Machine Learning for Quantum Computing

Shinichi Nakajima
Technische Universität Berlin
BIFOLD
Riken Ceter for AIP

In collaboration with K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, S. Kühn, K.-R. Müller, P. Stornati, P. Kessel, W. Bingting, S. Pedrielli, N. Elosegui, A. Singha

Contents

- Introduction to basic ML techniques
  - Gaussian process (GP)
  - Bayesian optimization (BO)
- Background
  - Variational quantum eigensolvers
  - Parameter shift rule and trigonometric polynomial form
  - Nakanishi-Fujii-Todo (NFT) methods
- Proposed Methods
  - VQE Kernel (generalization of Dirichlet kernel)
  - Expected maximum improvement over confident Regions
  - Experimental results
- Summary

Contents

- Introduction to basic ML techniques
  - Gaussian process (GP)
  - Bayesian optimization (BO)
- Background
  - Variational quantum eigensolvers
  - Parameter shift rule and trigonometric polynomial form
  - Nakanishi-Fujii-Todo (NFT) methods
- Proposed Methods
  - VQE Kernel (generalization of Dirichlet kernel)
  - Expected maximum improvement over confident Regions
  - Experimental results
- Summary

Gaussian process (GP)

Distribution vs. process

probability distribution

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim \mathrm{Gauss}_2\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

Gaussian distribution

Random variable **y** is a point in a finite dimensional space

stochastic process

$$y(\cdot) \sim \mathrm{GP}(\nu(\cdot), k(\cdot,\cdot))$$

Gaussian process

Random variable **y** is a point in a infinite dimensional space

Random variable in function space?

Random variable in function space?

$$y_1 = f(x_1)$$
$$y_2 = f(x_2)$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_2\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$
$$y_2 = f(x_2)$$

$$\boldsymbol{\mu} = \mathbf{0}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_2\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$
$$y_2 = f(x_2)$$

$$\boldsymbol{\mu} = \mathbf{0}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_2\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$
$$y_2 = f(x_2)$$

$$y = f(x)$$

$$\boldsymbol{\mu} = \mathbf{0}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_2\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$
$$y_2 = f(x_2)$$

$$\boldsymbol{\mu} = \mathbf{0}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_3\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$

$$y_2 = f(x_2)$$

$$y_3 = f(x_3)$$

$$y = f(x)$$

$$\boldsymbol{\mu} = \mathbf{0} \in \mathbb{R}^3$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_3\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$

$$y_2 = f(x_2)$$

$$y_3 = f(x_3)$$

$$y = f(x)$$

$$\boldsymbol{\mu} = \mathbf{0} \in \mathbb{R}^3$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.9 \\ 0.4 & 0.9 & 1.0 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_3\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$

$$y_2 = f(x_2)$$

$$y_3 = f(x_3)$$

$$y = f(x)$$

$$\boldsymbol{\mu} = \mathbf{0} \in \mathbb{R}^3$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.9 \\ 0.4 & 0.9 & 1.0 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_3\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$

$$y_2 = f(x_2)$$

$$y_3 = f(x_3)$$

$$y = f(x)$$

$$\boldsymbol{\mu} = \mathbf{0} \in \mathbb{R}^3$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.9 \\ 0.4 & 0.9 & 1.0 \end{pmatrix}$$

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_3\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$

$$y_2 = f(x_2)$$

$$y_3 = f(x_3)$$

smoothness!

$$\boldsymbol{\mu} = \mathbf{0} \in \mathbb{R}^3$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.9 \\ 0.4 & 0.9 & 1.0 \end{pmatrix}$$

$$\Sigma_{i,j} = \exp\left(-\frac{|x_i - x_j|^2}{\gamma^2}\right)$$

Gaussian kernel

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_K\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$y_1 = f(x_1)$$

$$\vdots$$

$$y_K = f(x_K)$$

$$y = f(x)$$

$$\boldsymbol{\mu} \in \mathbb{R}^K$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{1,1} & \cdots & \Sigma_{1,K} \\ \vdots & \ddots & \vdots \\ \Sigma_{K,1} & \cdots & \Sigma_{K,K} \end{pmatrix}$$

Uniform grid in domain

Random variable in function space?

$$p(\boldsymbol{y}) = \mathcal{N}_\infty(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$y_1 = f(x_1)$$

$$\vdots$$

$$y_\infty = f(x_\infty)$$

$$y = f(x)$$

$$\boldsymbol{\mu} \in \mathbb{R}^\infty$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{1,1} & \cdots & \Sigma_{1,\infty} \\ \vdots & \ddots & \vdots \\ \Sigma_{\infty,1} & \cdots & \Sigma_{\infty,\infty} \end{pmatrix}$$

infinitesimal grid → Gaussian process!

Probability distribution to stochastic process

<u>Gaussian distribution</u>

$$p(\boldsymbol{y}) = \mathcal{N}_{100000}\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$\boldsymbol{y} \in \mathbb{R}^{100000}$, random vector

$\boldsymbol{\mu} \in \mathbb{R}^{100000}$, mean vector

$\boldsymbol{\Sigma} \in \mathbb{R}^{100000 \times 100000}$,

covariance matrix

Distribution: probability defined in a finite dimensional space.

Probability distribution to stochastic process

Gaussian distribution

$$p(\boldsymbol{y}) = \mathcal{N}_{100000}\left(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$\boldsymbol{y} \in \mathbb{R}^{100000}$, random vector

$\boldsymbol{\mu} \in \mathbb{R}^{100000}$, mean vector

$\boldsymbol{\Sigma} \in \mathbb{R}^{100000 \times 100000}$, covariance matrix

Gaussian process

$$p(y(\cdot)) = \mathrm{GP}\left(y; \nu(\cdot), k(\cdot, \cdot)\right)$$

$y(x) \in \mathbb{R}$ for $x \in \mathbb{R}$, random function

$\nu(x) \in \mathbb{R}$ for $x \in \mathbb{R}$, mean function

$k(x, x') \in \mathbb{R}$ for $(x, x') \in \mathbb{R}^2$, kernel (covariance) function

Distribution: probability defined in a finite dimensional space.

Process: probability defined in an infinite dimensional space (e.g., function).

Samples from Gaussian process

$$f(\cdot) \sim \mathrm{GP}(\nu(\cdot), k(\cdot,\cdot)), \quad \text{where} \quad \nu(\boldsymbol{x}) = 0(\boldsymbol{x}) \equiv 0, \quad k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\gamma^2}\right).$$

$\gamma^2 = 0.1$

$\gamma^2 = 1$

$\gamma^2 = 10$

GP regression

GP regression

Model distribution (likelihood): $p(y|\boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\gamma^2}\right).$

GP regression

Unknown parameter (to be estimated)

Model distribution (likelihood): $p(y|\boldsymbol{x}, \overbrace{f(\cdot)}) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\gamma^2}\right).$

GP regression

Unknown parameter (to be estimated)

Model distribution (likelihood): $p(y|\boldsymbol{x}, \overbrace{f(\cdot)}) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\gamma^2}\right).$

Given a training set $\{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1}^N$

GP regression

Unknown parameter (to be estimated)

Model distribution (likelihood): $p(y|\boldsymbol{x}, \overbrace{f(\cdot)}) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\gamma^2}\right).$

Given a training set $\{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1}^N$

$$\boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} \qquad \boldsymbol{f} = \begin{pmatrix} f(\boldsymbol{x}^{(1)}) \\ \vdots \\ f(\boldsymbol{x}^{(N)}) \end{pmatrix}$$

Posterior: $p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}_N(\boldsymbol{f}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$

where $\boldsymbol{\mu}_f = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{y},$

$\boldsymbol{\Sigma}_f = \left(\boldsymbol{K}^{-1} + \sigma^{-2}\boldsymbol{I}_N\right)^{-1}.$

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}) \end{pmatrix}.$$

GP regression

Unknown parameter (to be estimated)

Model distribution (likelihood): $p(y|\boldsymbol{x}, \overbrace{f(\cdot)}) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\gamma^2}\right).$

Given a training set $\{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1}^N$

$$\boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} \qquad \boldsymbol{f} = \begin{pmatrix} f(\boldsymbol{x}^{(1)}) \\ \vdots \\ f(\boldsymbol{x}^{(N)}) \end{pmatrix}$$

Posterior: $p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}_N(\boldsymbol{f}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$

where $\boldsymbol{\mu}_f = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{y},$

$\boldsymbol{\Sigma}_f = \left(\boldsymbol{K}^{-1} + \sigma^{-2}\boldsymbol{I}_N\right)^{-1}.$

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}) \end{pmatrix}.$$

For test points $\{\boldsymbol{x}^{*(n)}\}_{n=1}^{N^*}$

GP regression

Unknown parameter (to be estimated)

Model distribution (likelihood): $p(y|\boldsymbol{x}, \overbrace{f(\cdot)}) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\gamma^2}\right).$

Given a training set $\{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1}^N$

$$\boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} \qquad \boldsymbol{f} = \begin{pmatrix} f(\boldsymbol{x}^{(1)}) \\ \vdots \\ f(\boldsymbol{x}^{(N)}) \end{pmatrix}$$

Posterior: $p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}_N(\boldsymbol{f}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$

where $\boldsymbol{\mu}_f = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{y},$

$\boldsymbol{\Sigma}_f = \left(\boldsymbol{K}^{-1} + \sigma^{-2}\boldsymbol{I}_N\right)^{-1}.$

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}) \end{pmatrix}.$$

For test points $\{\boldsymbol{x}^{*(n)}\}_{n=1}^{N^*}$

$$\boldsymbol{f}^* = \begin{pmatrix} f(\boldsymbol{x}^{*(1)}) \\ \vdots \\ f(\boldsymbol{x}^{*(N)}) \end{pmatrix}$$

Posterior: $p(\boldsymbol{f}^*|\boldsymbol{X}^*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}_{N^*}(\boldsymbol{f}^*; \boldsymbol{\mu}_{f^*}, \boldsymbol{\Sigma}_{f^*})$

where $\boldsymbol{\mu}_{f^*} = \boldsymbol{K}^{*\top}\left(\boldsymbol{K} + \sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{y},$

$\boldsymbol{\Sigma}_{f^*} = \boldsymbol{K}^{**} - \boldsymbol{K}^{*\top}\left(\boldsymbol{K} + \sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{K}^*.$

$$\boldsymbol{K}^* = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{*(M)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{*(M)}) \end{pmatrix}.$$

$$\boldsymbol{K}^{**} = \begin{pmatrix} k(\boldsymbol{x}^{*(1)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{*(1)}, \boldsymbol{x}^{*(M)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{*(M)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{*(M)}, \boldsymbol{x}^{*(M)}) \end{pmatrix}.$$

GP regression

Unknown parameter (to be estimated)

Model distribution (likelihood): $p(y|\boldsymbol{x}, \overbrace{f(\cdot)}) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$

Prior distribution: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with, e.g., $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\gamma^2}\right).$

Given a training set $\{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1}^N$

$$\boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} \qquad \boldsymbol{f} = \begin{pmatrix} f(\boldsymbol{x}^{(1)}) \\ \vdots \\ f(\boldsymbol{x}^{(N)}) \end{pmatrix}$$

Posterior: $p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}_N(\boldsymbol{f}; \boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$

where $\boldsymbol{\mu}_f = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{y},$

$\boldsymbol{\Sigma}_f = \left(\boldsymbol{K}^{-1} + \sigma^{-2}\boldsymbol{I}_N\right)^{-1}.$

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}) \end{pmatrix}.$$

For test points $\{\boldsymbol{x}^{*(n)}\}_{n=1}^{N^*}$

$$\boldsymbol{f}^* = \begin{pmatrix} f(\boldsymbol{x}^{*(1)}) \\ \vdots \\ f(\boldsymbol{x}^{*(N)}) \end{pmatrix}$$

Posterior: $p(\boldsymbol{f}^*|\boldsymbol{X}^*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}_{N^*}(\boldsymbol{f}^*; \boldsymbol{\mu}_{f^*}, \boldsymbol{\Sigma}_{f^*})$

where $\boldsymbol{\mu}_{f^*} = \boldsymbol{K}^{*\top}\left(\boldsymbol{K} + \sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{y},$

$\boldsymbol{\Sigma}_{f^*} = \boldsymbol{K}^{**} - \boldsymbol{K}^{*\top}\left(\boldsymbol{K} + \sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{K}^*.$

$$\boldsymbol{K}^* = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{*(M)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{*(M)}) \end{pmatrix}.$$

$$\boldsymbol{K}^{**} = \begin{pmatrix} k(\boldsymbol{x}^{*(1)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{*(1)}, \boldsymbol{x}^{*(M)}) \\ \vdots & & \vdots \\ k(\boldsymbol{x}^{*(M)}, \boldsymbol{x}^{*(1)}) & \cdots & k(\boldsymbol{x}^{*(M)}, \boldsymbol{x}^{*(M)}) \end{pmatrix}.$$

Posterior is GP! $p(f(\cdot)|\boldsymbol{y}) = \mathrm{GP}(f(\cdot); \mu_f(\cdot), s_f(\cdot,\cdot))$

GP regression

Prior: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\gamma^2}\right).$

GP regression

Prior: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\gamma^2}\right).$

Likelihood: $p(\boldsymbol{y}|\boldsymbol{X}, f(\cdot)) = \sum_{n=1}^{N} \mathcal{N}_1(y_n; f(\boldsymbol{x}_n), \sigma^2)$

observation!

GP regression

Prior: $p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$ with $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{\gamma^2}\right).$

Likelihood: $p(\boldsymbol{y}|\boldsymbol{X}, f(\cdot)) = \sum_{n=1}^{N} \mathcal{N}_1(y_n; f(\boldsymbol{x}_n), \sigma^2)$

Posterior:
$$\begin{aligned} p(f(\cdot)|\boldsymbol{X}, \boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{X}, f(\cdot))p(f(\cdot)) \\ &= \mathrm{GP}(f(\cdot); \mu_f(\cdot), s_f(\cdot)) \end{aligned}$$

observation!

or

Equivalent/related methods

GP regression:

$$p(y|\boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$
$$p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$$

If $\exists \boldsymbol{\Phi}(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^H$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$

Equivalent/related methods

GP regression:

$$p(y|\boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$
$$p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$$

If $\exists \boldsymbol{\Phi}(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^H$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$

(Bayesian) linear regression:

$$p(y|\boldsymbol{x}, \boldsymbol{b}) = \mathcal{N}_D(y; f(\boldsymbol{x}; \boldsymbol{b}), \sigma^2),$$
$$f(\boldsymbol{x}; \boldsymbol{b}) = \boldsymbol{b}^\top \boldsymbol{\phi}(\boldsymbol{x}),$$
$$p(\boldsymbol{b}) = \mathcal{N}_H(\boldsymbol{b}; \boldsymbol{0}, \boldsymbol{I}_H),$$

Equivalent/related methods

GP regression: Dual

$$p(y|\boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$
$$p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$$

If $\exists \boldsymbol{\Phi}(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^H$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$

Equivalent

(Bayesian) linear regression: Primal

$$p(y|\boldsymbol{x}, \boldsymbol{b}) = \mathcal{N}_D(y; f(\boldsymbol{x}; \boldsymbol{b}), \sigma^2),$$
$$f(\boldsymbol{x}; \boldsymbol{b}) = \boldsymbol{b}^\top \boldsymbol{\phi}(\boldsymbol{x}),$$
$$p(\boldsymbol{b}) = \mathcal{N}_H(\boldsymbol{b}; \boldsymbol{0}, \boldsymbol{I}_H),$$

Equivalent/related methods

GP regression: Dual

$$p(y|\boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$
$$p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$$

Kernel ridge regression:

$$\boldsymbol{\mu}_f = \boldsymbol{K}^{*\top}\left(\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N\right)^{-1} \boldsymbol{y},$$

If $\exists \boldsymbol{\Phi}(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^H$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$

Equivalent

(Bayesian) linear regression: Primal

$$p(y|\boldsymbol{x}, \boldsymbol{b}) = \mathcal{N}_D(y; f(\boldsymbol{x}; \boldsymbol{b}), \sigma^2),$$
$$f(\boldsymbol{x}; \boldsymbol{b}) = \boldsymbol{b}^\top \boldsymbol{\phi}(\boldsymbol{x}),$$
$$p(\boldsymbol{b}) = \mathcal{N}_H(\boldsymbol{b}; \boldsymbol{0}, \boldsymbol{I}_H),$$

Ridge regression:

$$\widehat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \left(\|\boldsymbol{y} - \boldsymbol{\Phi}^\top \boldsymbol{w}\|^2 + \sigma^2 \|\boldsymbol{w}\|^2 \right)$$
$$= \left(\boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}_M \right)^{-1} \boldsymbol{X}\boldsymbol{y}$$

Equivalent/related methods

Maximum a posteriori (MAP) approximation

GP regression: Dual → Kernel ridge regression:

$$p(y|\boldsymbol{x}, f(\cdot)) = \mathcal{N}_1(y; f(\boldsymbol{x}), \sigma^2)$$
$$p(f(\cdot)) = \mathrm{GP}(f(\cdot); 0(\cdot), k(\cdot,\cdot))$$

$$\boldsymbol{\mu}_f = \boldsymbol{K}^{*\top}\left(\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N\right)^{-1} \boldsymbol{y},$$

If $\exists \boldsymbol{\Phi}(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^H$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$

Equivalent

Maximum a posteriori (MAP) approximation

(Bayesian) linear regression: Primal → Ridge regression:

$$p(y|\boldsymbol{x}, \boldsymbol{b}) = \mathcal{N}_D(y; f(\boldsymbol{x}; \boldsymbol{b}), \sigma^2),$$
$$f(\boldsymbol{x}; \boldsymbol{b}) = \boldsymbol{b}^\top \boldsymbol{\phi}(\boldsymbol{x}),$$
$$p(\boldsymbol{b}) = \mathcal{N}_H(\boldsymbol{b}; \boldsymbol{0}, \boldsymbol{I}_H),$$

$$\widehat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \left(\|\boldsymbol{y} - \boldsymbol{\Phi}^\top \boldsymbol{w}\|^2 + \sigma^2 \|\boldsymbol{w}\|^2 \right)$$
$$= \left(\boldsymbol{X}\boldsymbol{X}^\top + \sigma^2 \boldsymbol{I}_M \right)^{-1} \boldsymbol{X}\boldsymbol{y}$$

Text book!

Gaussian Processes for Machine Learning (MIT Press)
Carl Edward Rasmussen and Christopher K. I. Williams

Chapters 1 and 2

Bayesian optimization

Black-box function optimization

$$\min \; f(x)$$

Black-box function optimization

$$\min \; f(x)$$

$\widehat{f}(\boldsymbol{x})$: surrogate function

- (Deterministic) regression allows *exploitation*.

Black-box function optimization

$$\min \; f(x)$$

$\widehat{f}(\boldsymbol{x}) + \widehat{\sigma}(\boldsymbol{x})$

$\widehat{f}(\boldsymbol{x})$: surrogate function

$\widehat{f}(\boldsymbol{x}) - \widehat{\sigma}(\boldsymbol{x})$

- (Deterministic) regression allows *exploitation*.

- Predictive distribution provides uncertainty,

Black-box function optimization

$$\min \; f(x)$$

$\widehat{f}(\boldsymbol{x}) + \widehat{\sigma}(\boldsymbol{x})$

$\widehat{f}(\boldsymbol{x})$: surrogate function

$\widehat{f}(\boldsymbol{x}) - \widehat{\sigma}(\boldsymbol{x})$

- (Deterministic) regression allows *exploitation*.

- Predictive distribution provides uncertainty, which allows reasonable *exploration*!

Black-box function optimization

$$\min \; f(x)$$

$\widehat{f}(\boldsymbol{x}) + \widehat{\sigma}(\boldsymbol{x})$

$\widehat{f}(\boldsymbol{x})$: surrogate function

$\widehat{f}(\boldsymbol{x}) - \widehat{\sigma}(\boldsymbol{x})$

- (Deterministic) regression allows *exploitation*.

- Predictive distribution provides uncertainty, which allows reasonable *exploration*!

<u>Exploration-exploitation trade-off</u>

- hypeparameter (architecture) optimization
- Reinforcement learning

Acquisition function

Next observation point is chosen by maximization acquisition function:

$$\widetilde{\boldsymbol{x}} = \underset{x'}{\operatorname{argmax}}\, a_{\mathcal{D}}(\boldsymbol{x}')$$

$a_{\mathcal{D}}(\boldsymbol{x}')$: Acquisition function based on observed data $\mathcal{D}$ until now

Acquisition function

Next observation point is chosen by maximization acquisition function:

$$\widetilde{\boldsymbol{x}} = \underset{x'}{\operatorname{argmax}}\, a_{\mathcal{D}}(\boldsymbol{x}')$$

$a_{\mathcal{D}}(\boldsymbol{x}')$: Acquisition function based on observed data $\mathcal{D}$ until now

- Lower confidence bound (LCB)

$$a_{\mathcal{D}}^{\mathrm{UCB}}(\boldsymbol{x}') = -\left(\mu_{\mathcal{D}}(\boldsymbol{x}') - \beta\sigma_{\mathcal{D}}(\boldsymbol{x}')\right)$$

where $\mu_{\mathcal{D}}(\boldsymbol{x}'), \sigma^2_{\mathcal{D}}(\boldsymbol{x}')$ are GP mean and variance at $\boldsymbol{x}'$

Acquisition function

Next observation point is chosen by maximization acquisition function:

$$\widetilde{\boldsymbol{x}} = \underset{x'}{\operatorname{argmax}}\, a_{\mathcal{D}}(\boldsymbol{x}')$$

$a_{\mathcal{D}}(\boldsymbol{x}')$: Acquisition function based on observed data $\mathcal{D}$ until now

- Lower confidence bound (LCB)

$$a_{\mathcal{D}}^{\mathrm{UCB}}(\boldsymbol{x}') = -\left(\mu_{\mathcal{D}}(\boldsymbol{x}') - \beta\sigma_{\mathcal{D}}(\boldsymbol{x}')\right)$$

where $\mu_{\mathcal{D}}(\boldsymbol{x}'), \sigma^2_{\mathcal{D}}(\boldsymbol{x}')$ are GP mean and variance at $\boldsymbol{x}'$

- Probability of improvement (PI)

$$a_{\mathcal{D}}^{\mathrm{PI}}(\boldsymbol{x}') = \left\langle \mathbf{1}(f^{\dagger} > f') \right\rangle_{p(f'|\boldsymbol{x}',\mathcal{D})},$$

where $f^{\dagger} = \min(\boldsymbol{f}) = \min_{n\in\{1,\ldots,N\}}(f_n)$ is the current best score.

Acquisition function

Next observation point is chosen by maximization acquisition function:

$$\widetilde{\boldsymbol{x}} = \underset{x'}{\operatorname{argmax}}\, a_{\mathcal{D}}(\boldsymbol{x}')$$

$a_{\mathcal{D}}(\boldsymbol{x}')$: Acquisition function based on observed data $\mathcal{D}$ until now

- Lower confidence bound (LCB)

$$a_{\mathcal{D}}^{\mathrm{UCB}}(\boldsymbol{x}') = -\left(\mu_{\mathcal{D}}(\boldsymbol{x}') - \beta\sigma_{\mathcal{D}}(\boldsymbol{x}')\right)$$

where $\mu_{\mathcal{D}}(\boldsymbol{x}'), \sigma^2_{\mathcal{D}}(\boldsymbol{x}')$ are GP mean and variance at $\boldsymbol{x}'$

- Probability of improvement (PI)

$$a_{\mathcal{D}}^{\mathrm{PI}}(\boldsymbol{x}') = \left\langle \mathbf{1}(f^{\dagger} > f') \right\rangle_{p(f'|\boldsymbol{x}',\mathcal{D})},$$

where $f^{\dagger} = \min(\boldsymbol{f}) = \min_{n\in\{1,\ldots,N\}}(f_n)$ is the current best score.

- Expected improvement (EI)

$$a_{\mathcal{D}}^{\mathrm{EI}}(\boldsymbol{x}') = \left\langle \max(0, f^{\dagger} - f') \right\rangle_{p(f'|\boldsymbol{x}',\mathcal{D})},$$

Contents

- Introduction to basic ML techniques
  - Gaussian process (GP)
  - Bayesian optimization (BO)
- Background
  - Variational quantum eigensolvers
  - Parameter shift rule and trigonometric polynomial form
  - Nakanishi-Fujii-Todo (NFT) methods
- Proposed Methods
  - VQE Kernel (generalization of Dirichlet kernel)
  - Expected maximum improvement over confident Regions
  - Experimental results
- Summary

Variational quantum eigensolvers (VQE)

Variational quantum eigensolvers

Goal:

Given a Hamiltonian, e.g., $H = -\sum_{j=1}^{Q-1} J_X \sigma_j^X \sigma_{j+1}^X - \sum_{j=1}^{Q} h_Z \sigma_j^Z$

find the ground-state energy $f_{\mathrm{GS}}^* = \langle \psi_{\mathrm{GS}} | H | \psi_{\mathrm{GS}} \rangle$

Variational quantum eigensolvers

Goal:

Given a Hamiltonian, e.g., $H = -\sum_{j=1}^{Q-1} J_X \sigma_j^X \sigma_{j+1}^X - \sum_{j=1}^{Q} h_Z \sigma_j^Z$

find the ground-state energy $f_{\mathrm{GS}}^* = \langle \psi_{\mathrm{GS}} | H | \psi_{\mathrm{GS}} \rangle$

With a quantum circuit $|\psi_{\boldsymbol{x}}\rangle = G(\boldsymbol{x})|\psi_0\rangle$ with controllable parameter $\boldsymbol{x} \in [0, 2\pi)^D$

solve $\underbrace{\min_{\boldsymbol{x}} f^*(\boldsymbol{x})}_{\text{classical computing}}$ where $f^*(\boldsymbol{x}) = \underbrace{\langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle}_{\text{quantum computing}}$

Variational quantum eigensolvers

Goal:

Given a Hamiltonian, e.g., $H = -\sum_{j=1}^{Q-1} J_X \sigma_j^X \sigma_{j+1}^X - \sum_{j=1}^{Q} h_Z \sigma_j^Z$

find the ground-state energy $f_{\mathrm{GS}}^* = \langle \psi_{\mathrm{GS}} | H | \psi_{\mathrm{GS}} \rangle$

With a quantum circuit $|\psi_{\boldsymbol{x}}\rangle = G(\boldsymbol{x})|\psi_0\rangle$ with controllable parameter $\boldsymbol{x} \in [0, 2\pi)^D$

solve $\underbrace{\min_{\boldsymbol{x}} f^*(\boldsymbol{x})}_{\text{classical computing}}$ where $f^*(\boldsymbol{x}) = \underbrace{\langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle}_{\text{quantum computing}}$

<u>Provides an upper-bound of the ground-state energy</u>

Quantum circuit

Quantum circuit

$$G(\boldsymbol{x})$$

$$G(\boldsymbol{x}) = \cdots \circ \underbrace{G_9(x_6) \circ G_8(x_5) \circ G_7(x_4)}_{\text{parameteric}} \cdots \underbrace{G_6 \circ G_5 \circ G_4}_{\text{non-parameteric}} \circ \underbrace{G_3(x_3) \circ G_2(x_2) \circ G_1(x_1)}_{\text{parameteric}}$$

Quantum circuit

$G(\boldsymbol{x})$

$$G(\boldsymbol{x}) = \cdots \circ \underbrace{G_9(x_6) \circ G_8(x_5) \circ G_7(x_4)}_{\text{parameteric}} \cdots \underbrace{G_6 \circ G_5 \circ G_4}_{\text{non-parameteric}} \circ \underbrace{G_3(x_3) \circ G_2(x_2) \circ G_1(x_1)}_{\text{parameteric}}$$

Each parametric gate is unitary

$G_{d'}(x) = U_{d'}(x) = \exp\left(-ixP_{d'}/2\right)$ with Pauli operators $P_{d'} \in \{\sigma_q^X, \sigma_q^Y, \sigma_q^Z\}_{q=1}^{Q}$

This restricts the function form of objective…

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad\qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Parameter shift rule:

Proposition *(Mitarai et al., 2018) The VQE objective function $f^*(\cdot)$ for any parametric quantum circuit $G(\cdot)$, Hermitian operator H, and initial state $|\psi_0\rangle$ satisfies*

$$2\frac{\partial}{\partial x_d} f^*(\boldsymbol{x}) = f^*\left(\boldsymbol{x} + \tfrac{\pi}{2}\boldsymbol{e}_d\right) - f^*\left(\boldsymbol{x} - \tfrac{\pi}{2}\boldsymbol{e}_d\right), \quad \forall \boldsymbol{x} \in [0, 2\pi)^D, d = 1, \ldots, D.$$

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad\qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Parameter shift rule:

Proposition *(Mitarai et al., 2018) The VQE objective function $f^*(\cdot)$ for any parametric quantum circuit $G(\cdot)$, Hermitian operator H, and initial state $|\psi_0\rangle$ satisfies*

$$2\frac{\partial}{\partial x_d} f^*(\boldsymbol{x}) = f^*\left(\boldsymbol{x} + \tfrac{\pi}{2}\boldsymbol{e}_d\right) - f^*\left(\boldsymbol{x} - \tfrac{\pi}{2}\boldsymbol{e}_d\right), \quad \forall \boldsymbol{x} \in [0, 2\pi)^D, d = 1, \ldots, D.$$

- Gradient estimate
- Differential equation

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad\qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Parameter shift rule:

Proposition *(Mitarai et al., 2018) The VQE objective function $f^*(\cdot)$ for any parametric quantum circuit $G(\cdot)$, Hermitian operator H, and initial state $|\psi_0\rangle$ satisfies*

$$2\frac{\partial}{\partial x_d} f^*(\boldsymbol{x}) = f^*\left(\boldsymbol{x} + \tfrac{\pi}{2}\boldsymbol{e}_d\right) - f^*\left(\boldsymbol{x} - \tfrac{\pi}{2}\boldsymbol{e}_d\right), \quad \forall \boldsymbol{x} \in [0, 2\pi)^D, d = 1, \ldots, D.$$

Trigonometric polynomial form:

Proposition *(Nakanishi et al. 2020) For the VQE objective function $f^*(\cdot)$ with any $G(\cdot), H$, and $|\psi_0\rangle$,*

$$\exists \boldsymbol{b} \in \mathbb{R}^{3^D} \quad \textit{such that} \quad f^*(\boldsymbol{x}) = \boldsymbol{b}^\top \cdot \mathbf{vec}\left(\otimes_{d=1}^{D} (1, \cos x_d, \sin x_d)^\top\right), \quad \forall \boldsymbol{x} \in [0, 2\pi)^D,$$

where $\otimes$ and $\mathbf{vec}(\cdot)$ denote the tensor product and the vectorization operator for a tensor, respectively.

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad\qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Parameter shift rule:

Proposition *(Mitarai et al., 2018) The VQE objective function $f^*(\cdot)$ for any parametric quantum circuit $G(\cdot)$, Hermitian operator H, and initial state $|\psi_0\rangle$ satisfies*

$$2\frac{\partial}{\partial x_d} f^*(\boldsymbol{x}) = f^*\left(\boldsymbol{x} + \tfrac{\pi}{2}\boldsymbol{e}_d\right) - f^*\left(\boldsymbol{x} - \tfrac{\pi}{2}\boldsymbol{e}_d\right), \quad \forall \boldsymbol{x} \in [0, 2\pi)^D, d = 1, \ldots, D.$$

Trigonometric polynomial form:

Proposition *(Nakanishi et al. 2020) For the VQE objective function $f^*(\cdot)$ with any $G(\cdot), H$, and $|\psi_0\rangle$,*

$$\exists \boldsymbol{b} \in \mathbb{R}^{3^D} \quad \textit{such that} \quad f^*(\boldsymbol{x}) = \boldsymbol{b}^\top \cdot \mathbf{vec}\left(\otimes_{d=1}^{D} (1, \cos x_d, \sin x_d)^\top\right), \quad \forall \boldsymbol{x} \in [0, 2\pi)^D,$$

where $\otimes$ and $\mathbf{vec}(\cdot)$ denote the tensor product and the vectorization operator for a tensor, respectively.

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad\qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

For each direction of $\boldsymbol{x} = (x_1, \ldots, x_D)$

$$f^*(x_d; \boldsymbol{x}_{d' \neq d}) = b_0 + b_1 \cos x_d + b_2 \sin x_d$$

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

For each direction of $\boldsymbol{x} = (x_1, \ldots, x_D)$

$$f^*(x_d; \boldsymbol{x}_{d' \neq d}) = b_0 + b_1 \cos x_d + b_2 \sin x_d$$

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

For each direction of $\boldsymbol{x} = (x_1, \ldots, x_D)$

$$f^*(x_d; \boldsymbol{x}_{d' \neq d}) = b_0 + b_1 \cos x_d + b_2 \sin x_d$$

Physical knowledge on objective function form

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

For each direction of $\boldsymbol{x} = (x_1, \ldots, x_D)$

$$f^*(x_d; \boldsymbol{x}_{d' \neq d}) = b_0 + b_1 \cos x_d + b_2 \sin x_d$$

Nakanishi-Fujii-Todo method

Strong state-of-the-art method performing sequential minimal optimization (SMO) that requires only 3 observations in each iteration.

Nakanishi-Fujii-Todo method

Strong state-of-the-art method performing sequential minimal optimization (SMO) that requires only 3 observations in each iteration.

Nakanishi-Fujii-Todo method

Strong state-of-the-art method performing sequential minimal optimization (SMO) that requires only 3 observations in each iteration.

Nakanishi-Fujii-Todo method

Strong state-of-the-art method performing sequential minimal optimization (SMO) that requires only 3 observations in each iteration.

Nakanishi-Fujii-Todo method

Strong state-of-the-art method performing sequential minimal optimization (SMO) that requires only 3 observations in each iteration.

Nakanishi-Fujii-Todo method

Strong state-of-the-art method performing sequential minimal optimization (SMO) that requires only 3 observations in each iteration.

Very efficient method, but …

Issue: Observation noise should be considered

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Issue: Observation noise should be considered

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

Issue: Observation noise should be considered

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

We can only observe noisy objective: $y = f^*(\boldsymbol{x}) + \varepsilon$

Issue: Observation noise should be considered

$$\min_{\boldsymbol{x}} f^*(\boldsymbol{x}) \qquad f^*(\boldsymbol{x}) = \langle \psi_0 | G(\boldsymbol{x})^\dagger H G(\boldsymbol{x}) | \psi_0 \rangle$$

We can only observe noisy objective: $y = f^*(\boldsymbol{x}) + \varepsilon$

Optimization with uncertainty taken into account?

Bayesian optimization!

Contents

- Introduction to basic ML techniques
  - Gaussian process (GP)
  - Bayesian optimization (BO)
- Background
  - Variational quantum eigensolvers
  - Parameter shift rule and trigonometric polynomial form
  - Nakanishi-Fujii-Todo (NFT) methods
- Proposed Methods
  - VQE Kernel (generalization of Dirichlet kernel)
  - Expected maximum improvement over confident Regions
  - Experimental results
- Summary

VQE (Dirichlet) kernel

Use GP with VQE kernel:

$$k^{\mathrm{VQE}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_0^2 \prod_{d=1}^{D} \left(\frac{\gamma^2 + 2\cos(x_d - x'_d)}{\gamma^2 + 2} \right)$$

that satisfies $k^{\mathrm{VQE}}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$, where

$$\boldsymbol{\phi}(\boldsymbol{x}) = \frac{\sigma_0}{(\gamma^2 + 2)^{D/2}} \underbrace{\mathbf{vec}\left(\otimes_{d=1}^{D} (\gamma, \sqrt{2}\cos x_d, \sqrt{2}\sin x_d)^\top \right)}_{\text{Basis of trigonometric polynomial (or Fourier series)}}$$

➡ Any (prior and posterior) sample from GP is in the trigonometric polynomial form.

Need new acquisition function

Most of the acquisition functions assume that
the optimum is one of the observed points.

$$a_{\mathcal{D}}^{\mathrm{UCB}}(\boldsymbol{x}') = -\left(\mu_{\mathcal{D}}(\boldsymbol{x}') - \beta\sigma_{\mathcal{D}}(\boldsymbol{x}')\right)$$

$$a_{\mathcal{D}}^{\mathrm{PI}}(\boldsymbol{x}') = \left\langle \mathbf{1}(f^{\dagger} > f') \right\rangle_{p(f'|\boldsymbol{x}',\mathcal{D})},$$

$$a_{\mathcal{D}}^{\mathrm{EI}}(\boldsymbol{x}') = \left\langle \max(0, f^{\dagger} - f') \right\rangle_{p(f'|\boldsymbol{x}',\mathcal{D})}, \qquad \text{where} \qquad f' = f(\boldsymbol{x}')$$

$f^*(\boldsymbol{x})$

In NFT, optimum can be far
from any observed point.

Confident Region (CoRe)

Current optimum

Confident Region (CoRe)

Confident Region (CoRe)

$$\mathcal{Z}_{\boldsymbol{X}} = \left\{ \boldsymbol{x} \in \mathcal{X}; s_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa^2 \right\}$$

Confident Region (CoRe)

$$\mathcal{Z}_{\boldsymbol{X}} = \left\{ \boldsymbol{x} \in \mathcal{X}; s_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa^2 \right\}$$

EMICoRe: Expected maximum improvement over CoRe

$$a^{\mathrm{EMICoRe}}(\boldsymbol{X}') = \frac{1}{M}\left\langle \max\left(0, \min_{\boldsymbol{x}\in\mathcal{Z}_{\boldsymbol{X}}} f(\boldsymbol{x}) - \min_{\boldsymbol{x}\in\mathcal{Z}_{(\boldsymbol{X},\boldsymbol{X}')}} f(\boldsymbol{x})\right)\right\rangle_{p(f(\cdot)|\boldsymbol{X},\boldsymbol{y})}$$

EMICoRe: Expected maximum improvement over CoRe

$$a^{\mathrm{EMICoRe}}(\boldsymbol{X}') = \frac{1}{M}\left\langle \max\left(0, \underbrace{\min_{\boldsymbol{x}\in\mathcal{Z}_{\boldsymbol{X}}} f(\boldsymbol{x})}_{\text{Optimum in prior CoRe (before new observations)}} - \underbrace{\min_{\boldsymbol{x}\in\mathcal{Z}_{(\boldsymbol{X},\boldsymbol{X}')}} f(\boldsymbol{x})}_{\text{Optimum in posterior CoRe (after new observations)}}\right)\right\rangle_{p(f(\cdot)|\boldsymbol{X},\boldsymbol{y})}$$

EMICoRe: Expected maximum improvement over CoRe

$$a^{\mathrm{EMICoRe}}(\boldsymbol{X}') = \frac{1}{M}\left\langle \max\left(0, \underbrace{\min_{\boldsymbol{x}\in\mathcal{Z}_{\boldsymbol{X}}} f(\boldsymbol{x})}_{\text{Optimum in prior CoRe (before new observations)}} - \underbrace{\min_{\boldsymbol{x}\in\mathcal{Z}_{(\boldsymbol{X},\boldsymbol{X}')}} f(\boldsymbol{x})}_{\text{Optimum in posterior CoRe (after new observations)}}\right)\right\rangle_{p(f(\cdot)|\boldsymbol{X},\boldsymbol{y})}$$

With EMICoRe, we optimize the distances (α_1, α_2) of new measurement points from the current optimum.

Experimental setting

Quantum circuit: Q-qbit, L-layered *Efficient SU(2)* with (Q, L) = (3, 3), (5, 3), (7, 5).

$$D = (2 + (L \cdot 2)) \cdot Q$$

Hamiltonian:

$$H = -\left[\sum_{j=1}^{Q-1}(J_X\sigma_j^X\sigma_{j+1}^X + J_Y\sigma_j^Y\sigma_{j+1}^Y + J_Z\sigma_j^Z\sigma_{j+1}^Z) + \sum_{j=1}^{Q}(h_X\sigma_j^X + h_Y\sigma_j^Y + h_Z\sigma_j^Z)\right]$$

Ising: $J_X = -1,\ J_Y = J_Z = 0, h_X = h_Y = 0,\ h_Z = -1$

Heisenberg: $J_X = J_Y = J_Z = 1, h_X = h_Y = h_Z = 1$

Quantum computation: Classical simulation via qiskit

Experimental results (Q, L) = (3, 3)

Ising:

Heisenberg:

Achieved energy

$f^*(\widehat{\boldsymbol{x}})$

Fidelity

$\langle \psi_{\mathrm{GS}} | \psi_{\widehat{\boldsymbol{x}}} \rangle$

Experimental results (Q, L) = (5, 3)

Ising:

Heisenberg:

Achieved energy

$f^*(\widehat{\boldsymbol{x}})$

Fidelity

$\langle \psi_{\mathrm{GS}} | \psi_{\widehat{\boldsymbol{x}}} \rangle$

Experimental results (Q, L) = (7, 5)

Achieved energy

$f^*(\widehat{\boldsymbol{x}})$

Fidelity

$\langle \psi_{\mathrm{GS}} | \psi_{\widehat{\boldsymbol{x}}} \rangle$

Summary

- Gaussian process (GP) and Bayesian optimization (BO)
  - GP: Distribution on function (infinite dimensional) values.
  - BO: Exploitation-exploration trade-off with acquisition functions.

- Variational quantum eigensolvers (VQE)
  - Classical-quantum computing protocol, where the classical part solves noisy black-box optimization.
  - Trigonometric polynomial form (TPF) allows NFT algorithm, sequential minimal optimization with a few observations in each step.

- VQE (Dirichlet) kernel and EMICoRe
  - Make use of TPF of VQE objective, and apply BO with GP regression.
  - Enhance NFT.

Experimental results (Q, L) = (5, 3) longer run

Ising:

Achieved energy

$f^*(\widehat{\boldsymbol{x}})$

Fidelity

$\langle \psi_{\mathrm{GS}} | \psi_{\widehat{\boldsymbol{x}}} \rangle$

Experimental results (Q, L) = (5, 3) EI baseline

Ising:

Achieved energy

$f^*(\widehat{\boldsymbol{x}})$

Fidelity

$\langle \psi_{\mathrm{GS}} | \psi_{\widehat{\boldsymbol{x}}} \rangle$

Experimental results (Q, L) = (3, 3) plain BO

Ising:

Achieved energy

$f^*(\widehat{\boldsymbol{x}})$

Fidelity

$\langle \psi_{\mathrm{GS}} | \psi_{\widehat{\boldsymbol{x}}} \rangle$