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 θ_1 and θ_2 on a differentiable manifold M ?

Computing Distances

Question How do we compute the shortest distance between θ_1 and θ_2 on a differentiable manifold M ?

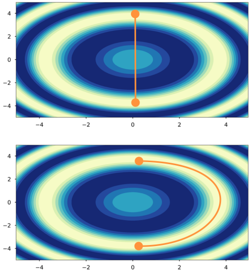
Answer Find the geodesic curve C and then,

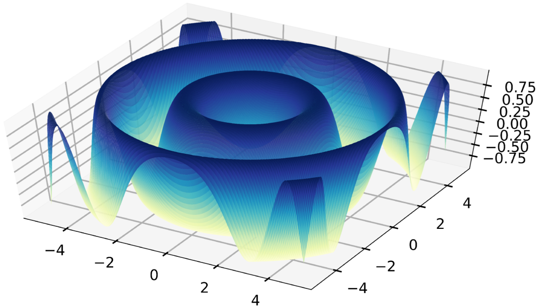
$$d(\theta_1, \theta_2) = \int_{-\infty}^{\infty} ||C'(t)||dt = \int_{-\infty}^{\infty} \sqrt{C'(t)^\top F C'(t)} dt$$

Warning In the Riemannian case, the metric F depends on t.

Why is it relevant ?

Observation 1 Information about the geometry of the manifold can help us navigate it.





Riemannian Gradient Optimization

Gradient Descent

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} f(\theta_t)$$

Riemannian Gradient Optimization

Gradient Descent

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

Riemannian Gradient Optimization

Gradient Descent

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

Problem Why use L_2 as the metric ?

Riemannian Gradient Optimization

Gradient Descent

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

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)

The Natural Gradient

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.

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

The Natural Gradient

Observation 2 Our ML models are probability distributions.

Remark 2 We often minimize the KL divergence.

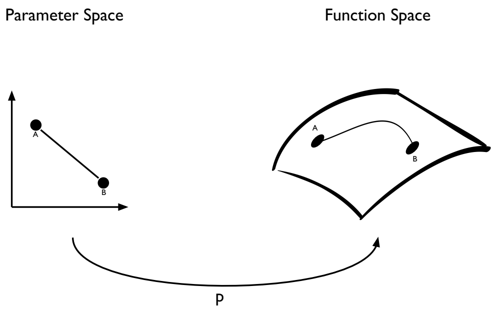
Idea Let's put 2 and 2 together.

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



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

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

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

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

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

We resort to approximations such as K-FAC, Topmoumoute, TANGO, TRPO, etc. . .

Relation to 2nd Order Optimization

$$
\begin{aligned}
F &= \mathbb{E}_{x \sim D_X}\left[\mathbb{E}_{y \sim P_{Y|X}}\left[\nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top\right]\right] \\
&= \mathbb{E}_{x \sim D_X}\left[\mathbb{E}_{y \sim P_{Y|X}}\left[-\nabla^2 \log p_\theta(y|x)\right]\right]
\end{aligned}
$$

Relation to 2nd Order Optimization

$$
\begin{aligned}
F &= \mathbb{E}_{x\sim D_X}\left[\mathbb{E}_{y\sim P_{Y|X}}\left[\nabla \log p_\theta(y|x)\cdot \nabla \log p_\theta(y|x)^\top\right]\right] \\
&= \mathbb{E}_{x\sim D_X}\left[\mathbb{E}_{y\sim P_{Y|X}}\left[-\nabla^2 \log p_\theta(y|x)\right]\right]
\end{aligned}
$$

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

$$
\begin{aligned}
F &= \mathbb{E}_{x \sim D_X}\left[\mathbb{E}_{y \sim P_{Y|X}}\left[\nabla \log p_\theta(y|x) \cdot \nabla \log p_\theta(y|x)^\top\right]\right] \\
&= \mathbb{E}_{x \sim D_X}\left[\mathbb{E}_{y \sim P_{Y|X}}\left[-\nabla^2 \log p_\theta(y|x)\right]\right]
\end{aligned}
$$

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 $p_\theta(x)$.

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

Evolutionary Strategies

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

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

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

Choosing the family of p_θ 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 $p_\theta(x)$.

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

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

Choosing the family of p_θ 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

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

ELBO

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

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 θ.

NGPE vs NGVI

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

NGPE [3]

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

NGVI [4]

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

Note This version is often tractable, if q_ξ is nice.

NGPE vs NGVI

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

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_ξ is nice.

A Cool Trick

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

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

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

A Cool Trick

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

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

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

Applied to the ELBO

Assuming q_{ξ} is Gaussian with mean μ and precision Λ,

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

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

Update Rules

Assuming

- $p(\theta) = N(0, \eta I)$,
- α, β learning rates,
- N mini-batch size,

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

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

Update Rules

Assuming

- $p(\theta) = N(0, \eta I)$,
- α, β learning rates,
- N mini-batch size,

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

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

Note $\nabla_\theta^2 \log p_\theta(y|x)$ 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: α: Stepsize
Require: β_1, β_2: Exponential decay rates for updating $\boldsymbol{\mu}$ and the Fisher $\mathbf{F}$
Require: $\lambda, \eta, \gamma_{\text{ex}}$: KL weighting, prior variance, extrinsic damping term
  $\mathbf{m} \leftarrow \mathbf{0}$
  Calculate the intrinsic damping term $\gamma_{\text{in}} = \frac{\lambda}{N\eta}$, total damping term $\gamma = \gamma_{\text{in}} + \gamma_{\text{ex}}$
  while stopping criterion not met **do**
    $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \frac{\lambda}{N} \operatorname{diag}(\mathbf{f} + \gamma_{\text{in}})^{-1})$
    $\mathbf{v} \leftarrow \nabla_{\mathbf{w}} \log p(y|\mathbf{x}, \mathbf{w}) - \gamma_{\text{in}} \cdot \mathbf{w}$
    $\mathbf{m} \leftarrow \beta_1 \cdot \mathbf{m} + (1 - \beta_1) \cdot \mathbf{v}$ (Update momentum)
    $\mathbf{f} \leftarrow \beta_2 \cdot \mathbf{f} + (1 - \beta_2) \cdot (\nabla_{\mathbf{w}} \log p(y|\mathbf{x}, \mathbf{w}))^2$
    $\tilde{\mathbf{m}} \leftarrow \mathbf{m}/(1 - \beta_1^k)$
    $\hat{\mathbf{m}} \leftarrow \tilde{\mathbf{m}}/(\mathbf{f} + \gamma)$
    $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \alpha \cdot \hat{\mathbf{m}}$ (Update parameters)
  end while

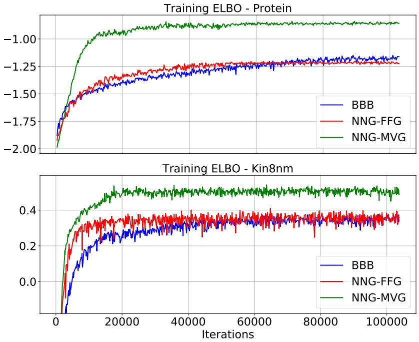
Note The major modification is sampling the parameters.

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

Conclusion
Noisy Natural Gradient $\Longrightarrow$ Variational Inference !

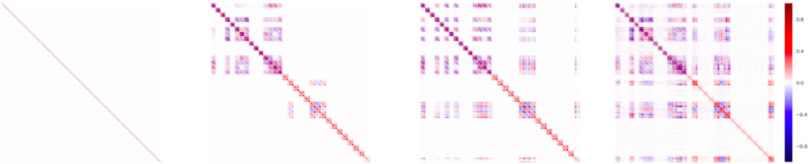
Quiz Does this ring a bell ?

Experiment – Regression



Conclusion Noisy K-FAC optimizes the ELBO faster and better.

Experiments – Boston Housing



(a) Fully factorized (b) Matrix-variate (c) Block tridiagonal (d) Full covariance

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

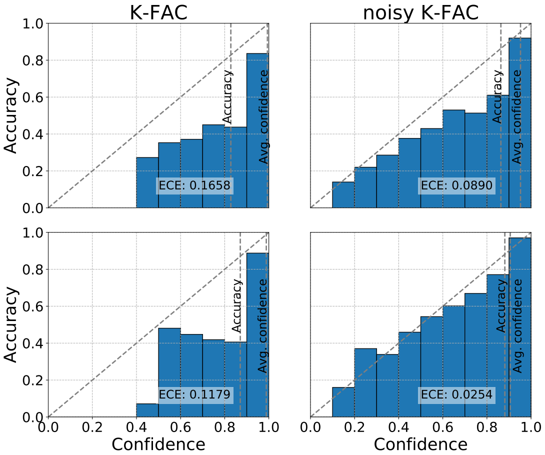
- **D** Data augmentation
- **B** Batch normalization
- **N/A** Unstable training

Method	Network	Test Accuracy			
			D	B	D + B
SGD	VGG16	81.79	88.35	85.75	91.39
KFAC	VGG16	82.39	88.89	86.86	**92.13**
BBB	VGG16	82.82	88.31	N/A	N/A
Noisy-Adam	VGG16	82.68	88.23	N/A	N/A
Noisy-KFAC	VGG16	**85.52**	**89.35**	**88.22**	92.01

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.

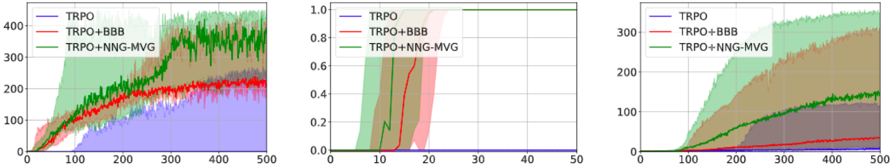
Table 3: Average test RMSE in active learning.

DATASET	PBP_R	PBP_A	NNG-FFG_R	NNG-FFG_A	NNG-MVG_R	NNG-MVG_A	HMC_R	HMC_A
BOSTON	6.716±0.500	5.480±0.175	5.911±0.250	5.435±0.132	5.831±0.177	5.220±0.132	5.750±0.222	**5.156±0.150**
CONCRETE	12.417±0.392	11.894±0.254	12.583±0.168	12.563±0.142	12.301±0.203	11.671±0.175	**10.564±0.198**	11.484±0.191
ENERGY	3.743±0.121	3.399±0.064	4.011±0.087	3.761±0.068	3.635±0.084	3.211±0.076	3.264±0.067	**3.118±0.062**
KIN8NM	0.259±0.006	0.254±0.005	0.246±0.004	0.252±0.003	0.243±0.003	0.244±0.003	0.226±0.004	**0.223±0.003**
NAVAL	0.015±0.000	0.016±0.000	0.013±0.000	0.013±0.000	0.010±0.000	**0.009±0.000**	0.013±0.000	0.012±0.000
POW. PLANT	5.312±0.108	5.068±0.082	5.812±0.119	5.423±0.111	5.377±0.133	4.974±0.078	5.229±0.097	**4.800±0.074**
WINE	0.945±0.044	0.809±0.011	0.730±0.011	0.748±0.008	0.752±0.014	0.746±0.009	**0.740±0.011**	0.749±0.010
YACHT	5.388±0.339	4.508±0.158	7.381±0.309	6.583±0.264	7.192±0.280	6.371±0.204	4.644±0.237	**3.211±0.120**

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

Experiments – Reinforcement Learning

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



(a) CartPoleSwingup (b) MountainCar (c) DoublePendulum

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

while not converged **do**

$\boldsymbol{\theta} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $\boldsymbol{\sigma} \leftarrow 1/\sqrt{N\mathbf{s} + \lambda}$

Randomly sample a data example $\mathcal{D}_i$

$\mathbf{g} \leftarrow -\nabla \log p(\mathcal{D}_i|\boldsymbol{\theta})$

$\mathbf{m} \leftarrow \gamma_1\, \mathbf{m} + (1-\gamma_1)\ (\mathbf{g} + \lambda\boldsymbol{\mu}/N)$

$\mathbf{s} \leftarrow \gamma_2\, \mathbf{s} + (1-\gamma_2)\, (\mathbf{g} \circ \mathbf{g})$

$\hat{\mathbf{m}} \leftarrow \mathbf{m}/(1-\gamma_1^t), \quad \hat{\mathbf{s}} \leftarrow \mathbf{s}/(1-\gamma_2^t)$

$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - \alpha\ \hat{\mathbf{m}}/(\sqrt{\hat{\mathbf{s}}} + \lambda/N)$

$t \leftarrow t + 1$

end while

Experiments – Logistic Regression

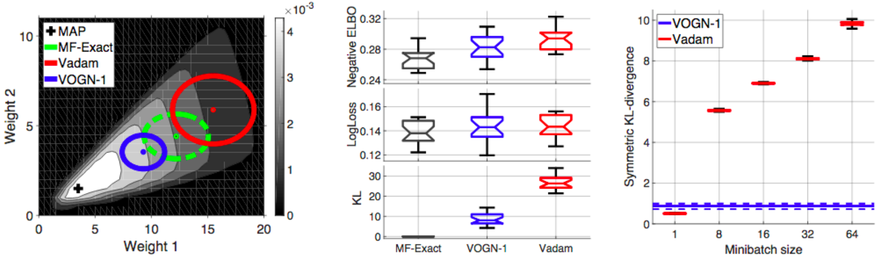


Figure 2. Experiments on Bayesian logistic regression showing (a) posterior approximations on a toy example, (b) performance on 'USPS-3v5' measuring negative ELBO, log-loss, and the symmetric KL divergence of the posterior approximation to MF-Exact, (c) symmetric KL divergence of Vadam for various minibatch 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
- *An Information Geometry of Statistical Manifold Learning*, Sun & Marchand-Maillet, 2014
- *Koszul Information Geometry and Souriau Geometric Temperature/Capacity of Lie Group Thermodynamics*, Barbaresco, 2014

Recommended Readings

1. New Perspectives and Insights on The Natural Gradient, Martens, 2014.
2. Objective Improvement in Information-Geometric Optimization, Ollivier, 2013. (Youtube)
3. Information Geometry for Neural Networks, Wagenaar, 1998.

Questions

Another relation to ELBO ?

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

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

1. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. (2016)
2. Ollivier, Y., Arnold, L., Auger, A., Hansen, N.: Information-Geometric optimization algorithms: A unifying picture via invariance principles. J. Mach. Learn. Res. 18, 1–65 (2017)
3. Martens, J.: New insights and perspectives on the natural gradient method. (2014)
4. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine. (2013)
5. Opper, M., Archambeau, C.: The variational gaussian approximation revisited. Neural Comput. 21, 786–792 (2009)