

Quantum kernel estimation with application to disability insurance

B. Djehiche and B. Löfdahl

KTH Department of Mathematics / SEB

Hands-on lab: Anastasiia Andrievska, RISE

QAS 2025, November 6

- ▶ Disability insurance
- ▶ Kernels and support vector regression
- ▶ Quantum computers
- ▶ Quantum kernel estimation
- ▶ Disability insurance model

- ▶ Health and disability insurance provides economic protection from illness or disability
- ▶ Typically, an insured individual receives a monthly payment from an insurance company in the case of illness
- ▶ The expected cost should be covered by premium payments
- ▶ The insurance company needs to predict future costs using statistical models based on historical data
 - ▶ Typically done by estimating transition probabilities between states such as 'healthy', 'ill', 'dead', ...

- ▶ Consider a population of insured individuals
- ▶ Let E_i be the number of healthy individuals from the population subgroup i
- ▶ We denote by D_i the number of individuals falling ill amongst the E_i insured healthy individuals:

$$D_i \sim \text{Bin}(E_i, p(x_i))$$

- ▶ For each i there is some associated data $x_i \in \mathbb{R}^d$ which may e.g. contain information about age, gender, ...
- ▶ $p(x_i)$ is the probability that an individual randomly selected from E_i falls ill

- ▶ We propose Support Vector Regression (SVR) to model the logistic disability inception probability:

$$\text{logit } p(x) := \log \frac{p(x)}{1 - p(x)} = \sum_{i=1}^n \alpha_i K(x, x_i) + \beta$$

- ▶ K is a quantum kernel estimated on a quantum computer.
- ▶ The parameters $\{\alpha_i\}_i$ and β subsequently fitted using SVR.
- ▶ Hybrid quantum-classical learner!
- ▶ Functional form guarantees $p(x) \in (0, 1)$.

Review: Kernels and support vector regression

- ▶ Let $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1, \dots, n$, be observations in a data set
- ▶ A *feature map* $\Phi : \mathbb{R}^d \mapsto \mathcal{F}$ maps a sample data point x to a feature vector $\Phi(x)$ in a feature space \mathcal{F} (Hilbert space with inner product $\langle \cdot, \cdot \rangle$)
- ▶ Φ naturally gives rise to a *kernel* through the relation

$$K(x, z) = \langle \Phi(x), \Phi(z) \rangle, \quad (1)$$

- ▶ $K(x, z)$ is a *similarity measure* between x and z in the feature space.
- ▶ The *reproducing kernel Hilbert space* associated with Φ is defined by

$$\mathcal{R} = \{f : \mathbb{R}^d \mapsto \mathbb{C}; \quad f(x) = \langle w, \Phi(x) \rangle \quad \forall x \in \mathbb{R}^d, w \in \mathcal{F}\}. \quad (2)$$

- ▶ $f(x) := \langle w, \Phi(x) \rangle$ can be interpreted as linear models in the feature space \mathcal{F} .

Review: Kernels and support vector regression

SVR can be formulated as a convex optimization problem of the form

$$\begin{aligned} \text{P: } \min_{w, b, \xi, \xi'} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi'_i) \\ \text{s.t.} \quad & (w^T \Phi(x_i) + b) - y_i \leq \varepsilon - \xi_i, \quad i = 1, \dots, n, \\ & y_i - (w^T \Phi(x_i) + b) \leq \varepsilon - \xi'_i, \quad i = 1, \dots, n, \\ & \xi_i, \xi'_i \geq 0, \quad i = 1, \dots, n, \end{aligned}$$

where ε determines the error tolerance of the solution, C is a regularization parameter, and $\xi_i \in \mathbb{R}$ and $\xi'_i \in \mathbb{R}$, $i = 1, \dots, n$, are slack variables.

Review: Kernels and support vector regression

The dual formulation D of P is (recall $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$)

$$\begin{aligned} \text{D: } \max_{\lambda, \lambda'} & -\frac{1}{2} \sum_{i,j=1}^n (\lambda_i - \lambda'_i)(\lambda_j - \lambda'_j) K(x_i, x_j) \\ & - \varepsilon \sum_{i=1}^n (\lambda_i - \lambda'_i) + \sum_{i=1}^n y_i (\lambda_i - \lambda'_i) \\ \text{s.t. } & \sum_{i=1}^n (\lambda_i - \lambda'_i) = 0, \\ & 0 \leq \lambda_i \leq C, i = 1, \dots, n, \\ & 0 \leq \lambda'_i \leq C, i = 1, \dots, n, \end{aligned}$$

The solutions of P and D coincide and are given by

$$f(x) = \sum_{i=1}^n \alpha_i K(x, x_i) + \beta, \quad (3)$$

Review: Kernels and support vector regression

- ▶ The feature map (and thus the kernel) can be chosen in many different ways
- ▶ Ideally, the feature map should be chosen such that the kernel can be efficiently computed
- ▶ Well known classical kernels include e.g. the Gaussian kernel:

$$K(x, z) = e^{-\gamma \|x - z\|^2}$$

- ▶ A modern alternative is provided by the class of quantum kernels
 - ▶ Data is mapped to *quantum states* in some *quantum feature (Hilbert) space* \mathcal{H}
 - ▶ Quantum kernels can be estimated using quantum computers!

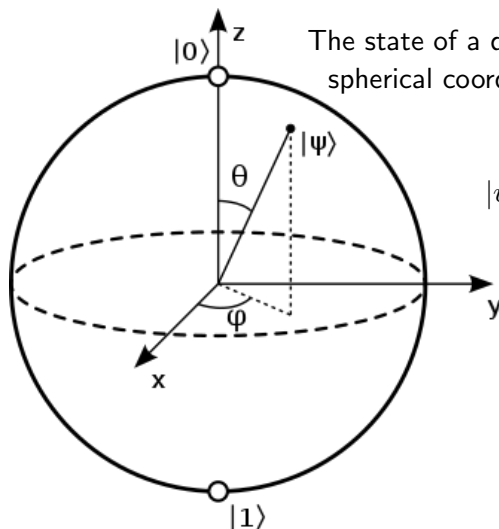
Review: Quantum computers

- ▶ A quantum computer is a computer that is governed by the laws of quantum physics
- ▶ In classical computers, information is represented by bits taking values in $\{0, 1\}$
- ▶ Quantum computers use *qubits*
 - ▶ Information represented by quantum state

$$|\psi\rangle = a|0\rangle + b|1\rangle, \quad |a|^2 + |b|^2 = 1.$$

- ▶ A quantum state induces a probability distribution on $\{0, 1\}$
- ▶ At *measurement* of the quantum state of the qubit, an outcome is determined

Review: Quantum computers



The state of a qubit can be represented using spherical coordinates on the *Bloch sphere*:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$$

Review: Quantum computers

- ▶ Programming a quantum computer with d qubits is performed by creating a *quantum circuit* \mathcal{A}
- ▶ \mathcal{A} induces a probability measure for a r.v. V on $\{0, 1\}^d$
- ▶ Running the circuit \mathcal{A} essentially means sampling from V
- ▶ Intuitively appealing to probabilists, statisticians, actuaries, quants, ...

Review: Quantum kernel estimation

- ▶ Let $\Phi : x \mapsto \Phi(x)$ be a *quantum feature map* that maps a data point to a quantum state in a Hilbert space \mathcal{H}
- ▶ Any quantum state $\psi \in \mathcal{H}$ satisfies the Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} \psi(t, x) = H\psi(t, x), \quad \psi(0, \cdot) \in \mathcal{H} \text{ is given,} \quad (4)$$

where H is the Hamiltonian operator associated to the quantum system.

- ▶ If H is time-independent, the solution to (4) is given by

$$\psi(t, x) = U(t)\psi(0, x), \quad (5)$$

where the operator U defined by

$$U(t) = e^{-iHt/\hbar} \quad (6)$$

is the unitary time evolution operator associated with H .

Review: Quantum kernel estimation

- ▶ For every pair (Φ, x) there is an operator $U_\Phi(x)$ (*feature embedding circuit*), implicitly determined by

$$\Phi(x) = U_\Phi(x)\Omega_0, \quad (7)$$

where Ω_0 denotes the ground state $(|0 \dots 0\rangle)$.

- ▶ Let the kernel K corresponding to Φ be given by

$$K(x, z) = |\langle \Phi(x), \Phi(z) \rangle|^2 = |\Omega_0^\dagger U_\Phi^\dagger(z) U_\Phi(x) \Omega_0|^2 \quad (8)$$

that is, $K(x, z)$ is given by the probability of obtaining the measurement outcome Ω_0 when measuring the quantum state $\Psi(x, z)$ defined by

$$\Psi(x, z) = U_\Phi^\dagger(z) U_\Phi(x) \Omega_0, \quad (9)$$

Review: Quantum kernel estimation

- ▶ The kernel can now be estimated on a quantum computer!
 - ▶ We load the state $\Psi(x, z)$ into a quantum circuit.
 - ▶ This circuit is run n times
 - ▶ $K(x, z)$ is estimated by the frequency of Ω_0 -measurements.
- ▶ The form (8) of the kernel is what allows us to estimate it using a quantum computer! i.e.

$$K(x, z) = |\langle \Phi(x), \Phi(z) \rangle|^2 = |\Omega_0^\dagger U_\Phi^\dagger(z) U_\Phi(x) \Omega_0|^2$$

- ▶ We propose to model the logistic disability inception probability $\text{logit } p(x)$ as

$$\text{logit } p(x) := \log \frac{p(x)}{1 - p(x)} = \sum_{i=1}^n \alpha_i K(x, x_i) + \beta,$$

where K is a quantum kernel (to be defined) that is to be estimated on a quantum computer, and the parameters $\{\alpha_i\}_i$ and β are to be subsequently fitted using SVR.

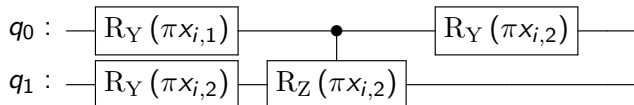
- ▶ Our data: gender $(x_{i,1})$ and age $(x_{i,2})$
- ▶ We choose the kernel K associated with the unitary operator $U_\Phi(\cdot)$ defined by

$$U_\Phi(x_i) = \left(I \otimes R_Y(\pi x_{i,2}) \right) C_{R_Z}(\pi x_{i,2}) \left(R_Y(\pi x_{i,2}) \otimes R_Y(\pi x_{i,1}) \right), \quad (10)$$

- ▶ $R_Y(\cdot)$ denotes a rotation around the Y -axis of the Bloch sphere
- ▶ $C_{R_Z}(\cdot)$ denotes a rotation around the Z -axis for the second qubit, conditional on the state of the first qubit.

Disability model

The unitary operator (10) can be represented by the quantum circuit



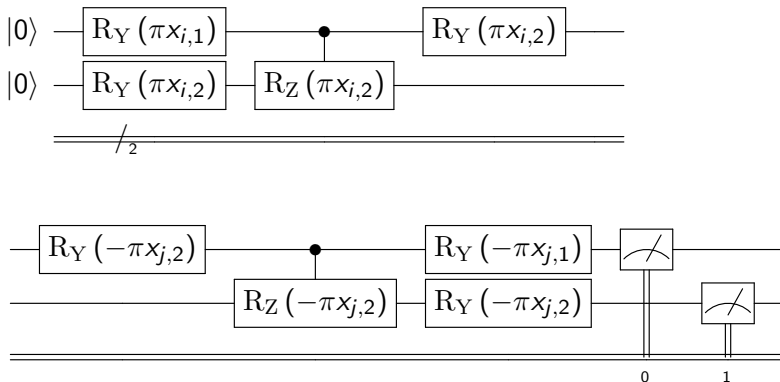
- ▶ $x_{i,1}$ takes the value 1 if the population subgroup is male, and 0 otherwise
- ▶ $x_{i,2}$ is the age of the population subgroup, in centuries.

This circuit is designed to

- ▶ clearly separate male and female subgroups.
- ▶ gradually increase the dissimilarity between different age groups as the difference in ages increases.

Disability model

For each pair (x_i, x_j) , we run this quantum circuit inserting the values of x_i , and then run the adjoint circuit inserting the values of x_j :



Numerical results: kernel

- ▶ We perform simulations on the IBM Yorktown quantum computer
- ▶ For each pair (x_i, x_j) we
 - ▶ run the circuit 8192 times and measure the outcomes
 - ▶ estimate $K(x_i, x_j)$ with the observed frequency of the ground state.
- ▶ Binomial sampling error small ($< 1\%$), hardware error dominates
- ▶ Results are compared with exact (classically determined) kernel

Numerical results: kernel

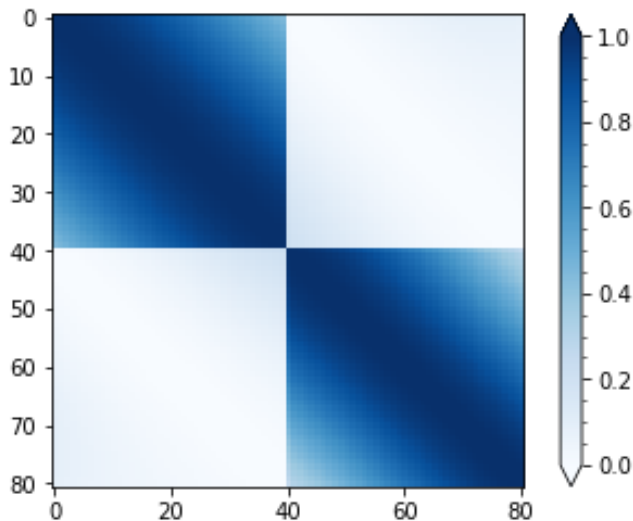


Figure: Classically determined Kernel matrix.

Numerical results: kernel

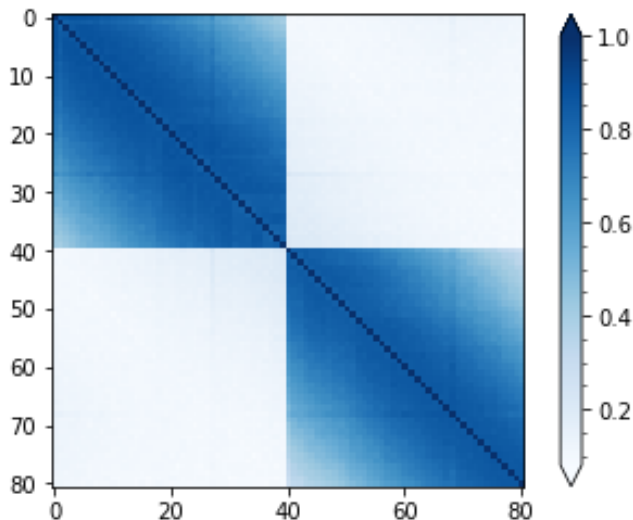


Figure: Kernel matrix estimated on the IBM Yorktown quantum

Numerical results: disability inception

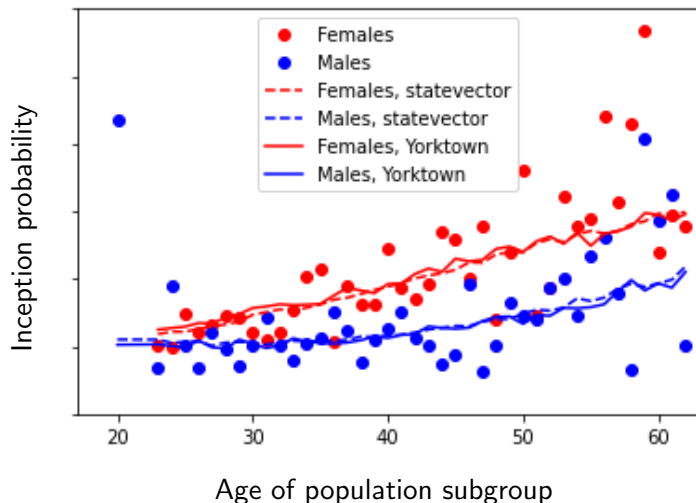


Figure: Out-of-sample disability inception rates estimated by state vector simulation and from the IBM Yorktown quantum computer.

Numerical results: disability inception

Leave-one-out crossvalidation:

Table: Weighted out-of-sample R^2 for the classical and quantum kernels.

kernel	R^2
polynomial	0.550
state vector quantum kernel	0.541
Gaussian kernel	0.529
Yorktown quantum kernel	0.518
sigmoid	0.494
linear	0.426

- ▶ We propose a hybrid classical-quantum approach to estimate disability inception probabilities
- ▶ Suggested model performs similar to existing classical model, even on noisy hardware
- ▶ The approach is not restricted to insurance applications, and can be used for general regression and classification problems, e.g. Credit Risk, Fraud detection, ...
- ▶ Outlook: As the hardware improves and becomes more powerful, this approach might be able to surpass classical models

Selected references



Havlíček et al. 2019.

Supervised learning with quantum-enhanced feature spaces.
Nature 567: 209–12.



Kostaki et al. 2011.

Support vector machines as tools for mortality graduation.
Canadian Studies in Population 38: 37–58.



Rebentrost et al. 2014.

Quantum support vector machine for big data classification.
Physical review letters 113: 130503.



Schölkopf et al. 2000.

New support vector algorithms.
Neural Computation 12: 1207–45.



Schuld and Killoran. 2019.

Quantum machine learning in feature hilbert spaces.
Physical Review Letters 122: 040504.