Learning with quantum kernels: early applications to material damage prediction

Giorgio Tosti Balducci

Delft University of Technology, The Netherlands

March 3, 2023

The team

- Boyang Chen - *Aerospace Engineering*
- Matthias Möller - *Applied Mathematics*
- Marc Gerritsma - *Aerospace Engineering*
- me - *Aerospace Engineering*




Quantum machine learning

Some informal definitions

Quantum-enhanced machine learning

Quantum computation to speed-up classical machine learning operations.

Example: SVM with quantum linear systems solver algorithm

Machine learning in quantum feature spaces

Information is *encoded* into quantum states and classical algorithms find the optimal model

Quantum machine learning

Some informal definitions

Quantum-enhanced machine learning

Quantum computation to speed-up classical machine learning operations.

Example: SVM with quantum linear systems solver algorithm

Machine learning in quantum feature spaces

Information is *encoded* into quantum states and classical algorithms find the optimal model

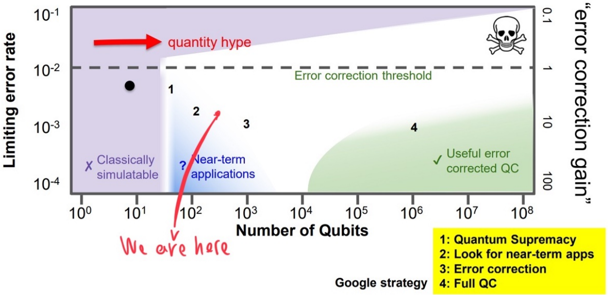
Note: not all quantum machine learning is done on quantum computers. QML can also mean to use classical machine learning for quantum mechanics.

Quantum machine learning in the near term

Quantum-enhanced ML gives provable speed-up, but requires **fault-tolerant hardware**. However . . .

Quantum machine learning in the near term

Quantum-enhanced ML gives provable speed-up, but requires **fault-tolerant hardware**. However ...



Source: https://www.hpcwire.com/2018/04/26/google-frames-quantum-race-as-two-dimensional/

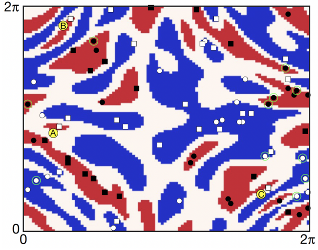
Quantum-enhanced ML is not viable in the near term.

ML in quantum feature spaces

Can we learn data patterns that are *hard to learn classically*?

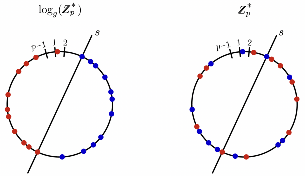
ML in quantum feature spaces

Can we learn data patterns that are *hard to learn classically*?



Source: Havlicek *et al.*, 2019, Nature Physics

Data generated from a 'quantum model' such to be hard to access classically.



Source: Liu *et al.*, 2020, arXiv:2010.02174

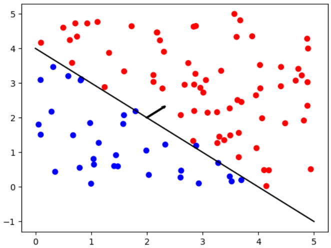
Data labelled by discrete loagarithm over a group generated by a large prime number.
Classically hard to compute, but efficient with Shor's quantum algorithm.

ML in quantum feature spaces

Are there *real world* datasets that are classically hard to classify, but 'understandable' for quantum computers?

Maximum margin classifier

Assume that the data is linearly separable (perhaps with some noise)



Model:

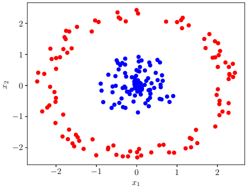
$$y = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right)$$

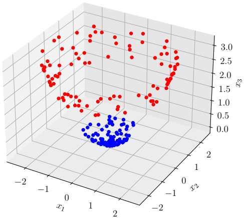
Support vector machine, primal form (*maximum geometric-margin classifier*)

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right) \geq 1, \quad i = 1, \dots, m \end{aligned}$$

Maximum margin classifier

What if the data is *not* linearly separable? We can introduce a feature map $\phi(x)$





$$\xrightarrow[\phi(x)=\left\{x_1, x_2, 0.5\left(x_1^2+x_2^2\right)\right\}]{\qquad\qquad\qquad}$$

Model: $y = \operatorname{sgn}\left(\langle \mathbf{w}, \phi(\mathbf{x})\rangle + b\right)$

Kernels

Representer theorem

The vector that expresses the optimal separating hyperplane in feature space is a linear combination of feature vectors.

$$\mathbf{w} = \sum_{i=1}^{N' \leq N} \alpha_i \phi\left(\mathbf{x}_i\right)$$

Kernels

Representer theorem

The vector that expresses the optimal separating hyperplane in feature space is a linear combination of feature vectors.

$$\mathbf{w} = \sum_{i=1}^{N' \leq N} \alpha_i \phi\left(\mathbf{x}_i\right)$$

So we can rewrite our model as

$$y = \operatorname{sgn}\left(\sum_{i=1}^{N' \leq N} \alpha_i \langle \phi\left(\mathbf{x}_i\right), \phi\left(\mathbf{x}\right) \rangle + b\right),$$

which is *linear in the feature space*.

The quantity $k\left(\mathbf{x}, \mathbf{x}'\right) = \langle \phi\left(\mathbf{x}\right), \phi\left(\mathbf{x}'\right) \rangle$ is called the *kernel* induced by the feature map ϕ.

Kernel trick

If we find an efficient explicit formula for the kernel, we don't need to compute the feature map directly and we can *implicitly* compute distances and classify in the feature space.

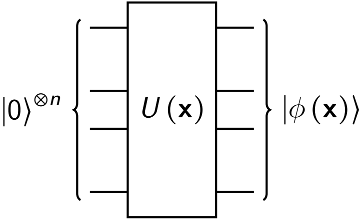
Example: quadratic kernel

$$
\begin{aligned}
\mathbf{x} &= \left(x_1,\, x_2\right)^{\top} \\
\phi\left(\mathbf{x}\right) &= \left(x_1^2,\, \sqrt{2}x_1x_2,\, x_2^2\right)^{\top} \\
k\left(\mathbf{x}, \mathbf{x}'\right) &= \left\langle \phi\left(\mathbf{x}\right),\, \phi\left(\mathbf{x}'\right)\right\rangle \\
&= \left(x_1x_1' + x_2x_2'\right)^2 \\
&= \left\langle \mathbf{x},\, \mathbf{x}'\right\rangle^2
\end{aligned}
$$

i.e. the kernel can be computed in the original space of data.

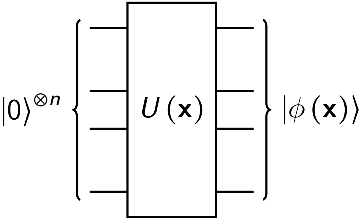
Quantum kernels

In terms of quantum states, we can encode $\mathbf{x}$ into a unitary operator $U(\mathbf{x})$. This creates $|\phi(\mathbf{x})\rangle = U(\mathbf{x})|0\rangle$

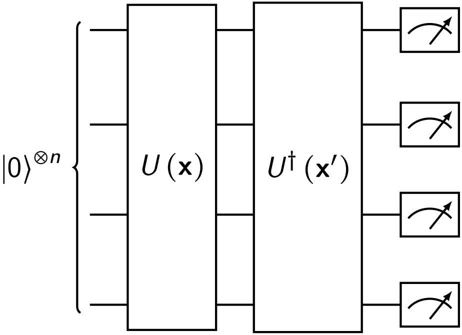


Quantum kernels

In terms of quantum states, we can encode $\mathbf{x}$ into a unitary operator $U(\mathbf{x})$. This creates $|\phi(\mathbf{x})\rangle = U(\mathbf{x})|0\rangle$

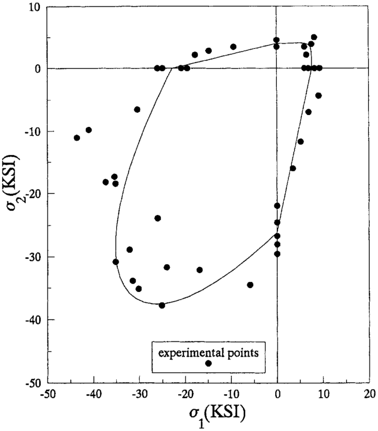


Defining $\rho(\mathbf{x}) = |\mathbf{x}\rangle\langle\mathbf{x}'|$, the *quantum kernel* $k(\mathbf{x},\mathbf{x}') = \mathrm{Tr}\{\rho(\mathbf{x})\rho(\mathbf{x}')\}$ implicitly accesses the 4^n-dimensional complex space



Material damage prediction

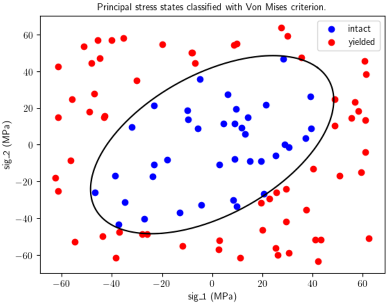
- Failure criteria are models of material failure. They generally come from semi-empirical relations (data + physics)
- They are effectively *classification problems*, where the decision boundary is the *failure envelope*



Source: Echaabi *et al.*, 1996, Polymer Composites

Von Mises criterion

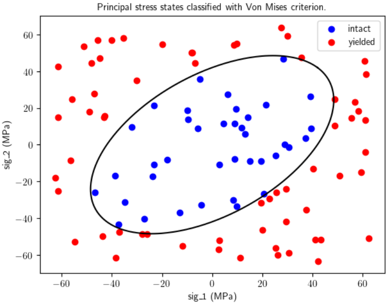
A `hello, world!` example in material damage prediction



$$\sigma_1^2 + \sigma_2^2 - \sigma_1 \sigma_2 - \sigma_y^2 = 0$$

Von Mises criterion

A `hello, world!` example in material damage prediction



$$\sigma_1^2 + \sigma_2^2 - \sigma_1 \sigma_2 - \sigma_y^2 = 0$$

'Analytic' feature map and kernel:

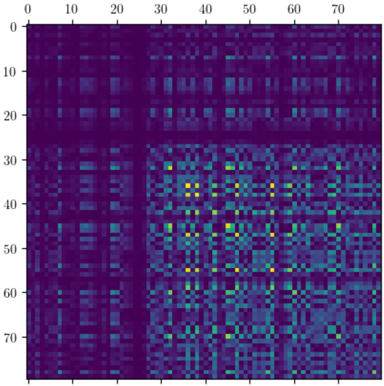
$$\phi(\mathbf{x}) = \left(x_1^2, x_2^2, x_1 x_2\right)^\top$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = \langle \phi(\mathbf{x}), \phi\left(\mathbf{x}'\right) \rangle$$

Von Mises criterion

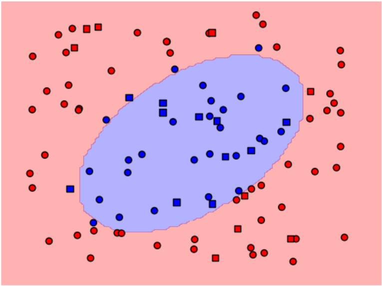
Classification with classical quadratic kernel

$$k(\mathbf{x}, \mathbf{x}') = \left(\gamma \langle \mathbf{x}, \mathbf{x}' \rangle + r\right)^2$$

Kernel matrix



Failure envelope



Von Mises criterion

Using a quantum 'quadratic' kernel

Strictly speaking, if we 'simply' encode **x** as the quantum state $|x\rangle$, it seems that we recover the quadratic kernel.

$$\phi\left(\mathbf{x}\right) = |x\rangle \longrightarrow k\left(\mathbf{x}, \mathbf{x}'\right) = \left|\mathbf{x}^{\dagger}\mathbf{x}'\right|^{2}$$

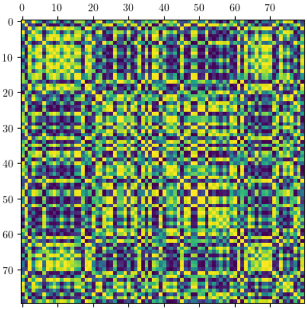
Von Mises criterion

Using a quantum 'quadratic' kernel

Strictly speaking, if we 'simply' encode $\mathbf{x}$ as the quantum state $|x\rangle$, it seems that we recover the quadratic kernel.

$$\phi(\mathbf{x}) = |x\rangle \longrightarrow k(\mathbf{x}, \mathbf{x}') = \left|\mathbf{x}^\dagger \mathbf{x}'\right|^2$$

However, the kernel matrix does not show any class separation...



Von Mises criterion

Using a quantum ‘quadratic’ kernel

Reason: normalization in the original feature space confuses the data

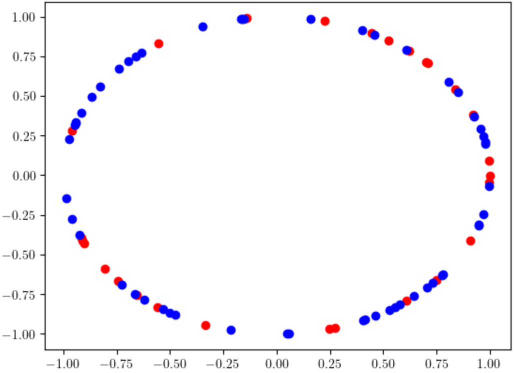
$$\phi\left(\mathbf{x}\right) = \left(x_1, x_2\right) / \|\mathbf{x}\|$$

Von Mises criterion

Using a quantum 'quadratic' kernel

Reason: normalization in the original feature space confuses the data

$$\phi(\mathbf{x}) = (x_1, x_2) / \|\mathbf{x}\|$$

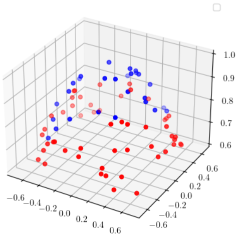


Von Mises criterion

Using the same kernel in a 4-dimensional space of unit vectors

Hack: add a 3rd dimension and normalize on a semi-sphere.

$$\phi(\mathbf{x}) = (x_1, x_2, 1, 0)^\top / \|\mathbf{x}\|$$

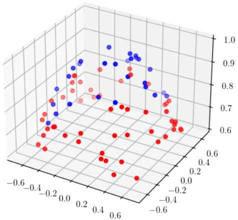


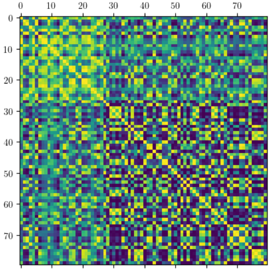
Von Mises criterion

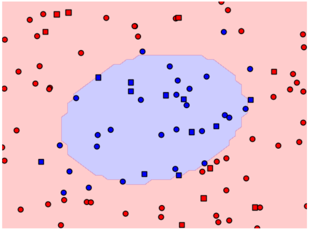
Using the same kernel in a 4-dimensional space of unit vectors

Hack: add a 3rd dimension and normalize on a semi-sphere.

$$\phi(\mathbf{x}) = (x_1, x_2, 1, 0)^\top / \|\mathbf{x}\|$$







Von Mises criterion

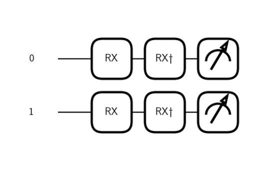
Rotation encoding kernels

Can we do better and avoid feature pre-processing that can confuse the label spaces?

Rotation encoding

Encode every feature as the rotation angle of a gate.

$$\phi(\mathbf{x}) = R(x_1) \otimes R(x_2) |0\rangle^{\otimes 2}$$



Von Mises criterion

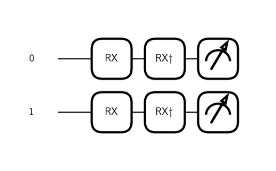
Rotation encoding kernels

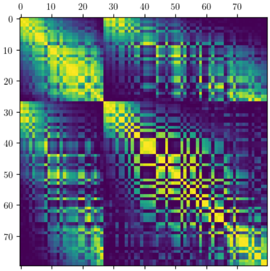
Can we do better and avoid feature pre-processing that can confuse the label spaces?

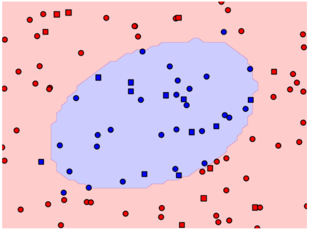
Rotation encoding

Encode every feature as the rotation angle of a gate.

$$\phi(\mathbf{x}) = R(x_1) \otimes R(x_2) \, |0\rangle^{\otimes 2}$$







Summary

Classifying stresses based on the Von Mises criterion has little direct interest, but

Summary

Classifying stresses based on the Von Mises criterion has little direct interest, but

1. It provides a reality-based example to compare the accuracies of classical and quantum kernels.

Summary

Classifying stresses based on the Von Mises criterion has little direct interest, but

1. It provides a reality-based example to compare the accuracies of classical and quantum kernels.
2. It highlights the key aspects of data encoding in quantum states
   - normalization
   - data scaling
   - data dimensionality

Where to go from here

- Study more complex datasets, where the 'ground truth' is unknown
  - Experimental data
  - Numerical, high-fidelity data

[1] La Rocca *et al.*, 2022, PRX Quantum

Where to go from here

- Study more complex datasets, where the 'ground truth' is unknown
  - Experimental data
  - Numerical, high-fidelity data
- Further investigate near-term encodings for damage prediction
  - In particular, quantum kernels generated by Hamiltonian evolution ($e^{iH\mathbf{x}}$) correspond to Fourier series in the data

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{n}, \mathbf{n}' \in \Omega} c_{\mathbf{n}, \mathbf{n}'} e^{i\mathbf{n}\mathbf{x}} e^{i\mathbf{n}'\mathbf{x}'}$$

[1] La Rocca *et al.*, 2022, PRX Quantum

Where to go from here

- Study more complex datasets, where the 'ground truth' is unknown
  - Experimental data
  - Numerical, high-fidelity data
- Further investigate near-term encodings for damage prediction
  - In particular, quantum kernels generated by Hamiltonian evolution ($e^{iH\mathbf{x}}$) correspond to Fourier series in the data

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{n}, \mathbf{n}' \in \Omega} c_{\mathbf{n}, \mathbf{n}'} e^{i\mathbf{n}\mathbf{x}} e^{i\mathbf{n}'\mathbf{x}'}$$

- Encode problem knowledge in the circuit to have control over the c_{n_i, n_j} coefficients
  - e.g. encoding symmetries [1]

[1] La Rocca *et al.*, 2022, PRX Quantum

Thank you!

Dual formulation of the maximum margin classifier

$$\begin{aligned}
\max_{\alpha} \quad & C(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\
\text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \\
& \sum_i \alpha_i y^{(i)} = 0
\end{aligned}$$

Classical kernels

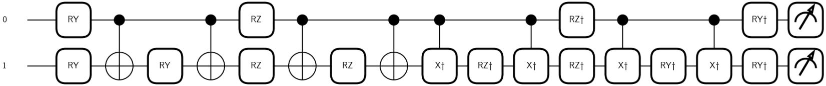
Name	Expression
polynomial	$(\gamma\langle \mathbf{x}, \mathbf{x}' \rangle + c_0)^d$
sigmoid	$\tanh(\gamma\langle \mathbf{x}, \mathbf{x}' \rangle + c_0)$
RBF	$\exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2\right)$

Table: Some popular kernels

Amplitude encoding circuits

1 qubit

2 qubits



Single rotation kernel encoding for 2 qubits

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \prod_{i=1}^{N} \cos\left(x_i - x_i'\right)$$

With 2 features ($\mathbf{x} = \left(x_1, x_2\right)^\top$),

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \cos\left(x_1 - x_1'\right) \cos\left(x_2 - x_2'\right)$$

Substitute $z_i = \left(x_i - x_i'\right)$ and Mc Laurin expand

$$\begin{aligned} k\left(\mathbf{x}, \mathbf{x}'\right) &\approxeq \left(1 - \frac{1}{2} z_1^2\right)\left(1 - \frac{1}{2} z_2^2\right) \\ &= 1 - \frac{1}{2} z_2^2 - \frac{1}{2} z_1^2 + \frac{1}{4} z_1^2 z_2^2 \\ &= \dots \end{aligned}$$

Re-substituting $\mathbf{x}$, $\mathbf{x}'$, one finds $x_i^2 x_i'^2$ and $x_i x_j x_i' x_j'$ terms as in the classical quadratic kernel, as well as additional 4$^{\text{th}}$ order terms in the features.