Quantum Natural Gradient

Variational Quantum Algorithms · Module 6 — Optimization Dynamics

Node 6.2

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Optimization Dynamics **Node:** 6.2

The vanilla gradient descent of node 6.1 has a glaring conceptual problem: **it depends on how you parameterize the ansatz**.

If you reparameterize $\theta \to 2\theta$ for some parameter, the gradient component scales by $1/2$ (chain rule), but the optimal step doesn't. Vanilla GD takes the wrong-sized step under reparameterization. For physically equivalent ansätze with different parameterizations, GD produces different trajectories.

This is unsatisfying. The **natural gradient** is the fix: rescale the gradient by a metric on parameter space, chosen so that the resulting update is invariant under reparameterization. For VQAs, that metric is the **quantum Fisher information matrix (QFIM)**, also known as the Fubini-Study metric, also discussed in node 2.9 as a property of the ansatz manifold.

This node walks through what natural gradient means, why it's the "right" geometry for VQAs, and what it costs to compute.

The Update Rule

Natural gradient descent uses the update

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta\, g^{-1}(\boldsymbol{\theta}_k)\, \nabla C(\boldsymbol{\theta}_k),$$

where $g(\boldsymbol{\theta})$ is the QFIM at point $\boldsymbol{\theta}$.

Compare to vanilla GD:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta\, I\, \nabla C(\boldsymbol{\theta}_k).$$

The only difference is replacing the identity I with g^{-1}. But this difference matters: it converts the update from a step in *parameter coordinates* to a step in *state space distance*.

What The QFIM Is

The QFIM is the Riemannian metric on the manifold of pure states, pulled back through the parameterization:

$$g_{ij}(\boldsymbol{\theta}) = 4\,\mathrm{Re}\left[\langle \partial_i\psi|\partial_j\psi\rangle - \langle \partial_i\psi|\psi\rangle\langle\psi|\partial_j\psi\rangle\right].$$

Equivalently, it is the Hessian of the **quantum infidelity** $1 - |\langle\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta}+\boldsymbol{\delta})\rangle|^2$ at $\boldsymbol{\delta} = 0$, divided by 2.

Key properties:

- **Positive semidefinite.** $g \geq 0$.
- **Reparameterization-covariant.** If $\boldsymbol{\theta} \to \boldsymbol{\phi}(\boldsymbol{\theta})$, then $g \to J^T g J$ where J is the Jacobian. The combination $g^{-1}\nabla C$ is invariant.
- **Captures state distinguishability.** Two parameter directions with small g_{ii} produce nearly indistinguishable states; large g_{ii} means easily distinguishable.
- **Has a kernel for gauge-redundant ansätze.** If the parameterization has continuous gauge directions (Section 3.4), the QFIM has zero eigenvalues in those directions, and g^{-1} doesn't exist.

The kernel issue is technically inconvenient and is usually handled by replacing g^{-1} with the **Moore-Penrose pseudoinverse** g^{+}.

Why Natural Gradient Is "Natural"

The intuition is that **vanilla GD treats parameter space as flat**, but the parameterization of the ansatz manifold doesn't preserve flatness in any meaningful way.

For example, if θ_1 and θ_2 are two parameters, vanilla GD assumes that "moving by δ in θ_1" and "moving by δ in θ_2" produce equal-sized state-space changes. This is rarely true. θ_1 might be a parameter that strongly affects the state (a single-qubit rotation on a superposition state), while θ_2 is a parameter with weak effect (a rotation on an eigenstate).

Natural gradient corrects for this by **measuring step size in state space, not parameter space**. The QFIM is the conversion factor.

The result: natural gradient takes uniform steps in the *physically meaningful* metric — Fubini-Study distance on the manifold of states. Two trajectories that produce the same sequence of states but use different parameter labels look identical to natural gradient.

This is the same idea as Fisher's natural gradient in statistics (Amari 1998), which uses the classical Fisher information matrix to make gradient descent on probability distributions invariant under reparameterization. The QFIM is the quantum version.

How To Compute The QFIM

For a parameterized circuit, the QFIM has $P \times P$ entries. Computing one entry requires evaluating two quantum-state inner products:

$$g_{ij} = 4\,\mathrm{Re}\left[\langle \partial_i\psi|\partial_j\psi\rangle - \langle \partial_i\psi|\psi\rangle\langle\psi|\partial_j\psi\rangle\right].$$

Each inner product can be evaluated with a **Hadamard test** circuit, which takes 1 ancilla qubit and a controlled version of the ansatz. So computing the full QFIM takes $O(P^2)$ circuit evaluations, each $O(N)$ shots — comparable to computing a Hessian.

For $P = 50$ and $N = 1000$, the QFIM takes $\sim 10^6$ shots per iteration. This is **5-10x more expensive** than computing the gradient itself, which limits how often you can afford to update the QFIM in practice.

Common cost-saving approximations:

1. **Block-diagonal QFIM.** Only compute the diagonal blocks corresponding to layers or qubits, ignoring cross-block correlations. Cost drops from $O(P^2)$ to $O(P \cdot \text{block-size})$.
2. **Diagonal QFIM.** Only compute g_{ii}, no off-diagonal entries. Cost drops to $O(P)$, the same as the gradient. The resulting "natural gradient" is no longer fully invariant but captures the per-coordinate scaling.

3. **Cached QFIM.** Compute the QFIM every K iterations rather than every iteration. Useful when the QFIM doesn't change much between steps.

4. **K-FAC-style approximations.** Borrowed from classical natural gradient literature: factor the QFIM into smaller, cheaper-to-invert pieces.

In practice, block-diagonal or diagonal QFIM is what most VQA implementations use. The fully invariant natural gradient is theoretically appealing but operationally expensive.

What Natural Gradient Buys You

The empirical case for natural gradient:

1. Faster convergence near critical points. At a minimum, the QFIM rescales the gradient so that all directions are equivalent. This eliminates the "ill-conditioning" problem of vanilla GD on stretched-out minima.

2. Better behavior in the barren plateau regime. In a plateau, the QFIM eigenvalues are typically very small (the state barely changes with parameter changes). Inverting the QFIM amplifies these directions. This can, in principle, recover signal from regions where vanilla GD has none — though in practice, the noise on the QFIM estimation is *also* large, so the amplification mostly amplifies noise.

3. Reparameterization invariance. Two ansätze with different parameter labels but the same physics give identical natural-gradient trajectories. This is a clean conceptual property and useful for fair comparison of ansatz designs.

4. Better step-size selection. With the QFIM, the appropriate step size is in *state-space units* (e.g., "move 0.01 in Fubini-Study distance"), which is more interpretable than parameter-space units.

5. Smoother optimization paths. Natural gradient produces trajectories that are smoother in state space — fewer zig-zags, more direct routes to the minimum. This reduces the number of steps needed.

What Natural Gradient Doesn't Buy You

It is important to be honest about what natural gradient *doesn't* fix:

1. Barren plateaus. Natural gradient doesn't eliminate them. If the gradient is below the shot noise floor, multiplying by g^{-1} (whose entries are also exponentially small) doesn't recover signal. The ratio $\nabla C/g$ might be more useful conceptually, but estimating both numerator and denominator in the noise floor regime is hopeless.

2. Local minima. Natural gradient is still a local first-order optimizer. It finds the critical point closest to the initialization. No global escape.

3. Cost. The QFIM is expensive. For $P > 100$ parameters, computing the full QFIM each iteration is prohibitive. The savings from "fewer iterations" don't always outweigh the "more shots per iteration."

In practice, natural gradient is most useful for **medium-sized VQAs** (10-50 parameters) where the shot cost of the QFIM is tolerable and the convergence speedup is meaningful.

The Stoudenmire-Schwab Variant

A specific natural gradient variant worth knowing about: **Stoudenmire-Schwab quantum natural gradient (Stokes et al. 2020)**.

This computes the QFIM efficiently by using the **block structure** of the parameterized ansatz: each ansatz layer depends on a small number of parameters, and the QFIM block for that layer can be computed by Hadamard tests within the layer alone, using the *previous* layer's output as the input state.

For a layered ansatz with L layers and q parameters per layer, the per-layer QFIM block is $q \times q$ and costs $O(q^2)$ circuit evaluations. The total QFIM is approximately block-diagonal with L blocks, costing $O(Lq^2) = O(Pq)$ total — linear in the number of parameters times the layer width.

For typical VQAs with q small (say 5-10 parameters per layer), this brings the QFIM cost down from quadratic to nearly linear in P, making natural gradient practical for ansätze of ~ 100 parameters.

Stokes et al. (2020) showed that this block-diagonal natural gradient gives most of the convergence benefit of the full QFIM at a fraction of the cost. In modern VQA practice, "natural gradient" usually refers to this block-diagonal version unless otherwise specified.

Structural Role

This node is the **geometric** entry point of Module 6. It introduces the QFIM (formally; node 2.9 introduced it informally) and motivates the use of intrinsic geometry rather than ambient parameter coordinates.

It forwards to Section 6.3 (Adam and adaptive methods, which use diagonal-curvature-like rescalings as a poor-man's natural gradient) and Section 6.6 (optimal transport, which generalizes the natural gradient idea to even richer geometries).

Relations to Other Concepts

- **Fisher information (classical)** — Amari's natural gradient.
- **QFIM / Fubini-Study metric** — the quantum analogue.
- **Riemannian optimization** — the general framework for optimization on curved spaces.
- **Newton's method** — second-order optimization that uses the cost Hessian (different from QFIM).
- **Stokes et al. 2020** — the block-diagonal practical QFIM construction.

Worked Example — QFIM For A Single-Qubit Ansatz

Consider $|\psi(\theta_1, \theta_2)\rangle = R_Y(\theta_2)R_X(\theta_1)|0\rangle$.

Compute the partial derivatives:

$$|\partial_1\psi\rangle = -\frac{i}{2}R_Y(\theta_2)XR_X(\theta_1)|0\rangle.$$
$$|\partial_2\psi\rangle = -\frac{i}{2}YR_Y(\theta_2)R_X(\theta_1)|0\rangle.$$

Compute the inner products:

$$\langle\partial_1\psi|\partial_1\psi\rangle = \frac{1}{4}\langle 0|R_X^\dagger X^2 R_X|0\rangle = \frac{1}{4}.$$
$$\langle\partial_2\psi|\partial_2\psi\rangle = \frac{1}{4}\langle\psi(0)|Y^2|\psi(0)\rangle = \frac{1}{4}.$$
$$\langle\partial_1\psi|\partial_2\psi\rangle = -\frac{1}{4}\langle 0|R_X^\dagger X R_Y^\dagger Y R_Y R_X|0\rangle = ?$$

Working out the third quantity gives a θ_1, θ_2-dependent value.

The QFIM is

$$g(\theta_1, \theta_2) = 4 \begin{pmatrix} \frac{1}{4} - |\langle \partial_1 \psi | \psi \rangle|^2 & \cdots \\ \cdots & \frac{1}{4} - |\langle \partial_2 \psi | \psi \rangle|^2 \end{pmatrix}.$$

For $\theta_1 = 0$, the state is $R_Y(\theta_2)|0\rangle = \cos(\theta_2/2)|0\rangle + \sin(\theta_2/2)|1\rangle$, and the QFIM evaluates to

$$g(0, \theta_2) = \begin{pmatrix} \cos^2(\theta_2/2) & 0 \\ 0 & 1 \end{pmatrix}.$$

Note: at $\theta_2 = \pi$, the (1,1) entry vanishes — the QFIM has a kernel here, and natural gradient is undefined in the θ_1 direction. The vanishing happens because at this point, the state $R_Y(\pi)|0\rangle = |1\rangle$ is an eigenstate of X (up to phase), so an infinitesimal rotation around the X-axis doesn't change the state. There is no direction in state space corresponding to motion in θ_1, so natural gradient pseudo-inverse must handle this gracefully.

This is the kind of degeneracy you see in QFIMs in practice; pseudoinverse handles it cleanly.

Visualization

Pending. A 2D parameter-space plot showing two trajectories on the same cost landscape: one from vanilla GD (zig-zag) and one from natural gradient (smoother). Annotations highlight the QFIM rescaling at each step. Caption: "Vanilla GD walks in coordinates. Natural gradient walks in state space."

Exercises

1. For $|\psi(\theta)\rangle = R_Y(\theta)|0\rangle$, compute the QFIM (which is just a scalar). Show that it equals 1 for all θ, and explain why.
2. For a 2-parameter ansatz with QFIM $g = \begin{pmatrix} 4 & 1 \\ 1 & 1 \end{pmatrix}$, compare the vanilla and natural gradient updates for a fixed gradient $\nabla C = (1, 1)^T$. Which direction does each take?
3. (VQA) Estimate the cost of a single natural gradient step (in shot count) for a 30-parameter ansatz, using the block-diagonal QFIM with 5 parameters per block. Compare to a vanilla gradient step.

Research Extensions

- **K-FAC and other approximate natural gradients** — borrowed from classical deep learning.
- **Online QFIM estimation** — updating the QFIM incrementally rather than recomputing from scratch.
- **Gauge-invariant natural gradient** — formulations that handle continuous gauge redundancies cleanly.
- **Hardware-aware QFIM** — accounting for noise in the QFIM computation itself.

Cross-links to Other Courses

- **Information Geometry** — Amari's natural gradient and Fisher information.
- **Differential Geometry** — Riemannian metrics, geodesics, parallel transport.
- **Module 2 (Ansatz as Manifold, Section 2.9)** — informal introduction to the QFIM.
- **Module 3 (Local Geometry, Section 3.5)** — gradient and Hessian alongside the QFIM.

Variational Quantum Algorithms · Module 6 — Optimization Dynamics · Node 6.2
Concepts: natural-gradient, fisher-information, gradient-descent, inner-product
Prerequisites: 6.1, 3.5, 2.9
Enables: 6.3, 6.5

hbar.university — own.your.cognition