Information Geometric Optimization

Seb Arnold - November 9, 2018
Outline

Goal
Draw connection from natural gradient to various ML/optimization topics.

Outline
1. Euclidean and Riemannian Gradient Optimization
2. Relation to Second-Order Optimization
3. Relation to Evolutionary Strategies
4. Relation to Variational Inference
5. Conclusions
Euclid vs Riemann

Euclidean Geometry

• Any 2 points can be connected.
• A straight line can be extended infinitely.
• A circle is described by its center and radius.
• All right angles are equal to one another.
• If two lines intersect, the sum of interior angles of any segment connecting them is less than 180 degrees.

Riemannian Geometry drops the last two axioms, in order to study smooth manifolds endowed with a local metric.
Computing Distances

**Question** How do we compute the shortest distance between $\theta_1$ and $\theta_2$ on a differentiable manifold $M$?
Computing Distances

**Question** How do we compute the shortest distance between $\theta_1$ and $\theta_2$ on a differentiable manifold $M$?

**Answer** Find the geodesic curve $C$ and then,

$$d(\theta_1, \theta_2) = \int_{-\infty}^{\infty} ||C'(t)|| dt = \int_{-\infty}^{\infty} \sqrt{C'(t)^\top F C'(t)} dt$$

**Warning** In the Riemannian case, the metric $F$ depends on $t$. 
**Observation 1** Information about the geometry of the manifold can help us navigate it.
Gradient Descent

\[ \theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} f(\theta_t) \]
Riemannian Gradient Optimization

Gradient Descent

\[ \theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} f(\theta_t) \approx \arg \max_{\theta'} \left\{ f(\theta') - \frac{1}{2\alpha} \| \theta_t - \theta' \|^2 \right\} + O(\alpha^2) \]
Riemannian Gradient Optimization

Gradient Descent

\[ \theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} f(\theta_t) \approx \arg\max_{\theta'} \left\{ f(\theta') - \frac{1}{2\alpha} \| \theta_t - \theta' \|_2^2 \right\} + O(\alpha^2) \]

**Problem** Why use \( L_2 \) as the metric?
Riemannian Gradient Optimization

Gradient Descent

\[ \theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} f(\theta_t) \approx \arg \max_{\theta'} \left\{ f(\theta') - \frac{1}{2\alpha} \| \theta_t - \theta' \|^2 \right\} + \mathcal{O}(\alpha^2) \]

**Problem** Why use $L_2$ as the metric?

Riemannian Gradient Descent

\[ \theta_{t+1} = \theta_t - \alpha F^{-1} \nabla_{\theta_t} f(\theta_t) \]

**Remark 1** We use $F^{-1}$ to recondition the problem back to Euclidean space. (c.f. Nash Embedding Theorem)
Observation 2 Our ML models are probability distributions.

Remark 2 We often minimize the KL divergence.
The Natural Gradient

**Observation 2** Our ML models are probability distributions.

**Remark 2** We often minimize the KL divergence.

**Idea** Let’s put 2 and 2 together.

Optimize on the Riemannian space of probability distributions defined by the KL divergence.

\[
F = \mathbb{E}_{x \sim P_X} \left[ \nabla \log p_\theta(x) \cdot \nabla \log p_\theta(x) \right]^\top
\]
The Natural Gradient

**Observation 2** Our ML models are probability distributions.

**Remark 2** We often minimize the KL divergence.

**Idea** Let’s put 2 and 2 together.

Optimize on the Riemannian space of probability distributions defined by the KL divergence.

\[
F = \mathbb{E}_{x \sim P_X} \left[ \nabla \log p_\theta (x) \cdot \nabla \log p_\theta (x)^\top \right]
\]

\[
\tilde{\nabla}_\theta = F^{-1} \nabla_\theta f(\theta)
\]
Parameter & Function Space

Parameter Space

Function Space

p
Popular Variation

Let

- \((x, y) \sim D_{X,Y}\) be a set of data points, and
- \(F^{-1}\nabla_{\theta} f(\theta)\) be the natural gradient of the model \(p_\theta(y|x)\).
Popular Variation

Let

- \((x, y) \sim D_{X,Y}\) be a set of data points, and
- \(F^{-1} \nabla \theta f(\theta)\) be the natural gradient of the model \(p_\theta(y|x)\).

**True Fisher Matrix**

\[
F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim p_{Y|X}} \left[ \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right] \right]
\]
Let

- \((x, y) \sim D_{X,Y}\) be a set of data points, and
- \(F^{-1}\nabla_{\theta} f(\theta)\) be the natural gradient of the model \(p_{\theta}(y|x)\).

**True Fisher Matrix**

\[
F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim p_Y|x} \left[ \nabla \log p_{\theta}(y|x) \cdot \nabla \log p_{\theta}(y|x)^{\top} \right] \right]
\]

**Empirical Fisher Matrix**

\[
\bar{F} = \mathbb{E}_{x,y \sim D_{X,Y}} \left[ \nabla \log p_{\theta}(y|x) \cdot \nabla \log p_{\theta}(y|x)^{\top} \right]
\]
The Issue

Computing $F^{-1}$ is often intractable!

Computing those expectations is hard.
The Issue

Computing $F^{-1}$ is often intractable!

Computing those expectations is hard.

Consider a Gaussian model with $\theta = [\mu, \Sigma]$. If $|\mu| = n$,

- storing $F$ requires $\mathcal{O}(n^4)$ memory,
- computing $F^{-1}$ requires at least $\mathcal{O}\left((n^2)^{2.373}\right)$ time complexity.
The Issue

Computing $F^{-1}$ is often intractable!

Computing those expectations is hard.

Consider a Gaussian model with $\theta = [\mu, \Sigma]$. If $|\mu| = n$,

- storing $F$ requires $\Theta(n^4)$ memory,
- computing $F^{-1}$ requires at least $\Theta((n^2)^{2.373})$ time complexity.

We resort to approximations such as K-FAC, Topmoumoute, TANGO, TRPO, etc…
Relation to 2nd Order Optimization

\[ F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_{y|x}} \left[ \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right] \right] \]

\[ = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_{y|x}} \left[ -\nabla^2 \log p_\theta(y|x) \right] \right] \]
Relation to 2nd Order Optimization

\[ F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_Y|x} \left[ \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right] \right] \]

\[ = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_Y|x} \left[ -\nabla^2 \log p_\theta(y|x) \right] \right] \]

Newton’s Method [1]

\[ H = \mathbb{E}_{x,y \sim D_{X,Y}} \left[ \nabla^2 \log p_\theta(y|x) \right] \]
Relation to 2nd Order Optimization

\[ F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_{y|x}} \left[ \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right] \right] \]

\[ = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_{y|x}} \left[ -\nabla^2 \log p_\theta(y|x) \right] \right] \]

Newton’s Method [1]

\[ H = \mathbb{E}_{x,y \sim D_{X,Y}} \left[ \nabla^2 \log p_\theta(y|x) \right] \]

Adagrad

\[ C = \mathbb{E}_{x,y \sim D_{X,Y}} \left[ \left( \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right)^\frac{1}{2} \right] \]

**Note** Adam is a diagonalized Adagrad + momentum.
Evolutionary Strategies

Consider an objective $\mathbb{E}_{x \sim P_x}[f(x)]$, and parameterized density $p_{\theta}(x)$.

$$F^{-1}\nabla_\theta \mathbb{E}_{x \sim P_x}[f(x)] = \mathbb{E}_{x \sim P_x}[f(x)F^{-1}\nabla_\theta \log p_{\theta}(x)]$$
Consider an objective $\mathbb{E}_{x \sim P_x}[f(x)]$, and parameterized density $p_\theta(x)$.

\[
F^{-1} \nabla_\theta \mathbb{E}_{x \sim P_x}[f(x)] = \mathbb{E}_{x \sim P_x}[f(x)F^{-1} \nabla_\theta \log p_\theta(x)]
\]

By making $f(x)$ invariant to increasing transformations, [2] obtain a time-continuous gradient-flow ODE.

Choosing the family of $p_\theta$ and discretizing it with Euler’s method, they recover

- CMA-ES for Gaussian families,
- PBIL for Bernouilli families,
- and more. (NES, CEM, cGA, EMNA, xNES, …)
Evolutionary Strategies

Consider an objective $\mathbb{E}_{x \sim P_x}[f(x)]$, and parameterized density $p_\theta(x)$.

$$F^{-1}\nabla_\theta \mathbb{E}_{x \sim P_x}[f(x)] = \mathbb{E}_{x \sim P_x}[f(x)F^{-1}\nabla_\theta \log p_\theta(x)]$$

By making $f(x)$ invariant to increasing transformations, [2] obtain a time-continuous gradient-flow ODE.

Choosing the family of $p_\theta$ and discretizing it with Euler’s method, they recover

- CMA-ES for Gaussian families,
- PBIL for Bernouilli families,
- and more. (NES, CEM, cGA, EMNA, xNES, …)

**Note** This works even if $f$ is not differentiable!
Pause

Questions / Break?

Up Next Noisy Natural Gradient $\approx$ Variational Inference
Variational Inference for BNNs

Goal

$$\max_{\xi} \text{KL}(q_{\xi}(\theta) \| p(\theta | x, y))$$

ELBO

$$\max_{\xi} \mathbb{E}_{\theta \sim Q_{\theta}} \left[ \mathbb{E}_{x, y \sim D_{X, Y}} \left[ \log p_{\theta}(y | x) \right] \right] - \lambda \text{KL}(q_{\xi}(\theta) \| p(\theta))$$

With

- $(x, y) \sim D_{X, Y}$ a set of data points,
- $p(\theta)$ the prior over the parameter of the model $p_{\theta}(y | x)$,
- $q_{\xi}(\theta)$ the variational posterior over the parameters $\theta$. 
NGPE vs NGVI

Both compute $\tilde{\nabla}_\xi = F^{-1} \nabla_\xi \mathcal{L}(\xi)$.

**NGPE [3]**

$$F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_{Y|X}} \left[ \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right] \right]$$

**NGVI [4]**

$$F = \mathbb{E}_{\theta \sim Q_\Theta} \left[ \nabla \log q_\xi(\theta) \cdot \nabla \log q_\xi(\theta)^\top \right]$$

**Note** This version is often tractable, if $q_\xi$ is nice.
NGPE vs NGVI

Both compute $\tilde{\nabla}_\xi = F^{-1} \nabla_\xi \mathcal{L}(\xi)$.

**NGPE [3]**

$$F = \mathbb{E}_{x \sim D_X} \left[ \mathbb{E}_{y \sim P_{y|x}} \left[ \nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top \right] \right]$$

**NGVI [4]**

$$F = \mathbb{E}_{\theta \sim Q_\theta} \left[ \nabla \log q_\xi(\theta) \cdot \nabla \log q_\xi(\theta)^\top \right]$$

**Note** This version is often tractable, if $q_\xi$ is nice.
A Cool Trick

For any $f(\theta)$, [5]

$$\nabla_{\mu} \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [f(\theta)] = \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [\nabla_{\theta} f(\theta)]$$

$$\nabla_{\Sigma} \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [f(\theta)] = \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [\nabla_{\theta}^2 f(\theta)]$$
A Cool Trick

For any $f(\theta)$, [5]

$$\nabla_\mu \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [f(\theta)] = \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [\nabla_\theta f(\theta)]$$

$$\nabla_\Sigma \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [f(\theta)] = \mathbb{E}_{\theta \sim N(\mu, \Sigma)} [\nabla^2_\theta f(\theta)]$$

**Applied to the ELBO**
Assuming $q_\xi$ is Gaussian with mean $\mu$ and precision $\Lambda$,

$$\tilde{\nabla}_\mu = \Lambda^{-1} \mathbb{E}_{\theta \sim Q_\theta} \left[ \nabla_\theta \log p_\theta(y|x) + \lambda \nabla_\theta \log p(\theta) \right],$$

$$\tilde{\nabla}_\Lambda = -\mathbb{E}_{\theta \sim Q_\theta} \left[ \nabla^2_\theta \log p_\theta(y|x) + \lambda \nabla^2_\theta \log p(\theta) \right] - \lambda \Lambda.$$
Update Rules

Assuming

- \( p(\theta) = N(0, \eta I) \),
- \( \alpha, \beta \) learning rates,
- \( N \) mini-batch size,

\[
\begin{align*}
\mu &\leftarrow \mu + \alpha \Lambda^{-1} \left[ \nabla_\theta \log p_\theta(y|x) - \frac{\lambda}{N\eta} \theta \right] \\
\Lambda &\leftarrow \left( 1 - \frac{\lambda \beta}{N} \right) \Lambda - \beta \left[ \nabla^2_\theta \log p_\theta(y|x) - \frac{\lambda}{N\eta} I \right]
\end{align*}
\]
Update Rules

Assuming

- \( p(\theta) = N(0, \eta I) \),
- \( \alpha, \beta \) learning rates,
- \( N \) mini-batch size,

\[
\mu \leftarrow \mu + \alpha \Lambda^{-1} \left[ \nabla_\theta \log p_\theta(y|x) - \frac{\lambda}{N\eta} \theta \right]
\]

\[
\Lambda \leftarrow \left( 1 - \frac{\lambda \beta}{N} \right) \Lambda - \beta \left[ \nabla^2_\theta \log p_\theta(y|x) - \frac{\lambda}{N\eta} I \right]
\]

**Note** \( \nabla^2_\theta \log p_\theta(y|x) \) is annoying. Let’s replace it with (a diagonal) \( F \) !
**Noisy Natural Gradient**

---

### Noisy Adam

**Algorithm 1** Noisy Adam. Differences from standard Adam are shown in blue.

**Require:** \( \alpha \): Stepsize

**Require:** \( \beta_1, \beta_2 \): Exponential decay rates for updating \( \mu \) and the Fisher \( \mathbf{F} \)

**Require:** \( \lambda, \eta, \gamma_{ex} \): KL weighting, prior variance, extrinsic damping term

\[ m \leftarrow 0 \]

Calculate the intrinsic damping term \( \gamma_{in} = \frac{\lambda}{N \eta} \), total damping term \( \gamma = \gamma_{in} + \gamma_{ex} \)

**while** stopping criterion not met **do**

\[ w \sim \mathcal{N}(\mu, \frac{\lambda}{N} \text{diag}(f + \gamma_{in})^{-1}) \]

\[ v \leftarrow \nabla_w \log p(y|x, w) - \gamma_{in} \cdot w \]

\[ m \leftarrow \beta_1 \cdot m + (1 - \beta_1) \cdot v \quad \text{(Update momentum)} \]

\[ f \leftarrow \beta_2 \cdot f + (1 - \beta_2) \cdot (\nabla_w \log p(y|x, w)^2 \]

\[ \bar{m} \leftarrow m / (1 - \beta^k) \]

\[ \bar{m} \leftarrow \bar{m} / (f + \gamma) \]

\[ \mu \leftarrow \mu + \alpha \cdot \bar{m} \quad \text{(Update parameters)} \]

**end while**

---

**Note** The major modification is sampling the parameters.

**Note** A similarly simple modification can be applied to K-FAC.

---

**Conclusion**

Noisy Natural Gradient \( \implies \) Variational Inference!

**Quiz** Does this ring a bell?
Conclusion Noisy K-FAC optimizes the ELBO faster and better.
Experiments – Boston Housing

Figure 1: Normalized precision matrices for Gaussian variational posteriors trained using noisy natural gradient. We used a network with 2 hidden layers of 15 units each, trained on the Boston housing dataset.

**Conclusion** Noisy K-FAC provides decent approximation of the full precision matrix.
Experiments – CIFAR10

- **D** Data augmentation
- **B** Batch normalization
- **N/A** Unstable training

<table>
<thead>
<tr>
<th>METHOD</th>
<th>NETWORK</th>
<th>TEST ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>D</td>
</tr>
<tr>
<td>SGD</td>
<td>VGG16</td>
<td>81.79</td>
</tr>
<tr>
<td>KFAC</td>
<td>VGG16</td>
<td>82.39</td>
</tr>
<tr>
<td>BBB</td>
<td>VGG16</td>
<td>82.82</td>
</tr>
<tr>
<td>Noisy-Adam</td>
<td>VGG16</td>
<td>82.68</td>
</tr>
<tr>
<td>Noisy-KFAC</td>
<td>VGG16</td>
<td><strong>85.52</strong></td>
</tr>
</tbody>
</table>

**Conclusion** Noisy methods generalize better.
Experiments – CIFAR10

**Conclusion** Noisy methods are better calibrated.
Experiments – Active Learning

UCI Datasets 20 training samples, 100 testing samples, rest is unlabeled pool.

Setup
Repeat for 10 iterations.

1. Fit train data.
2. Compute test error.
3. Compute posterior predictive variance for each pool sample.
4. Choose sample which most reduces posterior entropy. (highest info. gain)
5. Add it to train set with its label.

Note HMC is considered the gold standard.
Experiments – Active Learning

- R Samples uniformly at random. - A Samples from active learning.

Table 3: Average test RMSE in active learning.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PBP_R</th>
<th>PBP_A</th>
<th>NNG-FFF_R</th>
<th>NNG-FFF_A</th>
<th>NNG-MVG_R</th>
<th>NNG-MVG_A</th>
<th>HMC_R</th>
<th>HMC_A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td>6.716±0.500</td>
<td>5.480±0.175</td>
<td>5.911±0.250</td>
<td>5.435±0.132</td>
<td>5.831±0.177</td>
<td>5.220±0.132</td>
<td>5.750±0.222</td>
<td>5.156±0.150</td>
</tr>
<tr>
<td>Concrete</td>
<td>12.417±0.392</td>
<td>11.894±0.254</td>
<td>12.583±0.168</td>
<td>12.563±0.142</td>
<td>12.301±0.203</td>
<td>11.671±0.175</td>
<td><strong>10.564±0.198</strong></td>
<td>11.484±0.191</td>
</tr>
<tr>
<td>Energy</td>
<td>3.743±0.121</td>
<td>3.399±0.064</td>
<td>4.011±0.087</td>
<td>3.761±0.068</td>
<td>3.635±0.084</td>
<td>3.211±0.076</td>
<td>3.264±0.067</td>
<td><strong>3.118±0.062</strong></td>
</tr>
<tr>
<td>Kin8nm</td>
<td>0.259±0.006</td>
<td>0.254±0.005</td>
<td>0.246±0.004</td>
<td>0.252±0.003</td>
<td>0.243±0.003</td>
<td>0.244±0.003</td>
<td>0.226±0.004</td>
<td><strong>0.223±0.003</strong></td>
</tr>
<tr>
<td>Naval</td>
<td>0.015±0.000</td>
<td>0.016±0.000</td>
<td>0.013±0.000</td>
<td>0.013±0.000</td>
<td>0.010±0.000</td>
<td>0.009±0.000</td>
<td>0.013±0.000</td>
<td>0.012±0.000</td>
</tr>
<tr>
<td>Pow. Plant</td>
<td>5.312±0.108</td>
<td>5.068±0.082</td>
<td>5.812±0.119</td>
<td>5.423±0.111</td>
<td>5.377±0.133</td>
<td>4.974±0.078</td>
<td>5.229±0.097</td>
<td><strong>4.800±0.074</strong></td>
</tr>
<tr>
<td>Wine</td>
<td>0.945±0.044</td>
<td>0.809±0.011</td>
<td>0.730±0.011</td>
<td>0.748±0.008</td>
<td>0.752±0.014</td>
<td>0.746±0.009</td>
<td>0.740±0.011</td>
<td>0.749±0.010</td>
</tr>
<tr>
<td>Yacht</td>
<td>5.388±0.339</td>
<td>4.508±0.158</td>
<td>7.381±0.309</td>
<td>6.583±0.264</td>
<td>7.192±0.280</td>
<td>6.371±0.204</td>
<td>4.644±0.237</td>
<td><strong>3.211±0.120</strong></td>
</tr>
</tbody>
</table>

Conclusion NNG-MVG_R performs better than NNG-MVG_A and is closer to HMC_A than PBP_A and NNG-FFF_A. (But other uncertainty measures might be required.)
Experiments – Reinforcement Learning

**Setup** Use VIME, replacing BBB’s posterior with the one from NNG-MVG.

Figure 3: Performance of [TRPO] TRPO baseline with Gaussian control noise, [TRPO+BBB] VIME baseline with BBB dynamics network, and [TRPO+NNG-MVG] VIME with NNG-MVG dynamics network (ours). The darker-colored lines represent the median performance in 10 different random seeds while the shaded area show the interquartile range.

**Conclusion** Better uncertainty estimates help for exploration.
Weight-Perturbed Adam

Contrast with Zhang & al.

1. Focuses on Gaussian mean-field. (i.e. diagonal covariances)
2. Also motivated by “use natural gradient for VI and then simplify it.”
3. Their way of simplification:
   3.1 Start with Newton,
   3.2 Approximate Hessian with GGN,
   3.3 Approximate Hessian with $g^2$,
   3.4 Add momentum,
   3.5 Obtain Vadam.
4. Vprop, Vadagrad, and variants.

Vadam
1: while not converged do
2: $\theta \leftarrow \mu + \sigma \circ \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, $\sigma \leftarrow 1/\sqrt{Ns + \lambda}$
3: Randomly sample a data example $D_i$
4: $g \leftarrow -\nabla \log p(D_i|\theta)$
5: $m \leftarrow \gamma_m m + (1 - \gamma_m) (g + \lambda \mu / N)$
6: $s \leftarrow \gamma_s s + (1 - \gamma_s) (g \circ g)$
7: $\tilde{m} \leftarrow m / (1 - \gamma_m^t)$, $\tilde{s} \leftarrow s / (1 - \gamma_s^t)$
8: $\mu \leftarrow \mu - \alpha \tilde{m} / \sqrt{\tilde{s} + \lambda / N}$
9: $t \leftarrow t + 1$
10: end while
Experiments – Logistic Regression

Figure 2. Experiments on Bayesian logistic regression showing (a) posterior approximations on a toy example, (b) performance on ‘USPS-3v5’ measuring negative ELBO, log-loss, and the symmetric KL divergence of the posterior approximation to MF-Exact, (c) symmetric KL divergence of Vadam for various minibatch sizes on ‘Breast-Cancer’ compared to VOGN with a minibatch of size 1.

Conclusion Choosing different Hessian approximations results in qualitatively different posteriors. (Vadam vs VOGN-1)
More on Information Geometry

**Empirical**
- *Two-Stage Metric Learning*, Wang et al., 2014
- *Riemann Manifold Langevin and Hamiltonian Monte Carlo*, Girolami et al., 2011
- *Transfer Learning: A Riemannian Geometry Framework*, Zanini et al., 2018
- *A Natural Policy Gradient*, Kakade, 2002
- *Information Geometry of Quantum Resources*, Girolami, 2017

**Theoretical**
- *Information Geometry of Wasserstein Divergence*, Karakida & Amari, 2017
Recommended Readings

2. Objective Improvement in Information-Geometric Optimization, Ollivier, 2013. (Youtube)
Questions

Another relation to ELBO?

\[ \theta_t - \alpha \tilde{\nabla}_{\theta_t} \mathbb{E}_{x \sim p_{\theta_t}} [f(x)] \approx \arg \max_{\theta'} \left\{ \mathbb{E}_{x \sim p_{\theta_t}} [f(x)] - \frac{1}{2\alpha} \text{KL}(p_{\theta_t} || p_{\theta'}) \right\} \]

Why the Fisher?

- Motivation: The KL is the go-to divergence for distributions.
- Motivation: IGO can result in high densities for diverse parameter solutions.
- Consequence: Many nice theoretical properties. (Invariances)

Why not use the total variation divergence or optimal transport metrics?
References