error learner takes T quantum examples and produces hypothesis h that agrees with $c(s_i)$ for $\geq \frac{7}{8}$ of $i \in \{1, \ldots, d\}$.
  This is an approximate state identification problem
- Take a good error-correcting code $E : \{0,1\}^k \to \{0,1\}^d$, with $k = d/4$, distance between any two codewords $> d/4$:
  approximating codeword $E(z) \Leftrightarrow$ exactly identifying $E(z)$
- We now have an exact state identification problem with 2^k possible states. Quantum learner cannot be much better than the "Pretty Good Measurement," and we can analyze precisely how well PGM can do as a function of T.

  High success probability $\Rightarrow T \geq \Omega\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$

Similar results for agnostic learning

- Agnostic learning: unknown distribution D on (x, ℓ) generates examples. We want to learn to predict bit ℓ from x.
  This allows to model situations where we only have "noisy" examples for the target concept; maybe no fixed target concept even exists

- Best concept from $\mathcal{C}$ has error $\mathrm{OPT} = \min_{c \in \mathcal{C}} \Pr_{(x,\ell)\sim D}[c(x) \neq \ell]$

- Goal of the learner: output $h \in \mathcal{C}$ with error $\leq \mathrm{OPT} + \varepsilon$

- Classical sample complexity: $T = \Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$

  NB: generalization error $\varepsilon = O(1/\sqrt{T})$, not $O(1/T)$ as in PAC

- Again, we show the quantum sample complexity is the same, by analyzing PGM to get optimal quantum bound

Pretty good tomography

- Suppose we have some copies available of n-qubit mixed state ρ, and some observables we could measure
- Learning ρ requires roughly 2^{2n} measurements (& copies of ρ). This "full tomography" is very expensive already for $n = 8$
- Aaronson'06 used a classical PAC-learning result to get:
  Let $\mathcal{E}$ be set of measurement operators and D distribution on $\mathcal{E}$. From $O(n)$ i.i.d. data points of the form $(E, Tr(E\rho))$, where $E \sim D$, we can learn an n-qubit state σ such that: If $E \sim D$, then with high probability, $Tr(E\rho) \approx Tr(E\sigma)$.
- This learning algorithm has bad time complexity in general, but can be efficient in special cases (e.g., stabilizer states)
- Aaronson'17 also defined shadow tomography: find a σ that's good for all $E \in \mathcal{E}$ using $n \cdot \text{polylog}(|\mathcal{E}|)$ copies of ρ

Active learning

- In some situations, instead of passively receiving examples for the target concept $c : \{0,1\}^n \to \{0,1\}$ that we want to learn, we can actively "probe it"
- Membership query: ask $c(x)$ for any $x \in \{0,1\}^n$ of our choice
- Cases where quantum membership queries help:
  - Linear functions $\mathcal{C} = \{c(x) = s \cdot x \mid s \in \{0,1\}^n\}$:
    Fourier sampling learns target with 1 membership query
  - Point functions $\mathcal{C} = \{\delta_z \mid z \in \{0,1\}^n\}$:
    Grover learns target with $O(\sqrt{2^n})$ membership queries
- Quantum improvement cannot be very big: if $\mathcal{C}$ can be learned by Q quantum membership queries, then it can also be learned by $O(n\,Q^3)$ classical queries (Servedio & Gortler'04).
  Has been improved by $\log Q$ factor (ACLW'18)

Quantum improvements in time complexity

- Kearns & Vazirani'94 gave a concept class that is not efficiently PAC-learnable *if factoring is hard*

  Angluin & Kharitonov'95: concept class that is not efficiently learnable from membership queries *if factoring is hard*
- But factoring is *easy* for a quantum computer! Servedio & Gortler'04: these classes can be learned efficiently using Shor
- 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

Summary & Outlook

- Quantum machine learning combines two great fields
- You can get quadratic speed-ups for some ML problems, exponential speed-ups are under strong assumptions.
  Biggest issue: how to put big classical data in superposition
- In some scenarios: provably no quantum improvement
- Still, this area is very young, and I expect much more
- Optimization tools for quantum machine learning algorithms:
  - Minimization / maximization (Grover's algorithm)
  - Solving large systems of linear eqns (HHL algor.)
  - Solving linear and semidefinite programs
  - Gradient-descent with faster gradient-calculation
  - Physics methods: adiabatic computing, annealing

Some open problems

- Find a good ML-problem on classical data with a quantum method circumventing classical-data-to-quantum-data issue
- Find a good ML-problem where the HHL linear-systems solver can be applied & its pre-conditions are naturally satisfied
- Efficiently learn constant-depth formulas from uniform quantum examples, generalizing Bshouty-Jackson's DNF
- Show that if concept class $\mathcal{C}$ can be learned with Q quantum membership queries, then it can also be learned with $O(Q^2 + Qn)$ classical membership queries
- Can we do some useful quantum ML on ~ 100 qubits with moderate noise?

Further reading: Many recent surveys

- Wittek, *Quantum machine learning: What quantum computing means to data mining*, Elsevier, 2014
- Schuld et al., *An introduction to quantum machine learning*, arXiv:1409.30
- Adcock et al., *Advances in quantum machine learning*, arXiv:1512.0290
- Biamonte et al., *Quantum machine learning*, arXiv:1611.093
- Arunachalam & de Wolf, *A survey of quantum learning theory*, arXiv:1701.06806
- Ciliberto et al., *Quantum machine learning: a classical perspective*, arXiv:1707.08561
- Dunjko & Briegel, *Machine learning & artificial intelligence in the quantum domain*, arXiv:1709.02779