Quantumly sampling optimised random features to speed up kernel based methods for Machine Learning

arXiv:2004.10756

Sathyawageeswar Subramanian

Department of Computer Science & DIMAP
University of Warwick, UK

Sathya.Subramanian@warwick.ac.uk

Acknowledgements

Co-authors and Funding

This talk is based on join work with Dr. Hayata Yamasaki[1], Dr. Sho Sonoda[2], and Prof. Masato Koashi[3], published in NeurIPS 2020:

Learning with Optimized Random Features: Exponential Speedup by Quantum Machine Learning without Sparsity and Low-Rank Assumptions;
H Yamasaki, S Subramanian, S Sonoda, M Koashi; *Advances in Neural Information Processing Systems, 2020*.

Funding: This work was supported by CREST (Japan Science and Technology Agency) JPMJCR1671, Cross-ministerial Strategic Innovation Promotion Program (SIP) (Council for Science, Technology and Innovation (CSTI)), JSPS Overseas Research Fellowships, a Cambridge-India Ramanujan scholarship from the Cambridge Trust and the SERB (Govt. of India), and JSPS KAKENHI 18K18113.

[1] Austrian Academy of Sciences (JST PRESTO), [2] RIKEN AIP, [3] The University of Tokyo

Overview

Supervised Learning

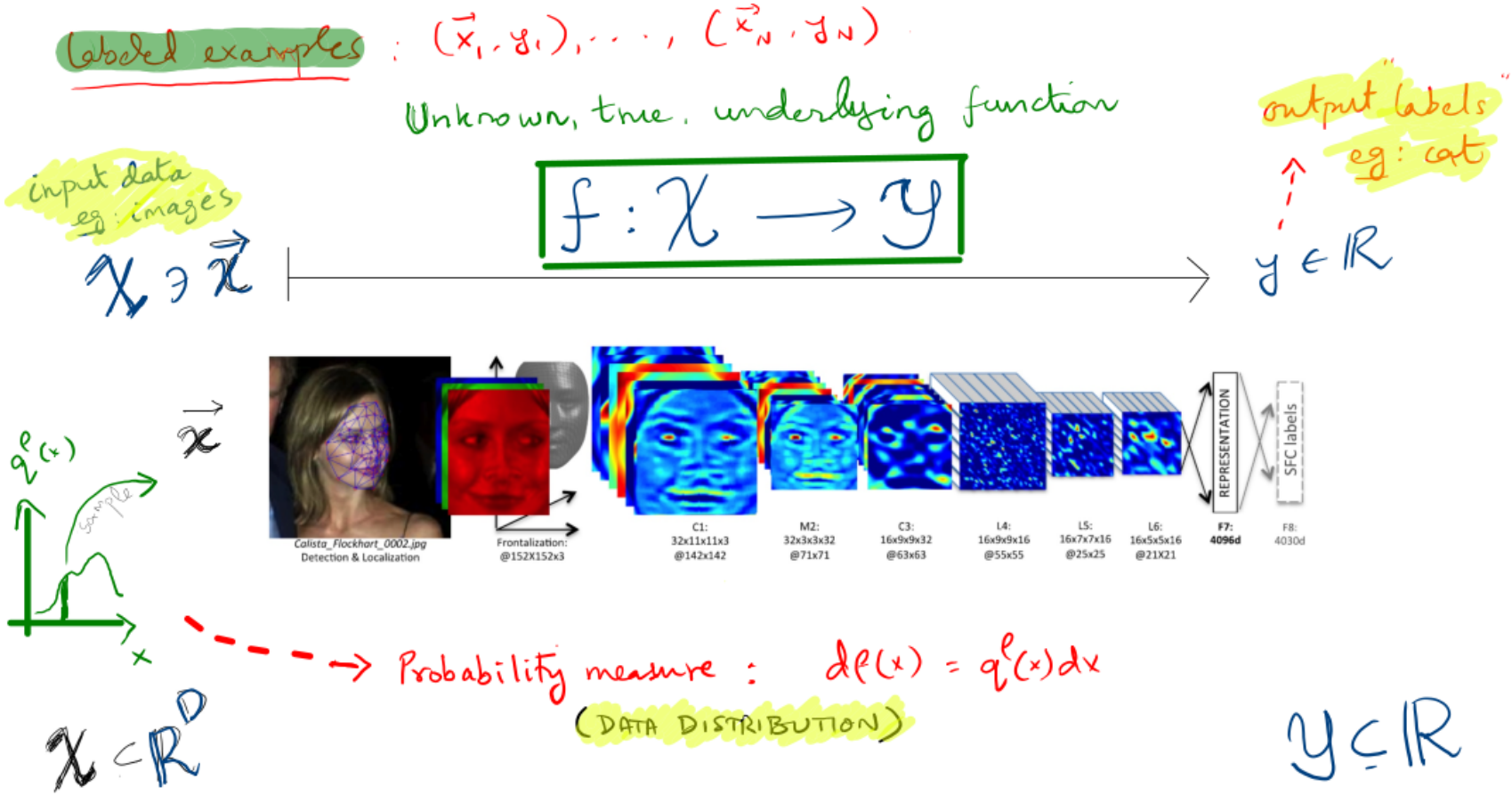

Image from Taigman, Yang, Ranzato, Wolf (2014)

Supervised Learning

Models and Kernels

- Learn (approximation) of f from a few labeled examples; N = sample complexity
- **Model**: Find best $\hat{f} \in \mathcal{F}$ a chosen space of functions
- **Generalisation error**: minimise average error over (unseen) data; $\int_{\tilde{\mathcal{X}}} d\rho(x) \left| \hat{f}(x) - f(x) \right| \leq \epsilon$ — learning to desired accuracy ϵ
- **Kernel**: "Similarity function" $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$; way to use known data to predict f on unknown data
- **Translation Invariance**: $k(x, y) = \tilde{k}(|x - y|)$

Random feature expansions

- **Idea**: express the unknown function as an integral / linear combination over "feature functions" determined using input data
- **Kernel** can be written as an average over feature functions, with some distribution over a 'feature space' $\mathcal{V} \subset \mathbb{R}^{D'}$
- **1D Feature (function)**: $\varphi : \mathcal{V} \times \mathcal{X} \to \mathbb{R}$, $\varphi(v_m, x) = \mathrm{e}^{-2\pi \mathrm{i} v_m \cdot x}$
- Ultimately, target function can be written as

$$\hat{f}_\alpha(x) := \sum_{m=0}^{M-1} \alpha_m \varphi(v_m, x),$$

with coefficients $\alpha_i \in \mathbb{R}$ to be determined by regression, minimising generalisation error

Random feature expansions

Optimised Random Features

- **Aim:** Minimise the number M of random features required and so the length of the linear combination $\hat{f}$ by using feature functions optimised for the input of labeled examples, and the choice of kernel
- **Conventional RF [Rahimi & Recht (2008)]**:
  Sample $v_0, \ldots, v_{M-1} \sim d\tau$; get features $\varphi(v_i, \cdot)$; Regression gives α_i;
  $d\tau$ data independent, # features $M = \widetilde{\mathcal{O}}(1/\epsilon^2)$
- **Optimised RF [Bach (2017)]**:
  Sample $v_0, \ldots, v_{M-1} \sim q_\epsilon^*(v) d\tau(v)$; ..regression gives α_i;
  $q_\epsilon^*(v) d\tau(v)$ data-optimised, # features $M = \widetilde{\mathcal{O}}(\log^2 1/\epsilon)$
- Provably optimal upto logarithmic factors
- After discretisation and other technicalities,

$$Q_\epsilon^*(\tilde{v}) \propto \left\langle \varphi(\tilde{v}, \cdot) \left| \hat{\mathrm{q}}^\rho (\hat{\Sigma} + \epsilon \mathbb{1})^{-1} \right| \varphi(\tilde{v}, \cdot) \right\rangle \tag{1}$$

Optimised Random Features

Our Problem setup

- **Integral operator** Σ:

$$(\Sigma f)(x') := \int_{\mathcal{X}} d\rho(x)\, k(x', x)\, f(x). \tag{2}$$

- After discretisation and other technicalities, (with standard braket notation for inner products etc.)

$$Q^*_\epsilon(\tilde{v}) \propto \left\langle \varphi(\tilde{v}, \cdot) \middle| \hat{\mathrm{q}}^\rho (\hat{\Sigma} + \epsilon \mathbb{1})^{-1} \middle| \varphi(\tilde{v}, \cdot) \right\rangle, \tag{3}$$

where $q^\rho(x)$ is the probability density function over the input space

Function / operator on $\mathcal{X}$	Vector / operator on $\mathcal{H}^X$
$f : \mathcal{X} \to \mathbb{C}$	$\lvert f\rangle \propto \sum_{\tilde{x}} f(\tilde{x}) \lvert \tilde{x}\rangle$
$\varphi(v, \cdot) : \mathcal{X} \to \mathbb{C}$	$\lvert \varphi(v, \cdot)\rangle \propto \sum_{\tilde{x}} \varphi(v, \tilde{x}) \lvert \tilde{x}\rangle$
$\tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	$\mathrm{k} := \sum_{\tilde{x}', \tilde{x}} \tilde{k}(\tilde{x}', \tilde{x}) \lvert \tilde{x}'\rangle\langle \tilde{x}\rvert$
$q^\rho : \mathcal{X} \to \mathbb{R}$	$\mathrm{q}^\rho := \sum_{\tilde{x}} q^\rho(\tilde{x}) \lvert \tilde{x}\rangle\langle \tilde{x}\rvert$
Σ acting on $f : \mathcal{X} \to \mathbb{C}$	$\Sigma := \mathrm{k}\mathrm{q}^\rho$
$\Sigma f : \mathcal{X} \to \mathbb{C}$	$\Sigma \lvert f\rangle$

Motivation

Why this method, and why Quantum?

- Scope & domain of applicability Bach's method extend beyond supervised ML (eg. ridge regression, clustering can all be 'kernelised'), to numerical computation of high-dimensional integrals in signal processing, applied mathematics, Bayesian inference, function approximation, optimisation

Motivation

Why this method, and why Quantum?

- Scope & domain of applicability Bach's method extend beyond supervised ML (eg. ridge regression, clustering can all be 'kernelised'), to numerical computation of high-dimensional integrals in signal processing, applied mathematics, Bayesian inference, function approximation, optimisation
- Sampling from the optimised distribution Q_ϵ^* over features = classical bottleneck; best known algorithm requires inversion of a full-rank, non-sparse, $\mathcal{O}(\mathrm{e}^D)$ dimensional matrix, incurring worst case runtime $\mathcal{O}(\mathrm{e}^D)$
- Q_ϵ^* involves inverting integral operator. Inversion $\implies$ HHL09? But dense, full-rank. We show how to overcome these difficulties.

Reference

1 Francis Bach (2017). On the equivalence between kernel quadrature rules & random feature expansions. Journal of Machine Learning Research, 18, 1–38.

Related works

- **Kernels & QML**: Quantum enhanced features [Havlicek et al. (Nature 2019)], Experimental Kernel-based QML [Bartkiewicz et al. (Nature Scientific Reports 2020)], Vedaie et al. (2011.09694), Schuld (2101.11020), Park et al. (2004.03489), Blank et al. (npj QI 2020) ...

Related works

- **Kernels & QML**: Quantum enhanced features [Havlicek et al. (Nature 2019)], Experimental Kernel-based QML [Bartkiewicz et al. (Nature Scientific Reports 2020)], Vedaie et al. (2011.09694), Schuld (2101.11020), Park et al. (2004.03489), Blank et al. (npj QI 2020) ...

Category	Method	Title
Quantum version of SVM	Grover algorithm	Quantum optimization for training support vector machines (Anguita et al. 2003)
Quantum version of SVM	HHL algorithm	Quantum support vector machine for big data classification (Rebentrost et al. 2014)
Experimental	NMR 4-qubit quantum processor	Experimental implementation of a quantum support vector machine (Li et al. 2015)
Experimental	IBM quantum experience	Quantum algorithm Implementations for beginners (Patrick et al. 2018)
Quantum version of SVM and ECOC	HHL algorithm	Quantum error-correcting output codes (Windridge et al. 2018)]
Kernel methods	Variational quantum circuit	Quantum machine learning in feature Hilbert spaces (Schuld and Killoran 2019)
Kernel methods	Variational quantum circuit	Supervised learning with quantum-enhanced feature spaces (Havlicek et al. 2019)
Kernel methods	Topological quantum computation	Hamming distance kernelisation via topological quantum computation (Di Pierro et al. 2017)

Table 1 from Mengoni, R., Di Pierro, A. Kernel methods in Quantum Machine Learning. Quantum Machince Intelligence **1**, 65–71 (2019).

Our contributions

- Even with powerful tools like QRAM, attaining genuine quantum speedups has been difficult
- Our work circumvents (1) sparsity and (2) low rank assumptions by exploiting a combination of the QFT and QSVT
- Our asymptotic speedup is exponential over the best known classical algorithm, in many useful parameter regimes
- Technical analysis of Fourier sparse operators and construction of block encodings could be more widely applicable

Main result

Quantumly sampling optimised random features

Theorem

Given D-dimensional data, for any accuracy $\epsilon > 0$ we can sample a (discrete) optimised random feature $\tilde{v} \in \tilde{\mathcal{V}}$ from a weighted distribution $Q^*_\epsilon(\tilde{v})P^\tau(\tilde{v})$ with

$$\sum_{\tilde{v}\in\tilde{\mathcal{V}}} |(Q(\tilde{v}) - Q^*_\epsilon(\tilde{v}))P^\tau(\tilde{v})| \leq \delta',$$

in runtime

$$T = \widetilde{\mathcal{O}}\left(D \log D\right) \times \widetilde{\mathcal{O}}\left(\frac{Q^*_{\max}}{\epsilon} \text{poly} \log \frac{1}{\delta'}\right).$$

Main result

Quantumly sampling optimised random features

Theorem

Given D-dimensional data, for any accuracy $\epsilon > 0$ we can sample a (discrete) optimised random feature $\tilde{v} \in \tilde{\mathcal{V}}$ from a weighted distribution $Q_\epsilon^*(\tilde{v})P^\tau(\tilde{v})$ with

$$\sum_{\tilde{v}\in\tilde{\mathcal{V}}} |(Q(\tilde{v}) - Q_\epsilon^*(\tilde{v}))P^\tau(\tilde{v})| \leq \delta',$$

in runtime

$$T = \widetilde{\mathcal{O}}\left(D \log D\right) \times \widetilde{\mathcal{O}}\left(\frac{Q_{\max}^*}{\epsilon} \text{poly} \log \frac{1}{\delta'}\right).$$

In particular, T is linear in D, while the best known classical algorithm for estimating $Q_\epsilon^*(\tilde{v})P^\tau(\tilde{v})$ requires $\mathcal{O}(\mathrm{e}^D)$ time.

Proof techniques

- First obtain consistent and asymptotically exact discretisation scheme, $\mathcal{X} \mapsto \tilde{\mathcal{X}}$, $\mathcal{V} \mapsto \tilde{\mathcal{V}}$ etc.
- Design quantum state with $Q^*_\epsilon(\tilde{v})$ as amplitudes
- Use properties of translation invariant kernel to obtain decomposition of integral operator into simple components: a full rank diagonal operator and the QFT: **technical tools** are perfect reconstruction of the kernel via translation invariance, regularisation
- Implement using QRAM, QSVT, QFT

Proof techniques

- **Want to sample from**:

$$Q_\epsilon^*(\tilde{v}) \propto \left\langle \varphi(\tilde{v},\cdot) \left| \hat{\mathsf{q}}^\rho (\hat{\Sigma} + \epsilon \mathbb{1})^{-1} \right| \varphi(\tilde{v},\cdot) \right\rangle \tag{4}$$

- **Quantum state**[2] that does the job:

$$|\Psi\rangle^{XX'} \propto \sum_{\tilde{v}\in\tilde{\mathcal{X}}} \hat{\Sigma}_\epsilon^{-\frac{1}{2}} |\tilde{v}\rangle^X \otimes \sqrt{\underline{\mathsf{Q}}^\tau} \mathsf{F}_D^\dagger \sqrt{\hat{\mathsf{q}}^\rho} |\tilde{v}\rangle^{X'} \tag{5}$$

- **Decomposition of** $\hat{\Sigma}$:

$$\hat{\Sigma}_\epsilon \propto \sqrt{\hat{\mathsf{q}}^\rho} \cdot \mathsf{F}_D^\dagger \mathsf{Q}^\tau \mathsf{F}_D \cdot \sqrt{\hat{\mathsf{q}}^\rho} + \epsilon \mathbb{1}, \tag{6}$$

where F_D is the quantum fourier transform (QFT) on $\mathbb{C}^D$.

[2] 'Aggregated' weights $Q^\tau(\tilde{v}) := \sum_{v\in\mathbb{Z}^D} q^\tau(\tilde{v}+v)$

Input model

QRAM for quantum access to Big Data

- **Classically**: memory contents at address k retrievable in $\mathcal{O}(1)$ time

$$\text{RAM} : k \mapsto f(k)$$

- **Quantumly**: unitary version of RAM running in quantum superposition [e.g. Jiang et al. (2019), Hann et al. (2019)]

$$\text{QRAM} : \sum_k \alpha_k |k\rangle |0\rangle \mapsto \sum_k \alpha_k |k\rangle |f(k)\rangle$$

- Linear algebra algorithms using QRAM typically have complexity scaling with the **Frobenius norm** of the input matrix

Input model: $\hat{\mathrm{q}}^{\rho}$

QRAM for quantum access to Big Data

- probability measure $d\rho(x) = q^{\rho}(x)dx$; empirical pmf $\hat{q}^{\rho}(x)$
- With $\mathcal{O}(N)$ classical preprocessing to count the N input data points, construct $\log N$ depth binary tree for addressing data [e.g. Kerenidis & Prakash (2017)]
- We use $\hat{q}^{\rho}(x)$ embedded into a diagonal operator $\hat{\mathbf{q}}$

$$|0\rangle \mapsto \sum_{x} \sqrt{\hat{\mathbf{q}}^{\rho}(x)}\, |x\rangle$$

- Frobenius norm of $\sqrt{\hat{\mathbf{q}}}$ is unity since q is a probability distribution

Matrix functions and block-encodings

- First introduced to study Hamiltonian simulation, has found a variety of applications in the last few years.
- A block encoding U_A of a Hermitian A is a unitary that encodes a (sub-)normalised version of A in its top left block

$$U_A = \begin{pmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix},$$

where $\alpha \geq \|A\|$

1 A Gilyen et al. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, 2018 (arXiv:1806.01838).

2 S Subramanian et al. Implementing smooth functions of a Hermitian matrix on a quantum computer, 2018 (arXiv:1806.06885).

3 A Childs et al. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision, 2015 (arXiv:1511.02306).

Matrix functions and block encodings: $\hat{\Sigma}^{-1/2}$

Quantum Singular Value Transformations

- Quantum Singular Value Transformations can use block encodings to probabilistically implement non-unitary operators

$$U_A = \begin{pmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix} \mapsto \begin{pmatrix} \tilde{f}(A) & \cdot \\ \cdot & \cdot \end{pmatrix} \approx U_{f(A)},$$

$$\left\| \tilde{f}(A) - f(A) \right\| < \delta.$$

- Using the block encoding of U_A roughly $\widetilde{\mathcal{O}}\left(\kappa \cdot \text{poly}\log \frac{D}{\delta}\right)$ times, where κ lower bounds the condition number of A, one can obtain a block-encoding $A^{-1/2}$ to precision δ via the method of QSVT using polynomial approximations of the target function $x^{-1/2}$

Avoiding sparsity and low-rank assumptions

Quantum Fourier Transform

- The structure in this sampling problem and its deep relation to Fourier analysis arising from the use of translation invariant kernels allows us to avoid assuming sparse or low rank input, in contrast to seminal quantum algorithms such as Harrow et al. (matrix inversion, 2009), Lloyd et al. (PCA, 2013), or Kerenidis & Prakash (recommendation systems, 2016)
- Classical FFT on dimension N requires $\mathcal{O}(N \log N)$ time, whereas QFT requires only $\widetilde{\mathcal{O}}(\log^2 N)$ time
- Essentially, translation-invariance indicates circulant matrix structure, can be diagonalised by the FFT and then easily inverted in the Fourier basis. Source of classical bottleneck could be the $\mathcal{O}(\mathrm{e}^D)$ dimensionality of $\hat{\Sigma}$

Avoiding sparsity and low-rank assumptions

Resistance to dequantisation by low-rank methods

- Low rank + QRAM recently found to be dequantisable (Tang et al. (2018), Le Gall et al. (2019), Chia et al. (2019, morning session)) via low-rank + classical ℓ_2 sampling
- Because we work with a full rank operator that might also be dense and have large spectral norm, these existing dequantisation methods are not directly applicable

Odds & Ends: after sampling the $\tilde{v}_i$ and deciding $\varphi(\tilde{v}_i, \cdot)$, we do doubly stochastic gradient descent for regression to obtain the coefficients, classically in linear $\mathcal{O}(D)$ time, without losing our speedup

Summary

- **Novelty**: (1) A niche where big data has small Frobenius norm, suitable for QRAM (2) Non-sparse and full rank operator inverted by taking advantage of Fourier sparsity + QFT + Quantum Singular Value Transformations
- We also show that careful application of (doubly) stochastic gradient descent allows regression to learn the coefficients α_m in $\mathcal{O}(D)$ time, without canceling out the quantum speedup
- Hence widely applicable, promising candidate for 'killer applications' of Quantum Computing / Quantum Machine Learning
- **Drawbacks**: Practicality - not NISQ friendly, we focus on asymptotic complexity, resulting circuits are huge and require thousands of qubits, long coherence times and fault tolerance for QSVT

Outlook

Interesting open directions

- **Specific applications of our framework**: (binary) classification, SVMs, regression
- Other applications where an operator is sparse in a Fourier transformed representation
- **Non-QRAM input models**: classical access [e.g. Arunachalam et al. (arXiv:2010.02174)]
- Investigating the potential for **NISQ** application in a QRAM-free model, optimising circuit constructions

For more..

- **NeurIPS 2020**: H Yamasaki, S Subramanian, S Sonoda, M Koashi; Learning with Optimized Random Features: Exponential Speedup by Quantum Machine Learning without Sparsity and Low-Rank Assumptions
- **Full(er) paper**: arXiv:2004.10756
- **More verbose explanations and details**: S Subramanian (2020). Quantum Algorithms for Matrix Problems and Machine Learning (Doctoral thesis).

Thank you!