Mini-Course on Information Geometry

Introduction

Herlock Rahimi
Department of Electrical and Computer Engineering
Yale University

June 5, 2025

Overview

Statistical Manifold and Fisher Metric

Let $\mathcal{M} = \{p(x;\xi)\}$ be a statistical model.

- Fisher information metric:

$$g_{ij}(\xi) = \mathbb{E}\left[\partial_i \log p(x;\xi) \partial_j \log p(x;\xi)\right]$$

- Gives $\mathcal{M}$ a Riemannian structure.

Affine Connections: $\nabla^{(\alpha)}$

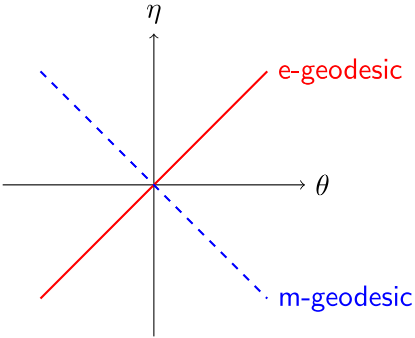
- Define a family of affine connections:

$$\Gamma_{ijk}^{(\alpha)} = \mathbb{E}\left[\partial_i \partial_j \log p + \frac{1-\alpha}{2} \partial_i \log p \partial_j \log p\right] \partial_k \log p$$

- Special cases:
  - $\alpha = 1$: e-connection (exponential)
  - $\alpha = -1$: m-connection (mixture)

Geodesics and Parallel Transport

- **e-geodesic:** Straight line in natural parameter space.
- **m-geodesic:** Linear combination of densities.
- **Dual flatness:** If both $\nabla^{(e)}$ and $\nabla^{(m)}$ are flat.



Dually Flat Structure

- Two affine coordinate systems:
  - θ: natural (e-) coordinates
  - η: expectation (m-) coordinates
- Convex potential functions:

$$\psi(\theta), \quad \varphi(\eta) \quad \text{(Legendre duals)}$$

- Metric:

$$g_{ij} = \partial_i \partial_j \psi(\theta)$$

Exponential and Logarithmic Maps

- **Exponential map:** $\exp_p(v)$: moves along geodesic starting at p in direction v.
- **Logarithmic map:** $\log_p(q)$: inverse of exp.
- In IG:

$$\log_p(q) = \nabla D(q \| p), \quad \exp_p(v) = \arg\min_q \left[D(q \| p) - \langle \nabla D(q \| p), v \rangle \right]$$

Canonical Divergence

- Kullback-Leibler divergence (KL):

$$D_{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

- In exponential families:

$$D(p_\theta\|p_{\theta'}) = \psi(\theta) - \psi(\theta') - (\theta - \theta')^T \eta(\theta')$$

- Convex in θ, defines a Bregman divergence.

Dual Divergence and Dual Connections

- Define $D^*(p\|q) = D(q\|p)$
- D is $\nabla^{(e)}$-convex, D^* is $\nabla^{(m)}$-convex
- Hessians define Fisher metric:

$$g_{ij} = \partial_i \partial_j D(p\|q)\Big|_{p=q}$$

Pythagorean Theorem in IG

Let $p, q, r \in \mathcal{M}$, with $p \perp q$ under g.

- e-projection $p \to q$, m-projection $q \to r$:

$$D(p \| r) = D(p \| q) + D(q \| r)$$

- Fundamental to EM algorithm and variational inference.

Sufficient Statistics and Geometry

- Statistic $T(x)$ is sufficient if $p(x;\theta) = h(x)\exp(\theta^T T(x) - \psi(\theta))$
- **Fisher metric is invariant** under sufficient statistic mapping.

Invariance of the Metric

- If $k(x)$ is sufficient, then:

$$D(q(x)\|p(x)) = D(q(k(x))\|p(k(x)))$$

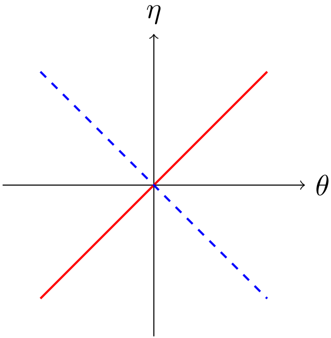
- **KL divergence** and **Fisher metric** remain unchanged.

Unifying Example: 1D Exponential Family

- $p(x;\theta) = \exp(\theta x - \psi(\theta))$
- $\eta = \mathbb{E}_\theta[x] = \psi'(\theta), \quad g(\theta) = \psi''(\theta)$
- KL divergence:
$$D(\theta\|\theta') = \psi(\theta) - \psi(\theta') - (\theta - \theta')\psi'(\theta')$$
- Convex duality: $\theta \leftrightarrow \eta$

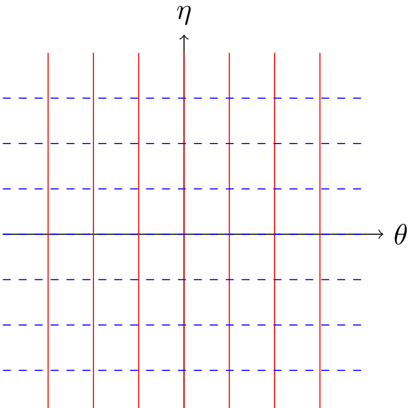
Geodesics in 1D Exponential Family

- **e-geodesic:** $\theta(t) = (1 - t)\theta_0 + t\theta_1$
- **m-geodesic:** $\eta(t) = (1 - t)\eta_0 + t\eta_1$
- **Legendre transform:** $\theta \leftrightarrow \eta, \quad \psi(\theta) + \varphi(\eta) = \theta\eta$



Foliations in Information Geometry

- **Foliation:** partitioning manifold into submanifolds (leaves).
- In IG:
  - e-leaves: surfaces of constant η
  - m-leaves: surfaces of constant θ
- Orthogonal under Fisher metric.



Pythagorean Theorem: Example

Given $\theta_0, \theta_1, \theta_2 \in \mathbb{R}$ in exponential family:

- Let θ_1 be the m-projection of θ_0, θ_2 the e-projection of θ_1:

$$D(\theta_0 \| \theta_2) = D(\theta_0 \| \theta_1) + D(\theta_1 \| \theta_2)$$

- True for canonical divergence D.

Summary and Discussion

- Fisher metric, e-/m-connections form the backbone of IG.
- Divergence functions unify distance, projections, and geodesics.
- Exponential/logarithmic maps give intrinsic manifold navigation.
- Foliations partition manifold by coordinate systems.
- Sufficient statistics and invariance ensure model-consistent structure.

Chapter Overview

- 8.1 EM Algorithm
  - Hidden variable models
  - Alternating projections as EM
  - Examples: Gaussian Mixture and RBM
- 8.2 Information Loss from Data Reduction
- 8.3 Estimation with Misspecified Models

8.1.1 Statistical Model with Hidden Variables

Let $x = (y, h)$ be a random vector with hidden part h. Given $y_1, \dots, y_N$ from the marginal:

$$p_Y(y; \xi) = \int p(y, h; \xi)\, dh$$

We estimate ξ via the simpler model $\mathcal{M} = \{p(x; \xi)\}$.

- Model $\mathcal{M}' = \{p_Y(y; \xi)\}$ may be intractable.
- Use the full model $\mathcal{M}$ with latent variables.

Empirical Distributions and Manifolds

Empirical distribution:

$$\bar{q}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)$$

With hidden variables:

$$\bar{q}(y, h) = \bar{q}_Y(y) q(h|y), \text{ where } q(h|y) \text{ is arbitrary}$$

- The set $\mathcal{D} = \{\bar{q}_Y(y) q(h|y)\}$ forms an m-flat submanifold in S.
- The model $\mathcal{M}$ is an e-flat submanifold.

Goal: Minimize KL divergence from data manifold $\mathcal{D}$ to model $\mathcal{M}$:

$$D_{\mathrm{KL}}[\mathcal{D} : \mathcal{M}] = \min_{q \in \mathcal{D}, p \in \mathcal{M}} \int q(y, h) \log \frac{q(y, h)}{p(y, h; \xi)} \, dy \, dh$$

Alternating minimization:

- E-step: fix ξ, minimize over $q \in \mathcal{D}$.
- M-step: fix q, minimize over ξ.

Geometry of the EM Algorithm

E-step: e-projection from $\mathcal{M}$ to $\mathcal{D}$
M-step: m-projection from $\mathcal{D}$ to $\mathcal{M}$
Likelihood to be maximized:

$$\mathcal{L}(\xi; \xi^{(t)}) = \sum_{i=1}^{N} \int p(h|y_i; \xi^{(t)}) \log p(y_i, h; \xi)\, dh$$

Theorem 8.2 (Amari)

Each iteration of E-step and M-step reduces KL divergence. Converges to local minimum.

Model:

$$p(y, \xi) = \sum_{j=1}^{k} w_j \mathcal{N}(y \mid \mu_j, 1), \quad \sum w_j = 1$$

Hidden variable h indicates which Gaussian y came from:

$$p(y, h = j; \xi) = w_j \mathcal{N}(y \mid \mu_j, 1)$$

E-step and M-step for Gaussian Mixture

E-step:

$$q_t(h|y) = \frac{w_h^{(t)} \mathcal{N}(y \mid \mu_h^{(t)}, 1)}{\sum_j w_j^{(t)} \mathcal{N}(y \mid \mu_j^{(t)}, 1)}$$

M-step: Update weights and means:

$$w_h^{(t+1)} = \frac{1}{N} \sum_i q_t(h|y_i), \quad \mu_h^{(t+1)} = \frac{\sum_i y_i q_t(h|y_i)}{\sum_i q_t(h|y_i)}$$

Chapter Overview

- 12.1 Natural Gradient Stochastic Descent Learning
  - On-line vs batch learning
  - Riemannian geometry of gradient descent
  - Absolute Hessian and SFN
  - Applications to RL and stochastic relaxation
  - Mirror descent, efficiency, adaptivity
- 12.2 Singularities in MLP Learning
  - Elimination and overlap singularities
  - Slow dynamics and plateau phenomena
  - Natural gradient overcomes singular regions

12.1.1 Supervised Learning Framework

We consider the standard supervised setting:

- Inputs $x \in \mathbb{R}^n$, targets $y \in \mathbb{R}$.
- Outputs are generated from $y = f(x) + \varepsilon$, with noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.
- Alternatively, a joint distribution: $p(x, y) = q(x)p_\varepsilon(y - f(x))$

We estimate parameters ξ of a model $f(x, \xi)$ to match $f(x)$.

Loss Function and Training Error

Instantaneous Loss: For regression,

$$l(x, y; \xi) = \frac{1}{2}(y - f(x, \xi))^2$$

Expected Loss (Generalization Error):

$$L(\xi) = \mathbb{E}_{p(x,y)}[l(x, y; \xi)]$$

Empirical Loss (Training Error):

$$L_{\text{train}}(\xi) = \frac{1}{T}\sum_{t=1}^{T} l(x_t, y_t; \xi)$$

Since $p(x, y)$ is unknown, we minimize L_{train}.

Batch vs On-line Gradient Descent

Batch Gradient Descent:

$$\xi_{t+1} = \xi_t - \eta_t \nabla L_{\text{train}}(\xi_t)$$

Stochastic Gradient Descent (On-line):

$$\xi_{t+1} = \xi_t - \eta_t \nabla l(x_t, y_t; \xi_t)$$

- Each update uses one sample only.
- Gradient is a noisy estimate of ∇L.
- Expectation: $\mathbb{E}[\nabla l(x_t, y_t; \xi_t)] = \nabla L(\xi_t)$

This yields stochastic descent with fluctuations.

12.1.2 Why Gradient Descent is Coordinate Dependent

In Euclidean space, gradient descent aligns with the steepest descent direction.
In Riemannian geometry:

- Length depends on metric tensor $G = (g_{ij})$
- Distance: $ds^2 = g_{ij} d\xi^i d\xi^j$
- Fair comparison of directions must respect this geometry.

Natural Gradient: Direction of Steepest Descent

Constrained maximization:

- Maximize: $\nabla L(\xi) \cdot a$ under $g_{ij} a^i a^j = 1$
- Use Lagrange multipliers:

$$\nabla L \cdot a - \lambda g_{ij} a^i a^j$$

- Optimal direction:

$$a \propto G^{-1} \nabla L$$

Natural Gradient:

$$\widetilde{\nabla} L = G^{-1} \nabla L$$

Steepest in terms of local Riemannian metric.

Update Rule Using Natural Gradient

On-line Learning with Natural Gradient:

$$\xi_{t+1} = \xi_t - \eta_t \widetilde{\nabla} l(x_t, y_t; \xi_t) = \xi_t - \eta_t G^{-1}(\xi_t) \nabla l(x_t, y_t; \xi_t)$$

Batch Version:

$$\xi_{t+1} = \xi_t - \eta_t \frac{1}{T} \sum_{i=1}^{T} \widetilde{\nabla} l(x_i, y_i; \xi_t)$$

The Fisher information often serves as the metric:

$$G(\xi) = \mathbb{E}[\nabla \log p(x; \xi) \nabla \log p(x; \xi)^T]$$

12.1.3 Hessian vs Fisher Information

Recall: Newton's method uses the Hessian:

$$\xi_{t+1} = \xi_t - \eta_t H^{-1}(\xi_t) \nabla l(x_t, y_t; \xi_t)$$

Hessian:

$$H(\xi) = \mathbb{E}[\nabla^2 l(x, y; \xi)]$$

Fisher information: If $l = -\log p(x; \xi)$,

$$G(\xi) = \mathbb{E}[\nabla \log p \nabla \log p^T] = \mathbb{E}[\nabla^2 l]$$

- At $\xi = \xi_0$, $H(\xi) = G(\xi)$.
- Elsewhere, they can differ significantly.

Saddle-Free Newton (SFN) via Absolute Hessian

Idea: Saddle points are common in high dimensions.

- Eigenvalues λ_i of H can be negative.
- Newton method may converge to saddle!

Solution: Use absolute Hessian:

$$|H| = O^T \mathrm{diag}(|\lambda_1|, \ldots, |\lambda_n|) O$$

Update Rule:

$$\xi_{t+1} = \xi_t - \eta_t |H(\xi_t)|^{-1} \nabla l(x_t, y_t; \xi_t)$$

This stabilizes and avoids attraction to saddles.

Comparison of Methods

- **Newton:** Fast near optimum, unstable near saddle.
- **Natural Gradient:** Geometrically meaningful, robust.
- **SFN:** Combines second-order speed with saddle-avoidance.

Around optimum: All methods align when $\xi \approx \xi_0$. **In singular regions:** Fisher and $|H|$ behave similarly — both vanish, but their inverses stabilize descent.

12.1.4 Stochastic Relaxation: Energy-Based View

Setup: Consider a cost function $L(x)$ to minimize.
Instead of deterministic search, use probabilistic relaxation:

$$p(x;\beta) = \frac{1}{Z(\beta)} e^{-\beta L(x)} \quad \text{(Gibbs distribution)}$$

- β is the inverse temperature.
- At high β, samples concentrate near minima of $L(x)$.
- $Z(\beta) = \int e^{-\beta L(x)} dx$ is the partition function.

Relaxation via Gradient on Manifold of Distributions

We define an objective over probability distributions $p(x;\xi)$:

$$F(\xi) = \mathbb{E}_{p(x;\xi)}[L(x)]$$

Goal: Find ξ^* minimizing $F(\xi)$.
Natural Gradient Descent:

$$\xi_{t+1} = \xi_t - \eta_t G^{-1}(\xi_t) \nabla F(\xi_t)$$

- $G(\xi)$ is Fisher information matrix of $p(x;\xi)$.
- Respects the geometry of the statistical manifold.

Why Natural Gradient Matters for Relaxation

- Euclidean gradient ignores statistical distance between distributions.
- Fisher information captures sensitivity of $p(x;\xi)$ to ξ.
- Natural gradient gives geodesic flow toward optimal ξ^*.

Outcome: Efficient convergence, avoids poor conditioning.

Stochastic Relaxation in Learning

Examples:

- Boltzmann Machines and RBMs: $p(x;\xi)$ is Gibbs distribution.
- Simulated Annealing: Gradually increase β.
- Contrastive Divergence: Use samples from relaxed $p(x;\xi)$.

Natural Gradient improves:

- Learning speed
- Robustness to curvature
- Interpretation as minimizing KL divergence

In reinforcement learning, we optimize a policy $\pi(a \mid s; \xi)$ to maximize expected return.

Objective:

$$J(\xi) = \mathbb{E}_\pi[R] = \mathbb{E}_\pi\left[\sum_t \gamma^t r_t\right]$$

Policy Gradient:

$$\nabla J(\xi) = \mathbb{E}_\pi\left[\nabla \log \pi(a_t \mid s_t; \xi) R_t\right]$$

Natural Policy Gradient

Fisher Information for Policy:

$$G(\xi) = \mathbb{E}_\pi \left[\nabla \log \pi(a_t \mid s_t; \xi) \nabla \log \pi(a_t \mid s_t; \xi)^T \right]$$

Natural Policy Gradient:

$$\widetilde{\nabla} J(\xi) = G(\xi)^{-1} \nabla J(\xi)$$

Update:

$$\xi_{t+1} = \xi_t + \eta_t \widetilde{\nabla} J(\xi_t)$$

This accounts for the curvature of the policy space.

Benefits of Natural Policy Gradient

- Respects the information geometry of the policy manifold.
- Invariant under reparametrization of π.
- Converges faster and more stably than vanilla gradient.
- Key in Trust Region Policy Optimization (TRPO) and PPO.

Relation to Stochastic Relaxation

- Policy $\pi(a \mid s; \xi)$ is a Gibbs distribution over actions.
- RL becomes a stochastic relaxation over policies.
- Natural gradient gives optimal local search direction.

Summary: Reinforcement learning benefits greatly from the natural gradient approach due to its principled geometric structure.

References

Amari, S. (2016). *Information Geometry and Its Applications*. Springer. Dempster et al. (1977), Csiszár and Tusnády (1984), Oizumi et al. (2011).
Amari, S. (2016). *Information Geometry and Its Applications*. Springer. Amari (1998), Cousseau et al. (2008), Dauphin et al. (2014), Peters and Schaal (2008), Martens (2015).