Kernel Methods in Quantum Machine Learning

Alex Della Schiava

Università degli Studi di Udine (DMIF) - Foundations of Neural Networks

September 9th, 2022

Quantum Machine Learning

Quantum Machine Learning (QML): Learning from data exploiting Quantum Computing.

Quantum Machine Learning

Quantum Machine Learning (QML): Learning from data exploiting Quantum Computing.

- Herein the focus is on learning from **classical data** through **quantum algorithms**.

Quantum Machine Learning

Quantum Machine Learning (QML): Learning from data exploiting Quantum Computing.

- Herein the focus is on learning from **classical data** through **quantum algorithms**.
- Initially all about *forcing* Quantum Computing into Deep Learning techniques.

Quantum Machine Learning

Quantum Machine Learning (QML): Learning from data exploiting Quantum Computing.

- Herein the focus is on learning from **classical data** through **quantum algorithms**.
- Initially all about *forcing* Quantum Computing into Deep Learning techniques.

Is there an alternative solution?

Quantum Machine Learning

Quantum Machine Learning (QML): Learning from data exploiting Quantum Computing.

- Herein the focus is on learning from **classical data** through **quantum algorithms**.
- Initially all about *forcing* Quantum Computing into Deep Learning techniques.

Is there an alternative solution?

M. Schuld, N. Killoran (2018), Quantum machine learning in feature Hilbert spaces

$$\Downarrow$$

Yes, **Kernel Methods.**

QML meets Kernel Methods

Current QML Computations (on classical data) work very much **similarly** to Kernel Methods:

QML meets Kernel Methods

Current QML Computations (on classical data) work very much **similarly** to Kernel Methods:

1. **Encode** data into a higher-dimensional space.] **Map**

QML meets Kernel Methods

Current QML Computations (on classical data) work very much **similarly** to Kernel Methods:

1. **Encode** data into a higher-dimensional space.
2. **Implicitly** manipulate higher-dimensional data.

Map

QML meets Kernel Methods

Current QML Computations (on classical data) work very much **similarly** to Kernel Methods:

1. **Encode** data into a higher-dimensional space.
2. **Implicitly** manipulate higher-dimensional data.
3. **Measure** higher-dimensional metrics over the initial data.

Map

Inner product

QML meets Kernel Methods

Current QML Computations (on classical data) work very much **similarly** to Kernel Methods:

1. **Encode** data into a higher-dimensional space.
2. **Implicitly** manipulate higher-dimensional data.
3. **Measure** higher-dimensional metrics over the initial data.

Map

Inner product

The **goal** of this talk is to formalize this apparent similarity.

QML meets Kernel Methods

Current QML Computations (on classical data) work very much **similarly** to Kernel Methods:

1. **Encode** data into a higher-dimensional space.
2. **Implicitly** manipulate higher-dimensional data.
3. **Measure** higher-dimensional metrics over the initial data.

Map (item 1)

Inner product (items 2–3)

The **goal** of this talk is to formalize this apparent similarity.

$\Downarrow$

Free-pass to exploit Kernel Theory in QML.

Outline

Quantum Mechanics

Dirac Notation

Given a vector $\boldsymbol{\psi} \in \mathbb{C}^2$:

$$\boldsymbol{\psi} = |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}; \quad \boldsymbol{\psi}^\dagger = \langle\psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}$$

Quantum Mechanics

Dirac Notation

Given a vector $\boldsymbol{\psi} \in \mathbb{C}^2$:

$$\boldsymbol{\psi} = |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}; \quad \boldsymbol{\psi}^\dagger = \langle\psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}$$

- Quantum mechanics talks about **quantum systems**.

Quantum Mechanics

Dirac Notation

Given a vector $\boldsymbol{\psi} \in \mathbb{C}^2$:

$$\boldsymbol{\psi} = |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}; \quad \boldsymbol{\psi}^\dagger = \langle\psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}$$

- Quantum mechanics talks about **quantum systems**.
- A quantum system is *fully described* by a **unit column vector**:

$$\boldsymbol{\psi} = |\psi\rangle \in \mathbb{C}^n \qquad\qquad \| \, |\psi\rangle \, \| = 1$$

Quantum Mechanics

Dirac Notation

Given a vector $\boldsymbol{\psi} \in \mathbb{C}^2$:

$$\boldsymbol{\psi} = |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}; \quad \boldsymbol{\psi}^\dagger = \langle\psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}$$

- Quantum mechanics talks about **quantum systems**.
- A quantum system is *fully described* by a **unit column vector**:

$$\boldsymbol{\psi} = |\psi\rangle \in \mathbb{C}^n \qquad\qquad \|\,|\psi\rangle\,\| = 1$$

- $\mathbb{C}^n$ is a complex-valued Hilbert space.
  - Comes, for free, with an **inner product** (dot product):

$$(|\psi\rangle\,, |\varphi\rangle)_{\mathbb{C}^n} = \langle\varphi|\psi\rangle_{\mathbb{C}^n}$$

Quantum Mechanics (cont'd)

Two ways to formulate quantum mechanics:

Example (A Qubit)

Consider a qubit described by the state vector $|\psi\rangle \in \mathbb{C}^2$.

Quantum Mechanics (cont'd)

Two ways to formulate quantum mechanics:

Example (A Qubit)

Consider a qubit described by the state vector $|\psi\rangle \in \mathbb{C}^2$.

State vector

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$|\psi\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$$

$$\|\,|\psi\rangle\,\| = \sqrt{|\alpha|^2 + |\beta|^2} = 1$$

- **Note:** $|0\rangle = \begin{bmatrix}1 \\ 0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0 \\ 1\end{bmatrix}$.

Quantum Mechanics (cont'd)

Two ways to formulate quantum mechanics:

Example (A Qubit)

Consider a qubit described by the state vector $|\psi\rangle \in \mathbb{C}^2$.

State vector	**Density matrix**
$\lvert\psi\rangle = \alpha \lvert 0\rangle + \beta \lvert 1\rangle$	$\rho = \lvert\psi\rangle \langle\psi\rvert \in \mathbb{C}^{2\times 2}$
$\lvert\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$	$\rho = \begin{bmatrix} \lvert\alpha\rvert^2 & \alpha \cdot \beta^* \\ \beta \cdot \alpha^* & \lvert\beta\rvert^2 \end{bmatrix}$
$\lVert \lvert\psi\rangle \rVert = \sqrt{\lvert\alpha\rvert^2 + \lvert\beta\rvert^2} = 1$	$\mathsf{tr}(\rho) = \lvert\alpha\rvert^2 + \lvert\beta\rvert^2 = 1$

- **Note:** $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.
- $\rho = \rho^\dagger$: Is **Hermitian**.

Quantum Mechanics (cont'd)

- The **evolution** of a quantum system is described in terms of **unitary transformations** U.
  - $U^\dagger U = UU^\dagger = I_{\mathbb{C}^{n\times n}}$
- Given a state $\rho = |\psi\rangle\langle\psi|$:

$$|\psi'\rangle = U|\psi\rangle \quad \rho' = U\rho U^\dagger$$

- where $\rho' = |\psi'\rangle\langle\psi'|$ is the post-evolution state.

Quantum Mechanics (cont'd)

- A **projective measurement** over a quantum state is described by a Hermitian operator $\mathcal{M} = \mathcal{M}^\dagger$.

Quantum Mechanics (cont'd)

- A **projective measurement** over a quantum state is described by a Hermitian operator $\mathcal{M} = \mathcal{M}^\dagger$.
- $\mathcal{M}$ has spectral decomposition: $\mathcal{M} = \sum_\lambda \lambda P_\lambda \quad = PDP^\dagger$
  - λ are the possible **outcomes** of the measurement.
  - P_λ are projectors over eigenspace λ.

Quantum Mechanics (cont'd)

- A **projective measurement** over a quantum state is described by a Hermitian operator $\mathcal{M} = \mathcal{M}^\dagger$.
- $\mathcal{M}$ has spectral decomposition: $\mathcal{M} = \sum_\lambda \lambda P_\lambda \quad = PDP^\dagger$
  - λ are the possible **outcomes** of the measurement.
  - P_λ are projectors over eigenspace λ.
- Measurement is a **probabilistic operation** which **alters the state** of the system.

Quantum Mechanics (cont'd)

- A **projective measurement** over a quantum state is described by a Hermitian operator $\mathcal{M} = \mathcal{M}^\dagger$.
- $\mathcal{M}$ has spectral decomposition: $\mathcal{M} = \sum_\lambda \lambda P_\lambda \quad = PDP^\dagger$
  - λ are the possible **outcomes** of the measurement.
  - P_λ are projectors over eigenspace λ.
- Measurement is a **probabilistic operation** which **alters the state** of the system.
- Since it is **probabilistic**, one can compute the **expected outcome**:

$$\mathbf{E}(\mathcal{M}) = \langle\psi| \mathcal{M} |\psi\rangle \qquad\qquad \mathbf{E}(\mathcal{M}) = \mathsf{tr}\big[\rho\mathcal{M}\big]$$

Quantum Mechanics (cont'd)

- Herein it is best to focus on **density matrices**.
    - It will make results prettier.

Quantum Mechanics (cont'd)

- Herein it is best to focus on **density matrices**.
  - It will make results prettier.
- The space of density matrices $\mathbb{C}^{n\times n}$ is equipped with the **Hilbert-Schmidt inner product** $\langle\cdot|\cdot\rangle_{\mathbb{C}^{n\times n}}$.

Quantum Mechanics (cont'd)

- Herein it is best to focus on **density matrices**.
  - It will make results prettier.
- The space of density matrices $\mathbb{C}^{n \times n}$ is equipped with the **Hilbert-Schmidt inner product** $\langle \cdot | \cdot \rangle_{\mathbb{C}^{n \times n}}$.
- Given $\rho = |\psi\rangle \langle\psi|$ and $\sigma = |\varphi\rangle \langle\varphi|$ where $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^n$:

$$\langle \sigma | \rho \rangle = \mathrm{tr}[\sigma^\dagger \rho] = \mathrm{tr}[\sigma \rho]$$

Quantum Mechanics (cont'd)

- Herein it is best to focus on **density matrices**.
  - It will make results prettier.
- The space of density matrices $\mathbb{C}^{n\times n}$ is equipped with the **Hilbert-Schmidt inner product** $\langle\cdot|\cdot\rangle_{\mathbb{C}^{n\times n}}$.
- Given $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\varphi\rangle\langle\varphi|$ where $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^n$:

$$\langle\sigma|\rho\rangle = \mathsf{tr}[\sigma^\dagger\rho] = \mathsf{tr}[\sigma\rho]$$

- Does it recall something? (Hint: $\mathbf{E}(\mathcal{M}) = \mathsf{tr}\left[\rho\mathcal{M}\right]$)

Quantum Mechanics (cont'd)

- Herein it is best to focus on **density matrices**.
  - It will make results prettier.
- The space of density matrices $\mathbb{C}^{n\times n}$ is equipped with the **Hilbert-Schmidt inner product** $\langle\cdot|\cdot\rangle_{\mathbb{C}^{n\times n}}$.
- Given $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\varphi\rangle\langle\varphi|$ where $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^n$:

$$\langle\sigma|\rho\rangle = \mathsf{tr}[\sigma^\dagger\rho] = \mathsf{tr}[\sigma\rho]$$

- Does it recall something? (Hint: $\mathbf{E}(\mathcal{M}) = \mathsf{tr}\left[\rho\mathcal{M}\right]$)
- With a little bit of calculation:

$$\mathsf{tr}[\sigma\rho] = |\,\langle\varphi|\psi\rangle_{\mathbb{C}^n}\,|^2$$

Outline

Kernel Methods

- Consider a **domain set** $\mathcal{X}$ and a **label set** $\mathcal{Y} = \{\pm 1\}$.
- A possibly *non-linearly separable* **training set**:

$$\mathcal{D} = \left\{(x^1, y^1), \ldots, (x^M, y^M) \in \mathcal{X} \times \mathcal{Y}\right\}$$

Kernel Methods

- Consider a **domain set** $\mathcal{X}$ and a **label set** $\mathcal{Y} = \{\pm 1\}$.
- A possibly *non-linearly separable* **training set**:

$$\mathcal{D} = \left\{(x^1, y^1), \ldots, (x^M, y^M) \in \mathcal{X} \times \mathcal{Y}\right\}$$

- Consider a **feature map** $\phi : \mathcal{X} \to \mathcal{F}$.
  - $\mathcal{F}$ is the **feature space** with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$.

Kernel Methods

- Consider a **domain set** $\mathcal{X}$ and a **label set** $\mathcal{Y} = \{\pm 1\}$.
- A possibly *non-linearly separable* **training set**:

$$\mathcal{D} = \left\{(x^1, y^1), \ldots, (x^M, y^M) \in \mathcal{X} \times \mathcal{Y}\right\}$$

- Consider a **feature map** $\phi : \mathcal{X} \to \mathcal{F}$.
  - $\mathcal{F}$ is the **feature space** with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$.
- Define the **kernel function**:

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{C}, \qquad \kappa(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{F}}$$

- κ *implicitly* exploits the **feature map**.

Kernel Methods (cont'd)

- **Goal:** *Learn* a **linear model** f in the feature space $\mathcal{F}$ of the form:

$$f(x) = \langle \phi(x), w \rangle_{\mathcal{F}}\,, \text{ for some } w \in \mathcal{F}$$

Kernel Methods (cont'd)

- **Goal:** *Learn* a **linear model** f in the feature space $\mathcal{F}$ of the form:

$$f(x) = \langle \phi(x), w \rangle_{\mathcal{F}}\,, \text{ for some } w \in \mathcal{F}$$

  - w defines the *linear decision boundary* in $\mathcal{F}$...
  - ...but there is no need to explicitly learn it!

Kernel Methods (cont'd)

- **Goal:** *Learn* a **linear model** f in the feature space $\mathcal{F}$ of the form:

$$f(x) = \langle \phi(x), w \rangle_{\mathcal{F}}, \text{ for some } w \in \mathcal{F}$$

  - w defines the *linear decision boundary* in $\mathcal{F}$...
  - ...but there is no need to explicitly learn it!
- **Key point:** $f(\cdot)$ can be expressed through calls of the kernel κ over x from the training set $\mathcal{D}$.

Outline

Kernel Methods for QML

- Following the work done in:

  M. Schuld (2021), Supervised quantum machine learning models are kernel methods

- Show that elements of Kernel Theory can be *extended* to a Quantum Computation on classical data.
- **Don't get lost.** Map data so that kernel functions may be executed on quantum machinery.

Outline

Quantum encodings are feature maps

Encoding classical data into quantum systems $\stackrel{?}{\equiv}$ **Feature map ϕ**

$\Downarrow$

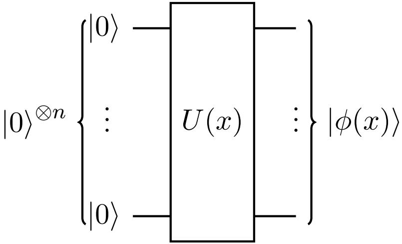
Done through quantum computation (circuit)

Quantum encodings are feature maps

Encoding classical data into quantum systems $\stackrel{?}{\equiv}$ **Feature map ϕ**

$\Downarrow$

Done through quantum computation (circuit)



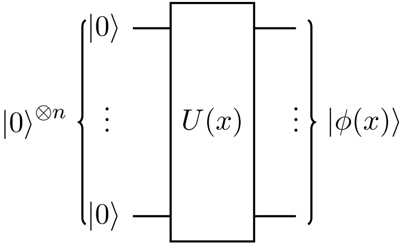
1. Prepare $|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$

Quantum encodings are feature maps

Encoding classical data into quantum systems $\stackrel{?}{\equiv}$ **Feature map ϕ**

$\Downarrow$

Done through quantum computation (circuit)



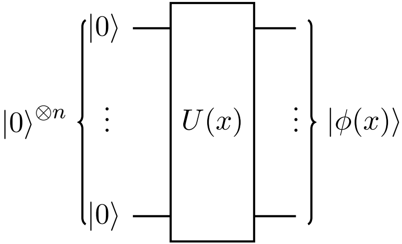
1. Prepare $|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$
2. With $x \in \mathcal{X}$ apply $U(x)\,|0\rangle^{\otimes n}$.
   - $U(x)$ depends on x.

Quantum encodings are feature maps

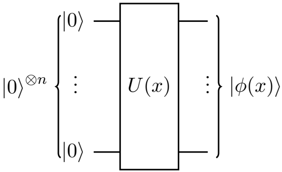
Encoding classical data into quantum systems $\stackrel{?}{=}$ **Feature map ϕ**

$\Downarrow$

Done through quantum computation (circuit)



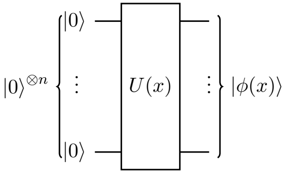
1. Prepare $|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$
2. With $x \in \mathcal{X}$ apply $U(x)\,|0\rangle^{\otimes n}$.
   - $U(x)$ depends on x.
3. Denote output as $|\phi(x)\rangle \in \mathbb{C}^{2^n}$.



1. Prepare $|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$.
2. With $x \in \mathcal{X}$ apply $U(x)\,|0\rangle^{\otimes n}$.
   - $U(x)$ depends on x.
3. Denote output as $|\phi(x)\rangle \in \mathbb{C}^{2^n}$.

- Seems like we made it! Just define the **feature map** as:

$$\phi : \mathcal{X} \to \mathbb{C}^{2^n}, \quad \phi(x) = |\phi(x)\rangle$$



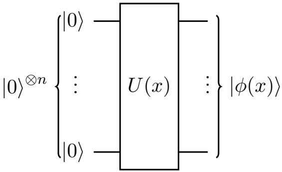
1. Prepare $|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$.
2. With $x \in \mathcal{X}$ apply $U(x)\,|0\rangle^{\otimes n}$.
   - $U(x)$ depends on x.
3. Denote output as $|\phi(x)\rangle \in \mathbb{C}^{2^n}$.

- Seems like we made it! Just define the **feature map** as:

$$\phi : \mathcal{X} \to \mathbb{C}^{2^n}, \quad \phi(x) = |\phi(x)\rangle$$

- This works fine, but for the sake of a better representation:

$$\phi : \mathcal{X} \to \mathbb{C}^{2^n \times 2^n}, \quad \phi(x) = |\phi(x)\rangle \langle \phi(x)| = \rho(x)$$



1. Prepare $|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$.
2. With $x \in \mathcal{X}$ apply $U(x)\,|0\rangle^{\otimes n}$.
   - $U(x)$ depends on x.
3. Denote output as $|\phi(x)\rangle \in \mathbb{C}^{2^n}$.

- Seems like we made it! Just define the **feature map** as:

$$\phi : \mathcal{X} \to \mathbb{C}^{2^n}, \quad \phi(x) = |\phi(x)\rangle$$

- This works fine, but for the sake of a better representation:

$$\phi : \mathcal{X} \to \mathbb{C}^{2^n \times 2^n}, \quad \phi(x) = |\phi(x)\rangle \langle\phi(x)| = \rho(x)$$

- Denote ϕ as the **data-encoding feature map**.
  - $\mathbb{C}^{2^n \times 2^n}$ is our new **feature space** $\mathcal{F}$.

Data-encoding example

Example (Basis encoding)

- Assume $\mathcal{X} = \{0,1\}^3$.
- Consider the state space $\mathbb{C}^{2^3}$, with computational basis:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}; \quad \dots \quad |7\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

- Denoting i_x as the decimal conversion of x.
- Define $\phi : \mathcal{X} \to \mathbb{C}^{2^3 \times 2^3}$ as:

$$\phi(x) = |i_x\rangle \langle i_x|$$

Outline

Quantum Kernels

- $\mathcal{F} = \mathbb{C}^{2^n \times 2^n}$ was shown to be a legit **feature space**...
  - ...but does it give rise to a legit **kernel**?

Quantum Kernels

- $\mathcal{F} = \mathbb{C}^{2^n \times 2^n}$ was shown to be a legit **feature space**...
  - ...but does it give rise to a legit **kernel**?

Definition (Quantum Kernel)

Given a data-encoding feature map $\phi : \mathcal{X} \to \mathcal{F}$, a quantum kernel is a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ such that, for any $x, x' \in \mathcal{X}$:

$$\kappa(x, x') = \langle \rho(x') | \rho(x) \rangle_{\mathcal{F}}$$

Quantum Kernels

- $\mathcal{F} = \mathbb{C}^{2^n \times 2^n}$ was shown to be a legit **feature space**...
  - ...but does it give rise to a legit **kernel**?

Definition (Quantum Kernel)

Given a data-encoding feature map $\phi : \mathcal{X} \to \mathcal{F}$, a quantum kernel is a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ such that, for any $x, x' \in \mathcal{X}$:

$$\kappa(x, x') = \langle \rho(x') | \rho(x) \rangle_{\mathcal{F}}$$

- The Hilbert-Schmidt inner product induces the following **quantum kernel**:

$$\kappa(x, x') = \langle \rho(x') | \rho(x) \rangle_{\mathcal{F}} = \mathsf{tr}\big[\rho(x')\rho(x)\big]$$

Quantum Kernels

- $\mathcal{F} = \mathbb{C}^{2^n \times 2^n}$ was shown to be a legit **feature space**...
  - ...but does it give rise to a legit **kernel**?

Definition (Quantum Kernel)

Given a data-encoding feature map $\phi : \mathcal{X} \to \mathcal{F}$, a quantum kernel is a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ such that, for any $x, x' \in \mathcal{X}$:

$$\kappa(x, x') = \langle \rho(x') | \rho(x) \rangle_{\mathcal{F}}$$

- The Hilbert-Schmidt inner product induces the following **quantum kernel**:

$$\kappa(x, x') = \langle \rho(x') | \rho(x) \rangle_{\mathcal{F}} = \mathsf{tr}\left[\rho(x')\rho(x)\right]$$

- By definition of inner product, it's a **valid kernel**.

Outline

1. Quantum Mechanics

2. Overview on Kernel Methods

3. Kernel Methods for Quantum Machine Learning
   - Quantum encodings are feature maps
   - Quantum Kernels are valid kernels
   - Quantum Models are Linear Models
   - Where to search Quantum Models
   - Finding Optimal Quantum Models

Linear Models

- A **linear model** is the **hypothesis** of our learning machinery.

Linear Models

- A **linear model** is the **hypothesis** of our learning machinery.
- More formally, in the case of **feature spaces**:

Definition (Linear Model)

Given a feature map $\phi : \mathcal{X} \to \mathcal{F}$ and $w \in \mathcal{F}$, a **linear model** is any function $f_w : \mathcal{X} \to \mathbb{C}$ of the form:

$$f_w(x) = \langle \phi(x), w \rangle_{\mathcal{F}}$$

Linear Models

- A **linear model** is the **hypothesis** of our learning machinery.
- More formally, in the case of **feature spaces**:

Definition (Linear Model)

Given a feature map $\phi : \mathcal{X} \to \mathcal{F}$ and $w \in \mathcal{F}$, a **linear model** is any function $f_w : \mathcal{X} \to \mathbb{C}$ of the form:

$$f_w(x) = \langle \phi(x), w \rangle_{\mathcal{F}}$$

- How to translate such concept in the quantum case?

Quantum Models

Goal: Make a prediction onto the state of a quantum system.

Quantum Models

Goal: Make a prediction onto the state of a quantum system.

$$\Downarrow$$

Tool: A **projective measurement** $\mathcal{M}$.

Quantum Models

Goal: Make a prediction onto the state of a quantum system.

$$\Downarrow$$

Tool: A **projective measurement** $\mathcal{M}$.

$$\Downarrow$$

Idea: Define the prediction as the **expected outcome** of $\mathcal{M}$.

Quantum Models

Goal: Make a prediction onto the state of a quantum system.

$$\Downarrow$$

Tool: A **projective measurement** $\mathcal{M}$.

$$\Downarrow$$

Idea: Define the prediction as the **expected outcome** of $\mathcal{M}$.

Definition (Quantum Model)

Given the **data-encoding feature map** $\phi : \mathcal{X} \to \mathcal{F}$ and $\mathcal{M}$ a projective measurement on the density matrices space $\mathcal{F}$. A **quantum model** is a function $f_\mathcal{M}$ of the form:

$$f_\mathcal{M}(x) = \mathsf{tr}\big[\rho(x)\mathcal{M}\big]$$

Quantum Models are Linear Models

Definition (Quantum Model)

Given the **data-encoding feature map** $\phi : \mathcal{X} \to \mathcal{F}$ and $\mathcal{M}$ a projective measurement on the density matrices space $\mathcal{F}$. A **quantum model** is a function $f_{\mathcal{M}}$ of the form:

$$f_{\mathcal{M}}(x) = \mathsf{tr}\big[\rho(x)\mathcal{M}\big]$$

The definition has not only a semantic value, it is also very convenient:

Quantum Models are Linear Models

Definition (Quantum Model)

Given the **data-encoding feature map** $\phi : \mathcal{X} \to \mathcal{F}$ and $\mathcal{M}$ a projective measurement on the density matrices space $\mathcal{F}$. A **quantum model** is a function $f_{\mathcal{M}}$ of the form:

$$f_{\mathcal{M}}(x) = \mathsf{tr}\big[\rho(x)\mathcal{M}\big]$$

The definition has not only a semantic value, it is also very convenient:

1. $\mathcal{M} \in \mathcal{F}$ (*"onto the density matrices space"*)

Quantum Models are Linear Models

Definition (Quantum Model)

Given the **data-encoding feature map** $\phi : \mathcal{X} \to \mathcal{F}$ and $\mathcal{M}$ a projective measurement on the density matrices space $\mathcal{F}$. A **quantum model** is a function $f_{\mathcal{M}}$ of the form:

$$f_{\mathcal{M}}(x) = \mathsf{tr}\big[\rho(x)\mathcal{M}\big]$$

The definition has not only a semantic value, it is also very convenient:

1. $\mathcal{M} \in \mathcal{F}$ (*"onto the density matrices space"*)
2. $f_{\mathcal{M}}(x) = \mathsf{tr}\big[\rho(x)\mathcal{M}\big] = \langle \rho(x) | \mathcal{M} \rangle_{\mathcal{F}}$

Quantum Models are Linear Models

Definition (Quantum Model)

Given the **data-encoding feature map** $\phi : \mathcal{X} \to \mathcal{F}$ and $\mathcal{M}$ a projective measurement on the density matrices space $\mathcal{F}$. A **quantum model** is a function $f_{\mathcal{M}}$ of the form:

$$f_{\mathcal{M}}(x) = \mathsf{tr}\left[\rho(x)\mathcal{M}\right]$$

The definition has not only a semantic value, it is also very convenient:

1. $\mathcal{M} \in \mathcal{F}$ (*"onto the density matrices space"*)
2. $f_{\mathcal{M}}(x) = \mathsf{tr}\left[\rho(x)\mathcal{M}\right] = \langle \rho(x) | \mathcal{M} \rangle_{\mathcal{F}}$

Consequence of (1) and (2): $f_{\mathcal{M}}$ is a **linear model**.

Outline

A Search Space for Quantum Models

Goal: Search for a specific quantum model.

A Search Space for Quantum Models

Goal: Search for a specific quantum model. **Where?**

A Search Space for Quantum Models

Goal: Search for a specific quantum model. **Where?**

- Classical Kernel Theory: **RKHS** (Reproducing Kernel Hilbert Space).
  - Is generated by a valid kernel.
  - Corresponds to the space of linear models.

A Search Space for Quantum Models

Goal: Search for a specific quantum model. **Where?**

- Classical Kernel Theory: **RKHS** (Reproducing Kernel Hilbert Space).
  - Is generated by a valid kernel.
  - Corresponds to the space of linear models.
- Quantum Kernel Theory:

$$\text{Quantum Kernel } \kappa \text{ is a valid kernel.}$$

A Search Space for Quantum Models

Goal: Search for a specific quantum model. **Where?**

- Classical Kernel Theory: **RKHS** (Reproducing Kernel Hilbert Space).
  - Is generated by a valid kernel.
  - Corresponds to the space of linear models.
- Quantum Kernel Theory:

$$\text{Quantum Kernel } \kappa \text{ is a valid kernel.}$$
$$\Downarrow$$
$$\text{It generates a RKHS } F.$$

Goal: Search for a specific quantum model. **Where?**

- Classical Kernel Theory: **RKHS** (Reproducing Kernel Hilbert Space).
  - Is generated by a valid kernel.
  - Corresponds to the space of linear models.
- Quantum Kernel Theory:

$$\text{Quantum Kernel } \kappa \text{ is a valid kernel.}$$
$$\Downarrow$$
$$\text{It generates a RKHS } F.$$
$$\Downarrow$$
$$\textbf{Is } \boldsymbol{F} \textbf{ the space of quantum models?}$$

Building the RKHS

How to build the RKHS F of the quantum kernel κ:

- **Elements.** Functions of the form $f : \mathcal{X} \to \mathbb{R}$.

Building the RKHS

How to build the RKHS F of the quantum kernel κ:

- **Elements.** Functions of the form $f : \mathcal{X} \to \mathbb{R}$.
- **Building blocks.** For each $x \in \mathcal{X}$ *put* into F the function f:

$$f(\cdot) = \kappa(x, \cdot)$$

Building the RKHS

How to build the RKHS F of the quantum kernel κ:

- **Elements.** Functions of the form $f : \mathcal{X} \to \mathbb{R}$.
- **Building blocks.** For each $x \in \mathcal{X}$ *put* into F the function f:

$$f(\cdot) = \kappa(x, \cdot) = \langle \cdot | \rho(x) \rangle_{\mathcal{F}}$$

- **Note:** These are **quantum models** induced by elements of $\mathcal{X}$.

Building the RKHS

How to build the RKHS F of the quantum kernel κ:

- **Elements.** Functions of the form $f : \mathcal{X} \to \mathbb{R}$.
- **Building blocks.** For each $x \in \mathcal{X}$ *put* into F the function f:

$$f(\cdot) = \kappa(x, \cdot) = \langle \cdot | \rho(x) \rangle_{\mathcal{F}}$$

- **Note:** These are **quantum models** induced by elements of $\mathcal{X}$.

- F is the **space** spanned by such functions:

$$f(\cdot) = \sum_i \alpha_i \kappa(x_i, \cdot)$$

with $\alpha_i \in \mathbb{R}$ and $x_i \in \mathcal{X}$.

Building the RKHS (cont'd)

- Last ingredient: **Inner product** $\langle \cdot, \cdot \rangle_F$.

Building the RKHS (cont'd)

- Last ingredient: **Inner product** $\langle \cdot, \cdot \rangle_F$.
- Given $f, g \in F$ such that:

$$f(\cdot) = \sum_i \alpha_i \kappa(x_i, \cdot), \qquad g(\cdot) = \sum_j \beta_j \kappa(x_j, \cdot)$$

Building the RKHS (cont'd)

- Last ingredient: **Inner product** $\langle \cdot, \cdot \rangle_F$.
- Given $f, g \in F$ such that:

$$f(\cdot) = \sum_i \alpha_i \kappa(x_i, \cdot), \qquad g(\cdot) = \sum_j \beta_j \kappa(x_j, \cdot)$$

- Define $\langle f, g \rangle_F$ as:

$$\langle f, g \rangle_F = \sum_{i,j} \alpha_i \beta_j \kappa(x_i, x_j)$$

Building the RKHS (cont'd)

- Last ingredient: **Inner product** $\langle \cdot, \cdot \rangle_F$.
- Given $f, g \in F$ such that:

$$f(\cdot) = \sum_i \alpha_i \kappa(x_i, \cdot), \qquad g(\cdot) = \sum_j \beta_j \kappa(x_j, \cdot)$$

- Define $\langle f, g \rangle_F$ as:

$$\langle f, g \rangle_F = \sum_{i,j} \alpha_i \beta_j \kappa(x_i, x_j)$$

- **Reproducing property:**

$$\langle f, \kappa(x, \cdot) \rangle_F = \sum_i \alpha_i \kappa(x_i, x) = f(x)$$

The RKHS is the space of Quantum Models

Finally, it is all set for the following result:

Theorem

The RKHS F is the space of Quantum Models. That is, any $f \in F$ is a quantum model.

$$f(x) = \sum_i \alpha_i \kappa(x_i, x) = \mathsf{tr}\left[\mathcal{M}\rho(x)\right]$$

The RKHS is the space of Quantum Models

Finally, it is all set for the following result:

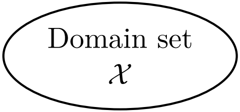
Theorem

The RKHS F is the space of Quantum Models. That is, any $f \in F$ is a quantum model.

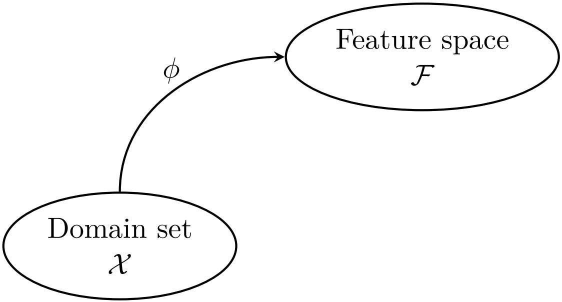
$$f(x) = \sum_i \alpha_i \kappa(x_i, x) = \mathsf{tr}\big[\mathcal{M}\rho(x)\big]$$

- **Side-note:** The structure of the RKHS is key in the definition of **Universal Kernel**.
  - Just like NNs, quantum models can be **Universal approximators**.

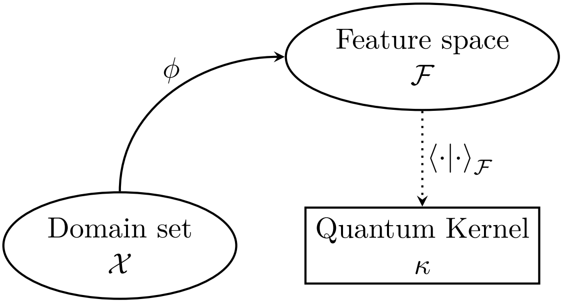
Conceptual summary



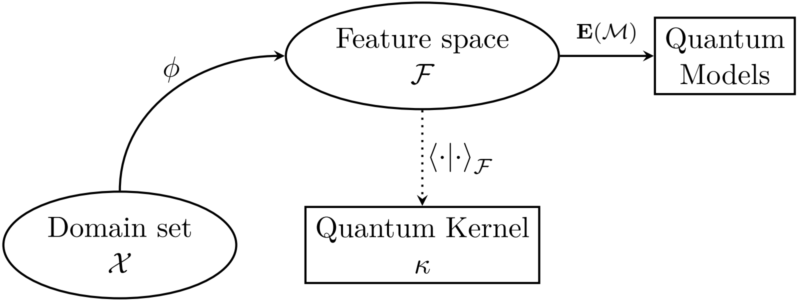
Conceptual summary



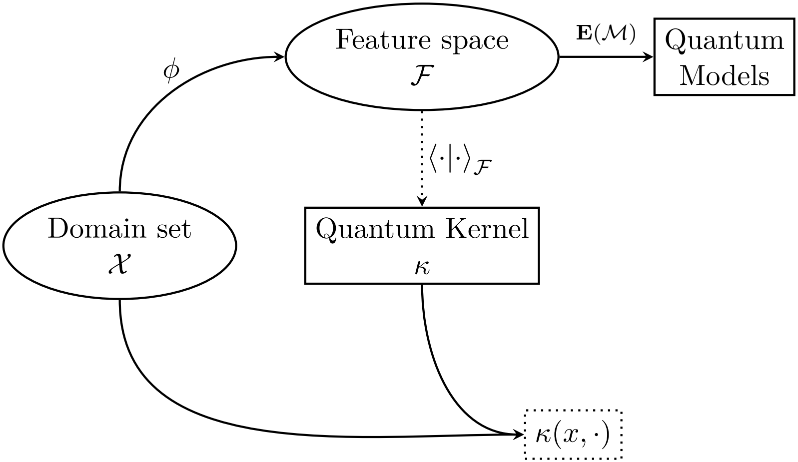
Conceptual summary



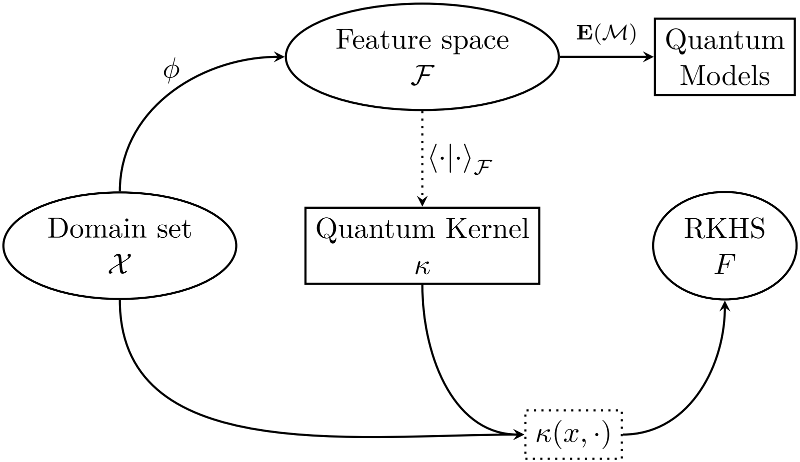
Conceptual summary



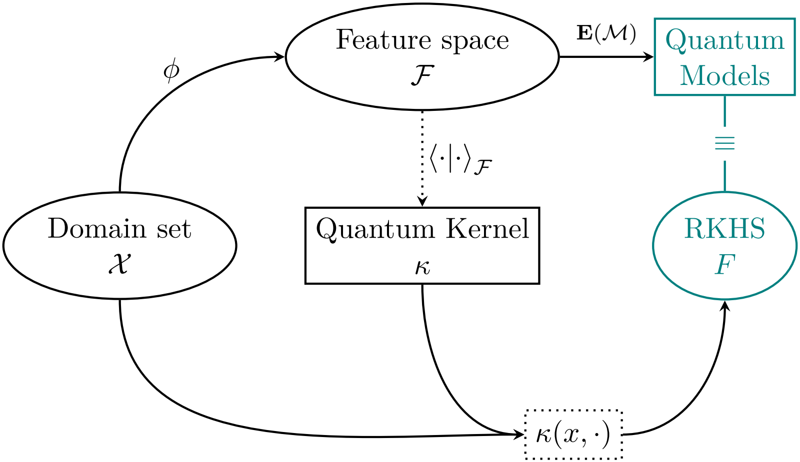
Conceptual summary



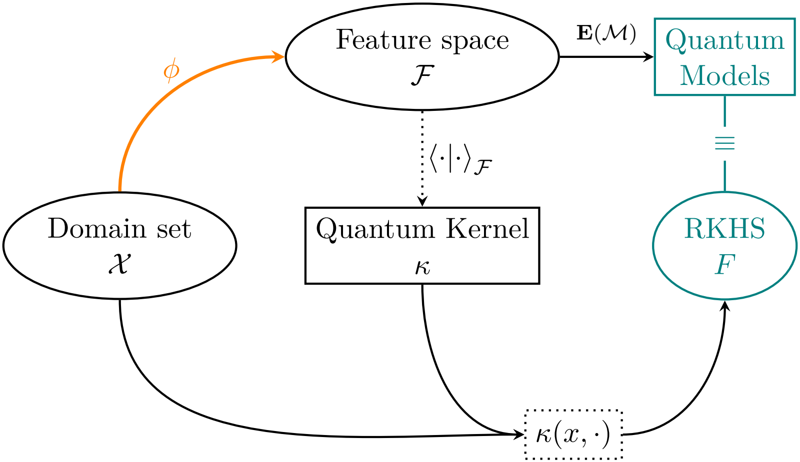
Conceptual summary



Conceptual summary



Conceptual summary



Data-encoding is all you need!

Outline

Assessing a Quantum Model

In order to find an **optimal model**, a metric is required. **cost/risk**.

Assessing a Quantum Model

In order to find an **optimal model**, a metric is required. **cost/risk**.

- **Loss function:** $L : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$

Assessing a Quantum Model

In order to find an **optimal model**, a metric is required. **cost/risk**.

- **Loss function:** $L : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$

This allows to define the **Empirical Risk** of a quantum model f.

- Given the training set $\mathcal{D} = \{(x^1, y^1), \ldots, (x^M, y^M)\}$:

$$\hat{\mathcal{R}}_L(f) = \frac{1}{M} \sum_{i=1}^{M} L(x^i, y^i, f(x^i))$$

- We want to *minimize* this quantity.

Minimizing the Risk

Goal: Minimize the *regularized* empirical risk:

$$\inf_{\mathcal{M}\in\mathcal{F}} \lambda\|\mathcal{M}\|_{\mathcal{F}}^2 + \hat{\mathcal{R}}_L(\mathsf{tr}\left[\rho(x)\mathcal{M}\right]) \tag{1}$$

- $\lambda\|\mathcal{M}\|_{\mathcal{F}}^2$ is the *regularization term*, with $\lambda \in \mathbb{R}^+$.

Minimizing the Risk

Goal: Minimize the *regularized* empirical risk:

$$\inf_{\mathcal{M}\in\mathcal{F}} \lambda\|\mathcal{M}\|_{\mathcal{F}}^2 + \hat{\mathcal{R}}_L(\mathsf{tr}\big[\rho(x)\mathcal{M}\big]) \tag{1}$$

- $\lambda\|\mathcal{M}\|_{\mathcal{F}}^2$ is the *regularization term*, with $\lambda \in \mathbb{R}^+$.

It would be convenient to rewrite Eq. (1) in terms of $f \in F$.

- $\Rightarrow$ This would become an optimization over the RKHS, allowing to exploit the structure just defined.

Minimizing the Risk (cont'd)

$$\lambda \|\mathcal{M}\|_{\mathcal{F}}^2 + \hat{\mathcal{R}}_L(\mathsf{tr}[\rho(x)\mathcal{M}])$$

Minimizing the Risk (cont'd)

$$\lambda\|\mathcal{M}\|_{\mathcal{F}}^2 + \hat{\mathcal{R}}_L(\mathsf{tr}[\rho(x)\mathcal{M}])$$

- **Second term.** Easy, a quantum model is a function $f \in F$ so:

$$\hat{\mathcal{R}}_L(\mathsf{tr}[\rho(x)\mathcal{M}]) = \hat{\mathcal{R}}_L(f)$$

Minimizing the Risk (cont'd)

$$\lambda \|\mathcal{M}\|_{\mathcal{F}}^2 + \hat{\mathcal{R}}_L(\mathsf{tr}[\rho(x)\mathcal{M}])$$

- **Second term.** Easy, a quantum model is a function $f \in F$ so:

$$\hat{\mathcal{R}}_L(\mathsf{tr}[\rho(x)\mathcal{M}]) = \hat{\mathcal{R}}_L(f)$$

- **First term.** Slightly more involved but still:

$$\|\mathcal{M}\|_{\mathcal{F}}^2 = \|f\|_F^2$$

Minimizing the Risk (cont'd)

$$\lambda\|\mathcal{M}\|_{\mathcal{F}}^2 + \hat{\mathcal{R}}_L(\mathsf{tr}\left[\rho(x)\mathcal{M}\right])$$

- **Second term.** Easy, a quantum model is a function $f \in F$ so:

$$\hat{\mathcal{R}}_L(\mathsf{tr}\left[\rho(x)\mathcal{M}\right]) = \hat{\mathcal{R}}_L(f)$$

- **First term.** Slightly more involved but still:

$$\|\mathcal{M}\|_{\mathcal{F}}^2 = \|f\|_F^2$$

- Finally leading to:

$$\inf_{f \in F} \lambda\|f\|_F^2 + \hat{\mathcal{R}}_L(f)$$

Representer Theorem

- The **search** for an optimal quantum model can be done **over the RKHS** F of κ.

Representer Theorem

- The **search** for an optimal quantum model can be done **over the RKHS** F of κ.
- But F is spanned by $[\kappa(x, \cdot)]_{x \in \mathcal{X}}$
  - ...it's **too large** (possibly infinite-dimensional).

Representer Theorem

- The **search** for an optimal quantum model can be done **over the RKHS** F of κ.
- But F is spanned by $[\kappa(x, \cdot)]_{x \in \mathcal{X}}$
  - ...it's **too large** (possibly infinite-dimensional).

Is it possible to **reduce** this search space?

Representer Theorem

- The **search** for an optimal quantum model can be done **over the RKHS** F of κ.
- But F is spanned by $[\kappa(x, \cdot)]_{x \in \mathcal{X}}$
  - ...it's **too large** (possibly infinite-dimensional).

Is it possible to **reduce** this search space?

$\Downarrow$

Representer Theorem

Representer Theorem (cont'd)

Theorem (Representer Theorem)

Given a training set $\mathcal{D} = \{(x^1, y^1), \ldots, (x^M, y^M)\}$, *the optimal model* f_{opt} *such that:*

$$f_{opt} = \inf_{f \in F} \lambda \|f\|_F^2 + \hat{\mathcal{R}}_L(f)$$

can be fully expressed in the span of $[\kappa(x^i, \cdot)]_{1 \leq i \leq M}$*:*

$$f_{opt}(\cdot) = \sum_{i=1}^{M} \alpha_i \kappa(x^i, \cdot)$$

- The optimization problem is at most $\boldsymbol{M}$**-dimensional!**

Finding the optimal model

Consequence:

Finding $f_{\text{opt}}(\cdot) = \sum_{i=1}^{M} \alpha_i \kappa(x^i, \cdot)$ $\equiv$ Finding $\boldsymbol{\alpha} \in \mathbb{R}^M$

- The optimization problem can be conveniently expressed with respect to $\boldsymbol{\alpha}$:

$$\inf_{\boldsymbol{\alpha} \in \mathbb{R}^M} \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \hat{\mathcal{R}}_L(f_{opt})$$

- Supposing L to be a convex function: **Complexity** $\mathcal{O}(M^2)$.
  - $\mathcal{O}(M)$ on quantum computers!

Conclusions

Kernel Methods **extend** to Supervised QML problems!

Conclusions

Kernel Methods **extend** to Supervised QML problems!

...but is it a good idea to use them?

Conclusions

Kernel Methods **extend** to Supervised QML problems!

...but is it a good idea to use them?

Do quantum kernels offer an actual advantage?

Conclusions

Kernel Methods **extend** to Supervised QML problems!

...but is it a good idea to use them?

Do quantum kernels offer an actual advantage?

Two ways to analyze the question:

Conclusions

Kernel Methods **extend** to Supervised QML problems!

...but is it a good idea to use them?

Do quantum kernels offer an actual advantage?

Two ways to analyze the question:

1. Compared to the **classical case**.

Conclusions

Kernel Methods **extend** to Supervised QML problems!

...but is it a good idea to use them?

Do quantum kernels offer an actual advantage?

Two ways to analyze the question:

1. Compared to the **classical case**.
2. Compared to **other QML techniques**.

Conclusions (cont'd)

Quantum Kernels vs. Classical Kernels

Conclusions (cont'd)

Quantum Kernels vs. Classical Kernels

$\Downarrow$

Open problem.

- There are efficient procedures to compute similarity between quantum states (SWAP test) ...
  - ...but **no actual speed-up**.

Conclusions (cont'd)

Quantum Kernels vs. Classical Kernels

$\Downarrow$

Open problem.

- There are efficient procedures to compute similarity between quantum states (SWAP test) ...
  - ...but **no actual speed-up**.
- There exist *classically intractable kernels* that are efficiently computable through quantum algorithms:

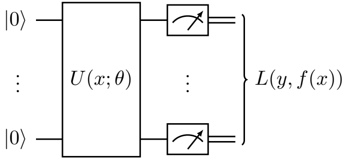
Y. Liu, S. Arunachalam, K. Temme (2021), A rigorous and robust quantum speed-up in supervised machine learning

Conclusions (cont'd)

Quantum Kernels vs. Variational Circuits

Variational Circuits:

- ✔ Training complexity: $\mathcal{O}(|\theta|M)$.
- ✘ Not always find an **optimal model**.
- ✘ Training: Classical.

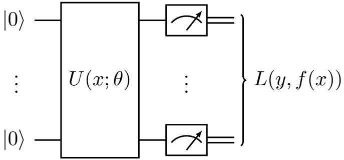


Conclusions (cont'd)

Quantum Kernels vs. Variational Circuits

Variational Circuits:

- ✔ Training complexity: $\mathcal{O}(|\theta|M)$.
- ✘ Not always find an **optimal model**.
- ✘ Training: Classical.



Kernel Methods:

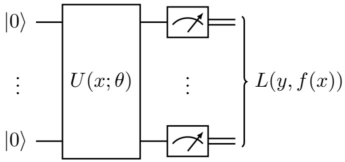
- ✔ Guaranteed to find an optimal model.
- ✔ Training: Possibly Quantum.
- ✔ Training complexity: $\mathcal{O}(M^2)$.
- ✘ **Prediction complexity:** $\mathcal{O}(M^2)$!

Conclusions (cont'd)

Quantum Kernels vs. Variational Circuits

Variational Circuits:

- ✔ Training complexity: $\mathcal{O}(|\theta|M)$.
- ✘ Not always find an **optimal model**.
- ✘ Training: Classical.



Kernel Methods:

- ✔ Guaranteed to find an optimal model.
- ✔ Training: Possibly Quantum.
- ✔ Training complexity: $\mathcal{O}(M^2)$.
- ✘ **Prediction complexity:** $\mathcal{O}(M^2)$!

What's best?

The End

Thank you for your attention.