Kernelized Wasserstein Natural Gradient

Michael Arbel [1] Arthur Gretton [1] Wuchen Li [2] Guido Montufar [2,3]

[1]Gatsby Computational Neuroscience Unit, UCL, London

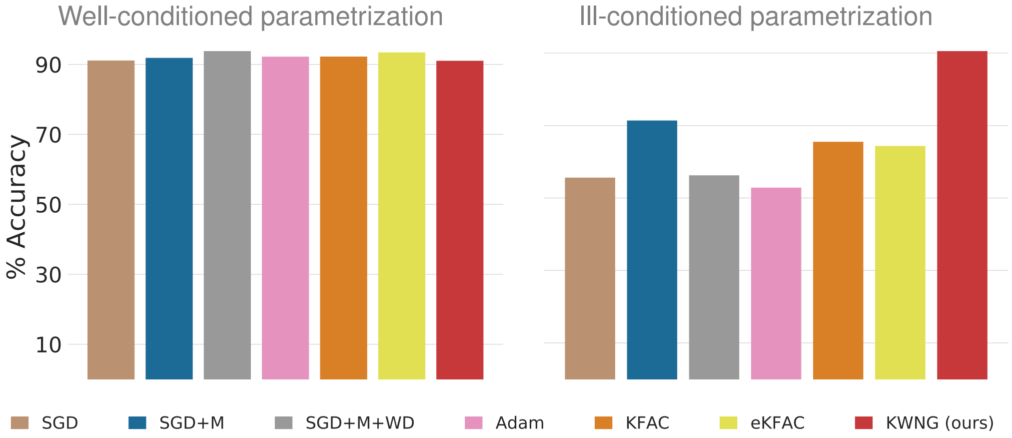
[2]University of California, Los Angeles

[3]Max Planck Institute for Mathematics in the Sciences, Leipzig

April 9, 2020

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

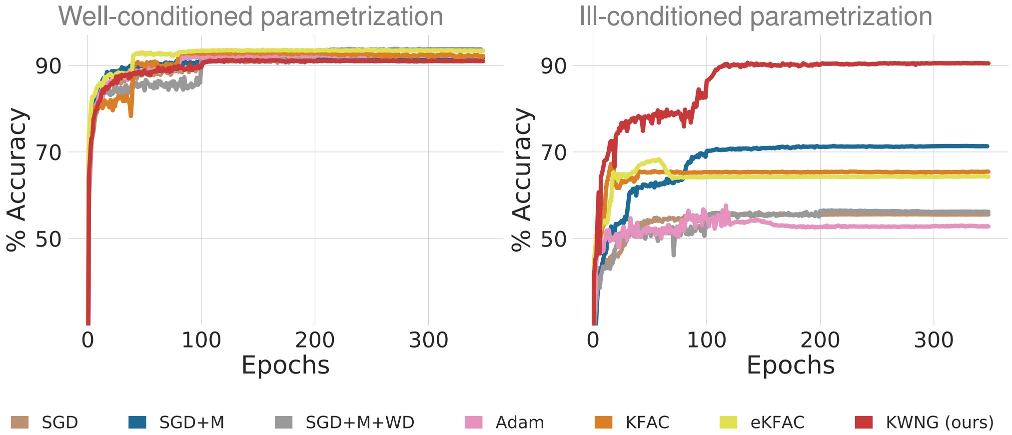
✓ Approximately Invariant to re-parametrization



Cifar10 classification task using ResNet-18 networks.

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

✓ Approximately Invariant to re-parametrization

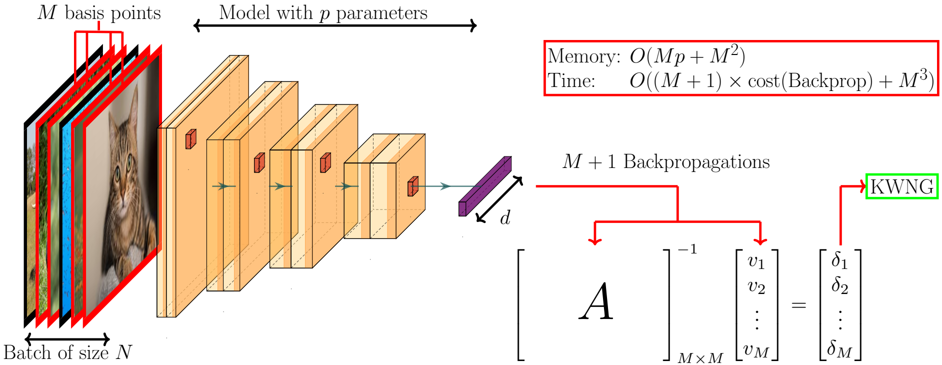


Cifar10 classification task using ResNet-18 networks.

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

✓ Approximately invariant to re-parametrization

✓ Fast and scalable



KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

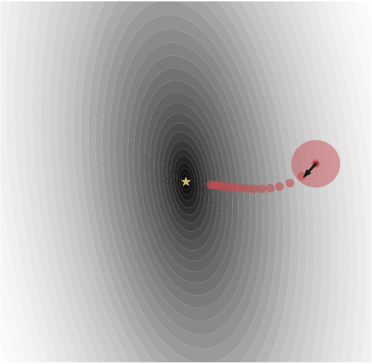
- ✓ Approximately invariant to re-parametrization
- ✓ Fast and scalable
- ✓ Can be used as a drop-in optimizer

```
from kwng import KWNG,  KWNGWrapper
from gaussian import Gaussian
kernel = Gaussian()
KWNGEstimator = KWNG (kernel,
                         num_basis= 10,
                         eps= 1e-4 )
w_optimizer = KWNGWrapper(optimizer,
                        criterion,
                        net,
                        KWNGEstimator)
loss,pred = w_optimizer.step(inputs, targets)
```

Euclidean Gradient

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(p_{\theta_k})^\top u + \frac{1}{2}\|u\|^2$$

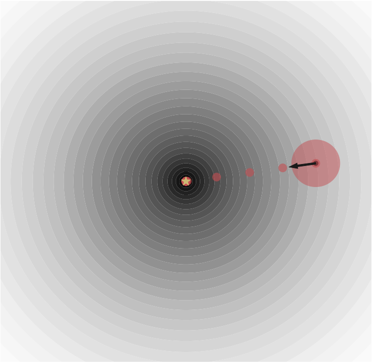


Euclidean Gradient

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(\rho_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(\rho_{\theta_k})^\top u + \frac{1}{2}\|u\|^2$$

- Different re-parametrization: $\psi = s(\theta)$



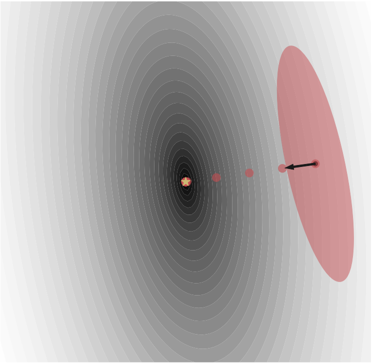
Fisher Natural Gradient

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(\rho_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(\rho_{\theta_k})^\top u + \frac{1}{2} \underbrace{u^\top G_F(\theta_k) u}_{\approx KL(p_{\theta_k} || p_{\theta_k+u})}$$

- Fisher information matrix:

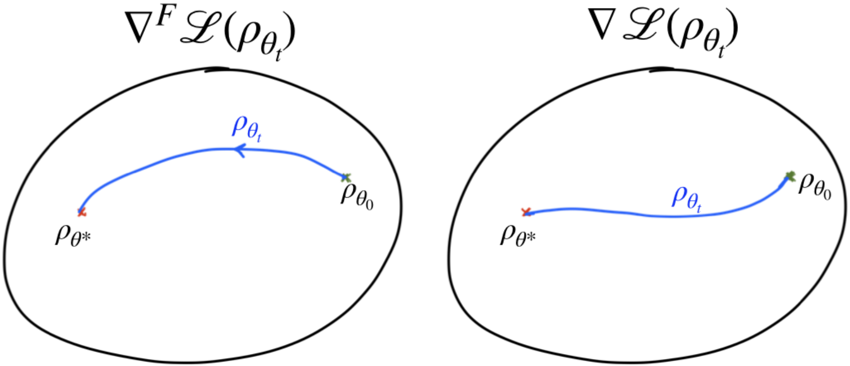
$$G_F(\theta) = \mathbb{E}_{\rho_\theta}\left[\nabla_\theta \log(\rho_\theta)(X) \nabla_\theta \log(\rho_\theta)(X)^\top\right]$$



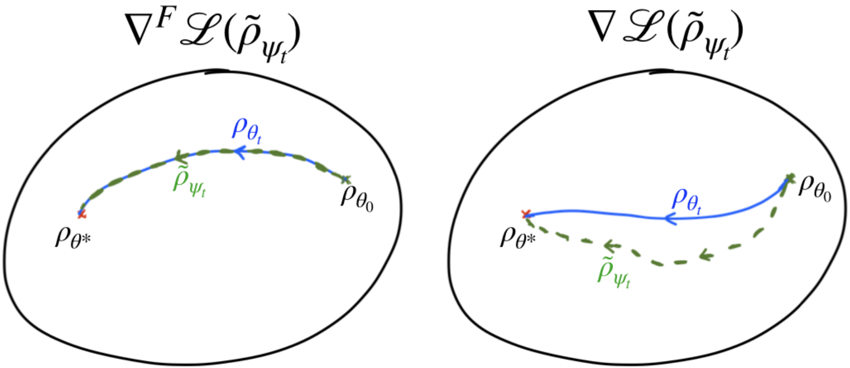
Pros:

- Invariant to parametrization

Invariance to re-parametrization



Invariance to re-parametrization



- Re-parametrization: $\psi = \Psi(\theta)$ and write $\tilde{\rho}_\psi = \rho_\theta$.
- Invariance to re-parametrization: $\Rightarrow \psi_t = \Psi(\theta_t)$

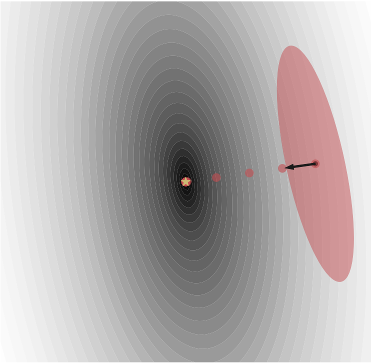
Fisher Natural Gradient

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(\rho_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(\rho_{\theta_k})^\top u + \frac{1}{2} \underbrace{u^\top G_F(\theta_k) u}_{\approx KL(p_{\theta_k} || p_{\theta_k + u})}$$

- Fisher information matrix:

$$G_F(\theta) = \mathbb{E}_{\rho_\theta}\left[\nabla_\theta \log(\rho_\theta)(X) \nabla_\theta \log(\rho_\theta)(X)^\top\right]$$



Pros:

- Invariant to parametrization

Cons:

- Not scalable, but efficient approximations exist: [Martens and Grosse, 2015, Grosse and Martens, 2016]
- Ill-suited for implicit models:

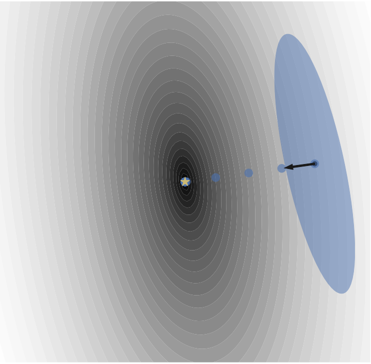
$$X \sim \rho_\theta \iff X = h_\theta(Z), \qquad Z \sim \nu$$

Wasserstein Natural Gradient [Li and Montufar, 2018]

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(p_{\theta_k})^\top u + \frac{1}{2} \underbrace{u^\top G_W(\theta_k) u}_{\approx W_2^2(p_{\theta_k}, p_{\theta_k+u})}$$

- Wasserstein information matrix: $G_W(\theta)$



Pros:

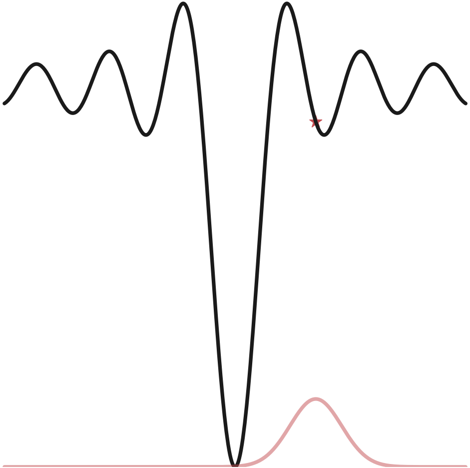
- Invariant to parametrization
- Works with implicit model
- Scalable approximation

Cons:

- ~~Not scalable~~
- ~~Ill-suited for implicit models:~~

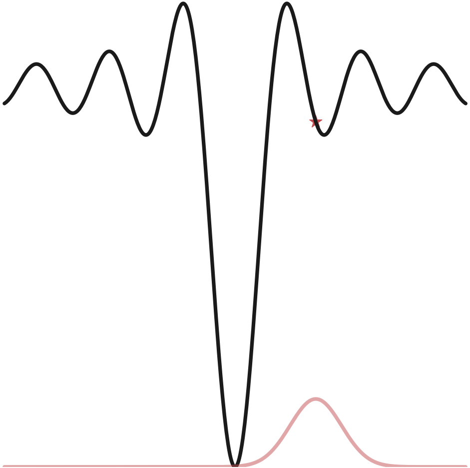
Wasserstein Natural Gradient: The Gaussian Family

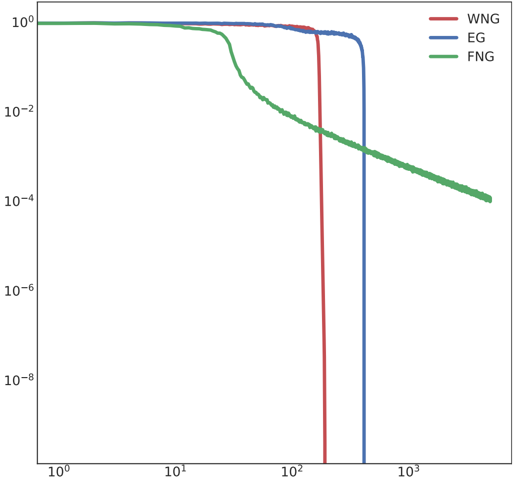
$$\mathcal{L}(\mu, \Sigma) := \int f(x)\mathcal{N}(x, \mu, \Sigma)dx$$



Wasserstein Natural Gradient: The Gaussian Family

$$\mathcal{L}(\mu, \Sigma) := \int f(x)\mathcal{N}(x, \mu, \Sigma)dx$$



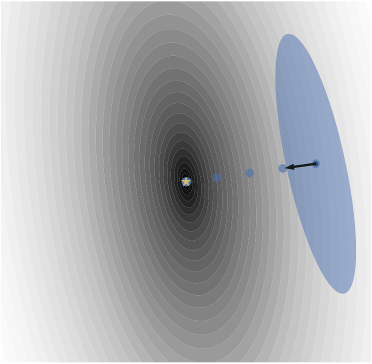


Wasserstein Natural Gradient [Li and Montufar, 2018]

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(p_{\theta_k})^\top u + \frac{1}{2} \underbrace{u^\top G_W(\theta_k) u}_{\approx W_2^2(p_{\theta_k}, p_{\theta_k+u})}$$

- Wasserstein information matrix: $G_W(\theta)$



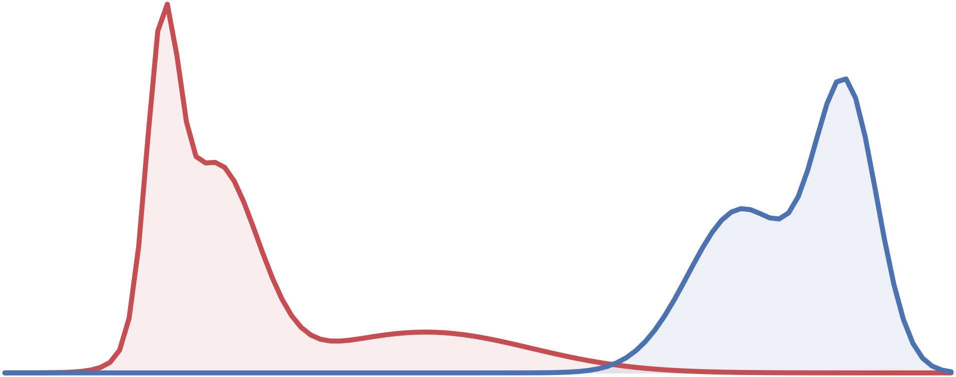
Pros:

- Invariant to parametrization
- Works with implicit model
- Scalable approximation

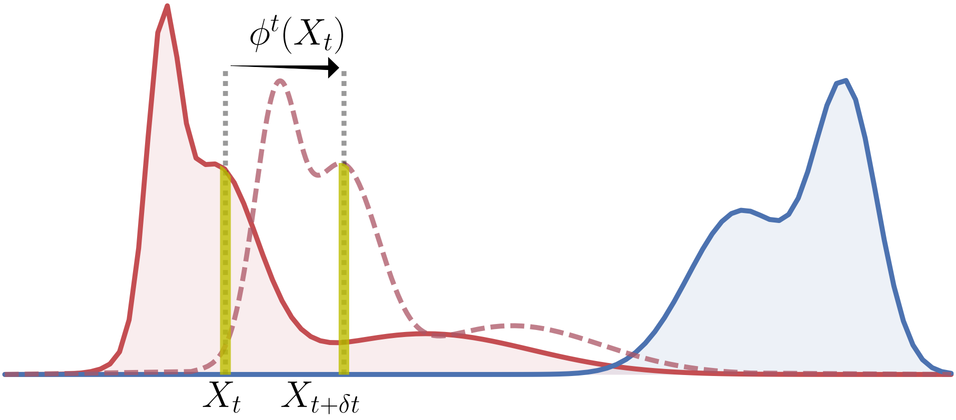
Cons:

- ~~Not scalable~~
- ~~Ill-suited for implicit models:~~

Dynamic formulation of the Wasserstein distance



Dynamic formulation of the Wasserstein distance



$$W_2^2(p,q) := \inf_{(\rho_t,\phi_t)} \int_0^1 \int \|\phi^t(x)\|^2 \, \mathrm{d}\rho_t(x) dt, \quad \partial_t \rho_t + div(\rho_t \phi^t) = 0$$

Dynamic formulation of the Wasserstein distance

- The Wasserstein distance as a geodesic distance [Benamou and Brenier, 2000]

$$W_2^2(p,q) := \inf_{(\rho_t,\phi_t)} \int_0^1 \int \|\phi^t(x)\|^2 \,\mathrm{d}\rho_t(x)dt, \quad \partial_t \rho_t + div(\rho_t \phi^t) = 0$$

Dynamic formulation of the Wasserstein distance

- The Wasserstein distance as a geodesic distance [Benamou and Brenier, 2000]

$$W_2^2(p,q) := \inf_{(\rho_t,\phi_t)} \int_0^1 \int \|\phi^t(x)\|^2 \,\mathrm{d}\rho_t(x) dt, \quad \partial_t \rho_t + div(\rho_t \phi^t) = 0$$

- Wasserstein metric:

$$g_\rho(\delta,\delta) := \int \|\phi(x)\|^2 \,\mathrm{d}\rho(x), \quad \delta + div(\rho\phi) = 0.$$

Dynamic formulation of the Wasserstein distance

- The Wasserstein distance as a geodesic distance [Benamou and Brenier, 2000]

$$W_2^2(p,q) := \inf_{(\rho_t,\phi_t)} \int_0^1 \int \|\phi^t(x)\|^2 \, \mathrm{d}\rho_t(x) dt, \quad \partial_t \rho_t + div(\rho_t \phi^t) = 0$$

- Wasserstein metric:

$$g_\rho(\delta,\delta) := \int \|\phi(x)\|^2 \, \mathrm{d}\rho(x), \quad \delta + div(\rho\phi) = 0.$$

- Wasserstein Information matrix:

$$u^\top G_W(\theta) u := g_{\rho_\theta}(\nabla_\theta \rho_\theta^\top u, \nabla_\theta \rho_\theta^\top u) = \int \|\phi(x)\|^2 d\rho_\theta(x)$$

$$\nabla_\theta \rho_\theta^\top u + div(\rho_\theta \phi) = 0.$$

The triple tricks

- The duality trick: Variational expression for elliptic equations:

$$\nabla \rho_\theta^\top u + div(\rho_\theta \phi_u) = 0$$

$$\Downarrow$$

$$\sup_{f \in C_c^\infty(\Omega)} \nabla_\theta \mathbb{E}_{\rho_\theta}[f(X)]^\top u - \frac{1}{2}\mathbb{E}_{\rho_\theta}\left[\|\nabla f(X)\|^2\right]$$

The triple tricks

- The duality trick: Variational expression for elliptic equations:

$$\nabla \rho_\theta^\top u + div(\rho_\theta \phi_u) = 0$$

$$\Downarrow$$

$$\frac{1}{2} u^\top G_W(\theta) u = \frac{1}{2} \int \|\phi\|^2 d\rho_\theta = \sup_{f \in C_c^\infty(\Omega)} \nabla_\theta \mathbb{E}_{\rho_\theta}[f(X)]^\top u - \frac{1}{2} \mathbb{E}_{\rho_\theta}\left[\|\nabla f(X)\|^2\right]$$

The triple tricks

- The duality trick: Variational expression for elliptic equations:

$$\nabla \rho_\theta^\top u + div(\rho_\theta \phi_u) = 0$$

$$\Downarrow$$

$$\frac{1}{2} u^\top G_W(\theta) u = \frac{1}{2} \int \|\phi\|^2 d\rho_\theta = \sup_{f \in C_c^\infty(\Omega)} \nabla_\theta \mathbb{E}_{\rho_\theta}[f(X)]^\top u - \frac{1}{2} \mathbb{E}_{\rho_\theta}\left[\|\nabla f(X)\|^2\right]$$

- The reparametrization trick:

$$\nabla_\theta \mathbb{E}_{\rho_\theta}[f(X)]^\top u = \mathbb{E}_\eta \left[\nabla_\theta f(g_\theta(Z))\right]^\top u, \qquad X = g_\theta(Z), \quad Z \sim \eta$$

The triple tricks

- The duality trick: Variational expression for elliptic equations:

$$\nabla \rho_\theta^\top u + div(\rho_\theta \phi_u) = 0$$

$$\Downarrow$$

$$\frac{1}{2} u^\top G_W(\theta) u = \frac{1}{2} \int \|\phi\|^2 d\rho_\theta = \sup_{f \in C_c^\infty(\Omega)} \nabla_\theta \mathbb{E}_{\rho_\theta}[f(X)]^\top u - \frac{1}{2} \mathbb{E}_{\rho_\theta} \left[\|\nabla f(X)\|^2 \right]$$

- The reparametrization trick:

$$\nabla_\theta \mathbb{E}_{\rho_\theta}[f(X)]^\top u = \mathbb{E}_\eta \left[\nabla_\theta f(g_\theta(Z))\right]^\top u, \qquad X = g_\theta(Z), \quad Z \sim \eta$$

- The kernel trick: Choose a nice kernel k and find solutions of the form:

$$\hat{f}(x) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, x) \subset \mathcal{H}_M$$

Saddle-point formulation

$$\min_u \nabla_\theta \mathcal{L}(p_\theta)^\top u + \frac{1}{2} u^\top G_W(\theta) u$$

$$\Downarrow$$

$$\min_u \sup_{f \in \mathcal{H}_M} \nabla_\theta \mathcal{L}(p_\theta)^\top u + \nabla_\theta \mathbb{E}_{p_\theta} [f(X)]^\top u - \frac{1}{2} \mathbb{E}_{p_\theta} \left[\|\nabla f(X)\|^2 \right]$$

- $\mathcal{H}_M$ contains functions of the form:

$$f(x) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, x)$$

Saddle-point formulation

$$\min_u \nabla_\theta \mathcal{L}(p_\theta)^\top u + \frac{1}{2} u^\top G_W(\theta) u + \overbrace{\frac{\epsilon}{2}\|u\|^2}^{\text{damping}}$$

$$\Downarrow$$

$$\min_u \sup_{f \in \mathcal{H}_M} \nabla_\theta \mathcal{L}(p_\theta)^\top u + \nabla_\theta \mathbb{E}_{p_\theta}[f(X)]^\top u - \frac{1}{2}\mathbb{E}_{p_\theta}\left[\|\nabla f(X)\|^2\right] + \overbrace{\frac{\epsilon}{2}\|u\|^2}^{\text{damping}}$$

- $\mathcal{H}_M$ contains functions of the form:

$$f(x) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, x)$$

Saddle-point formulation

$$\min_u \nabla_\theta \mathcal{L}(p_\theta)^\top u + \frac{1}{2} u^\top G_W(\theta) u + \frac{\epsilon}{2} \|u\|^2$$

$$\Downarrow$$

$$\sup_{f \in \mathcal{H}_M} \min_u \nabla_\theta \mathcal{L}(p_\theta)^\top u + \nabla_\theta \mathbb{E}_{p_\theta}[f(X)]^\top u - \frac{1}{2}\mathbb{E}_{p_\theta}\left[\|\nabla f(X)\|^2\right] + \frac{\epsilon}{2}\|u\|^2$$

- $\mathcal{H}_M$ contains functions of the form:

$$f(x) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, x)$$

- Optimal $f^\star$ obtained by solving a quadratic problem of size M in $(\alpha_1, ..., \alpha_M)$
- Wasserstein natural descent direction:

$$\widehat{\mathcal{D}}_k = -\frac{1}{\epsilon}\left(\nabla_\theta \mathcal{L}(p_{\theta_k}) + \nabla_\theta \mathbb{E}_{p_{\theta_k}}[f^\star(X)]\right)$$

Infinitely many features with kernels!

- Kernel: "similarity" function $k(x, y) \in \mathbb{R}$
  - e.g. gaussian kernel

$$k(x, y) = exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$$

Infinitely many features with kernels!

- Kernel: "similarity" function $k(x, y) \in \mathbb{R}$
  - e.g. gaussian kernel

$$k(x, y) = exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$$

- Reproducing kernel Hilbert space $\mathcal{H}$ contains functions of the form:

$$f(y) = \sum_{m}^{M} \alpha_m k(X_m, y), \qquad f(y) = \sum_{m}^{M} \alpha_m \partial_{i_m} k(X_m, y)$$

Infinitely many features with kernels!

- Kernel: "similarity" function $k(x, y) \in \mathbb{R}$
  - e.g. gaussian kernel

$$k(x, y) = exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$$

- Reproducing kernel Hilbert space $\mathcal{H}$ contains functions of the form:

$$f(y) = \sum_m^M \alpha_m k(X_m, y), \qquad f(y) = \sum_m^M \alpha_m \partial_{i_m} k(X_m, y)$$

- But $\mathcal{H}$ is much bigger: can be dense on $C_b(\Omega)$.

Infinitely many features with kernels!

- Kernel: "similarity" function $k(x, y) \in \mathbb{R}$
  - e.g. gaussian kernel

$$k(x, y) = exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$$

- Reproducing kernel Hilbert space $\mathcal{H}$ contains functions of the form:

$$f(y) = \sum_m^M \alpha_m k(X_m, y), \qquad f(y) = \sum_m^M \alpha_m \partial_{i_m} k(X_m, y)$$

- But $\mathcal{H}$ is much bigger: can be dense on $C_b(\Omega)$.
- Reproducing property:

$$f(y) = \langle f, k(x, .)\rangle_{\mathcal{H}}$$

Infinitely many features with kernels!

- Kernel: "similarity" function $k(x,y) \in \mathbb{R}$
  - e.g. gaussian kernel

$$k(x,y) = exp(-\frac{1}{2\sigma^2}\|x-y\|^2)$$

- Reproducing kernel Hilbert space $\mathcal{H}$ contains functions of the form:

$$f(y) = \sum_{m}^{M} \alpha_m k(X_m, y), \qquad f(y) = \sum_{m}^{M} \alpha_m \partial_{i_m} k(X_m, y)$$

- But $\mathcal{H}$ is much bigger: can be dense on $C_b(\Omega)$.
- Reproducing property:

$$f(y) = \langle f, k(x,.)\rangle_{\mathcal{H}}$$

- Inner product $\langle .,. \rangle_{\mathcal{H}}$ defined implicitly using k:
  - $\langle k(x,.), k(y,.)\rangle_{\mathcal{H}} = k(x,y)$

Representer Theorem

- General Loss function of the form:

$$L(f) = \int \mathcal{R}((\partial_i f(x))_{1 \le i \le d}, y) dp(x, y) + \frac{1}{2} \lambda \|f\|_{\mathcal{H}}^2$$

Representer Theorem

- General Loss function of the form:

$$L(f) = \int \mathcal{R}((\partial_i f(x))_{1 \leq i \leq d}, y) dp(x, y) + \frac{1}{2} \lambda \|f\|_{\mathcal{H}}^2$$

- Empirical version using samples (X_n, Y_n):

$$\hat{L}(f) = \frac{1}{N} \sum_{n}^{N} \mathcal{R}(\partial_i f(X_n), Y_n) + \frac{1}{2} \lambda \|f\|_{\mathcal{H}}^2$$

Representer Theorem

- General Loss function of the form:

$$L(f) = \int \mathcal{R}((\partial_i f(x))_{1\leq i\leq d}, y)dp(x,y) + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}}$$

- Empirical version using samples (X_n, Y_n):

$$\hat{L}(f) = \frac{1}{N}\sum_{n}^{N} \mathcal{R}(\partial_i f(X_n), Y_n) + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}}$$

- Representer theorem says: Optimal empirical solution of the form:

$$f^\star(y) = \sum_{n,i} \alpha_{n,i}\partial_i k(X_n, y)$$

Representer Theorem

- General Loss function of the form:

$$L(f) = \int \mathcal{R}((\partial_i f(x))_{1 \le i \le d}, y) dp(x, y) + \frac{1}{2}\lambda \|f\|_{\mathcal{H}}^2$$

- Empirical version using samples (X_n, Y_n):

$$\hat{L}(f) = \frac{1}{N}\sum_{n}^{N} \mathcal{R}(\partial_i f(X_n), Y_n) + \frac{1}{2}\lambda \|f\|_{\mathcal{H}}^2$$

- Representer theorem says: Optimal empirical solution of the form:

$$f^{\star}(y) = \sum_{n,i} \alpha_{n,i} \partial_i k(X_n, y)$$

- Only need to find α: solve finite dimensional optimization problem.

Representer Theorem and Nystrom Methods

- Optimal empirical solution of the form:

$$f^{\star}(y) = \sum_{n,i} \alpha_{n,i} \partial_i k(X_n, y)$$

- Expensive to compute $\alpha_{n,i}$: cost in time $O(N^3 d^3)$ for quadratic loss
- Nystrom method [1]: Reduce computational cost:

$$\hat{f}_M^*(y) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, y)$$

[1] [Rudi et al., 2015, Sutherland et al., 2017]

Representer Theorem and Nystrom Methods

- Optimal empirical solution of the form:

$$f^\star(y) = \sum_{n,i} \alpha_{n,i} \partial_i k(X_n, y)$$

- Expensive to compute $\alpha_{n,i}$: cost in time $O(N^3 d^3)$ for quadratic loss
- Nystrom method [1]: Reduce computational cost:

$$\hat{f}^*_M(y) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, y)$$

M sub-samples from $(X_i)_{1 \le i \le N}$

[1] [Rudi et al., 2015, Sutherland et al., 2017]

Representer Theorem and Nystrom Methods

- Optimal empirical solution of the form:

$$f^{\star}(y) = \sum_{n,i} \alpha_{n,i} \partial_i k(X_n, y)$$

- Expensive to compute $\alpha_{n,i}$: cost in time $O(N^3 d^3)$ for quadratic loss
- Nystrom method [1]: Reduce computational cost:

$$\hat{f}^*_M(y) = \sum_{m=1}^{M} \alpha_m \partial_{i_m} k(X_m, y)$$

Randomly sampled from $\{1, ..., d\}$ (for i_m)

M sub-samples from $(X_i)_{1 \le i \le N}$ (for X_m)

[1] [Rudi et al., 2015, Sutherland et al., 2017]

KWNG: Sample based version

- After some further calculations:

$$\nabla^W \mathcal{L}(\theta) \approx \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \lambda\epsilon K + \epsilon CC^\top)^\dagger T\right) \nabla \mathcal{L}(\theta)$$

- Similar to a Woodbury matrix identity

KWNG: Sample based version

- After some further calculations:

$$T := \nabla\tau(\theta) \text{ with } \tau(\theta)_m = \frac{1}{N}\sum_{n=1}^{N} \partial_{i_m} k(X_m, h_\theta(Z_n))$$

$$\nabla^W \mathcal{L}(\theta) \approx \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \lambda\epsilon K + \epsilon C C^\top)^\dagger T\right) \nabla\mathcal{L}(\theta)$$

- Similar to a Woodbury matrix identity

KWNG: Sample based version

- After some further calculations:

$$T := \nabla\tau(\theta) \text{ with } \tau(\theta)_m = \frac{1}{N}\sum_{n=1}^{N} \partial_{i_m} k(X_m, h_\theta(Z_n))$$

$$\nabla^W \mathcal{L}(\theta) \approx \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \lambda\epsilon K + \epsilon C C^\top)^\dagger T\right)\nabla\mathcal{L}(\theta)$$

$$K_{m,m'} = \partial_{i_m}\partial_{i_{m'}+d} k(X_m, X_{m'})$$

- Similar to a Woodbury matrix identity

KWNG: Sample based version

- After some further calculations:

$$T := \nabla \tau(\theta) \text{ with } \tau(\theta)_m = \frac{1}{N}\sum_{n=1}^{N} \partial_{i_m} k(X_m, h_\theta(Z_n))$$

$$\nabla^W \mathcal{L}(\theta) \approx \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \lambda\epsilon K + \epsilon C C^\top)^\dagger T\right) \nabla \mathcal{L}(\theta)$$

$$K_{m,m'} = \partial_{i_m}\partial_{i_{m'}+d} k(X_m, X_{m'})$$

$$C_{m,(n,i)} = \frac{1}{\sqrt{N}}\partial_{i_m}\partial_{i+d} k(X_m, X_n)$$

- Similar to a Woodbury matrix identity

Theory

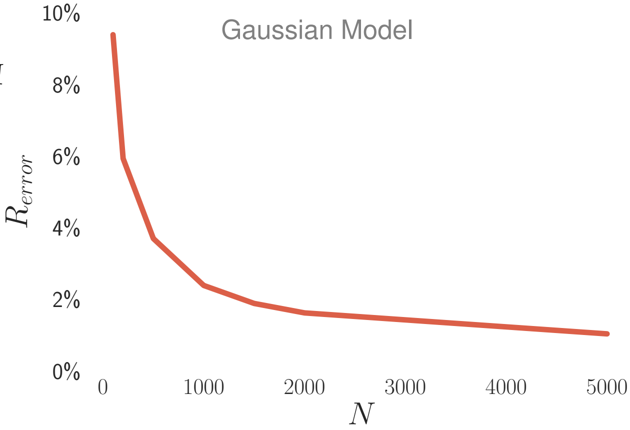
How small M can be and still be sure it works?

- Need fewer basis points M than data points N

$$M \approx \sqrt{N}$$

- Relative error decreases with more data ($N \to +\infty$)

$$R_{error} \sim \frac{1}{N^{\frac{1}{4}}}$$



Theory: Consistency and convergence rates

Main assumption: Let ϕ_u be the solution to the PDE:

$$\nabla \rho_\theta^\top u + div(\rho_\theta \phi_u) = 0$$

For any precision $\kappa > 0$, there exists $f \in \mathcal{H}$:

$$\int \|\phi_u - \nabla f\|^2 d\rho_\theta \leq \kappa \qquad \|f\|_{\mathcal{H}} \leq C\kappa^{-c}$$

Theory: Consistency and convergence rates

Main assumption: Let ϕ_u be the solution to the PDE:

$$\nabla \rho_\theta^\top u + div(\rho_\theta \phi_u) = 0$$

For any precision $\kappa > 0$, there exists $f \in \mathcal{H}$:

$$\int \|\phi_u - \nabla f\|^2 d\rho_\theta \leq \kappa \qquad \|f\|_{\mathcal{H}} \leq C\kappa^{-c}$$

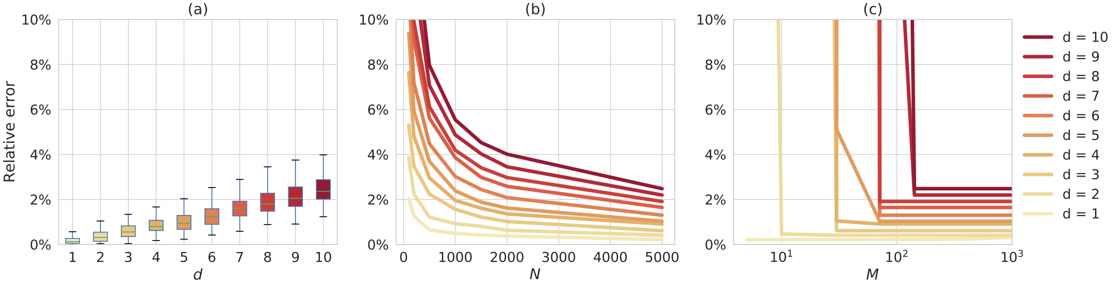
Theorem

Let δ be such that $0 \leq \delta \leq 1$. Under additional mild assumptions, for N large enough, $M \sim (d\mathrm{N}^{\frac{2+c}{4+c}} \log(\mathrm{N}))$, $\lambda \sim \mathrm{N}^{\frac{2+c}{4+c}}$ and $\epsilon \lesssim \mathrm{N}^{-\frac{1}{4+c}}$, it holds with probability at least $1 - \delta$ that:

$$\|\widehat{\nabla^W \mathcal{L}(\theta)} - \nabla^W \mathcal{L}(\theta)\|^2 = \mathcal{O}\left(N^{-\frac{2}{4+c}}\right).$$

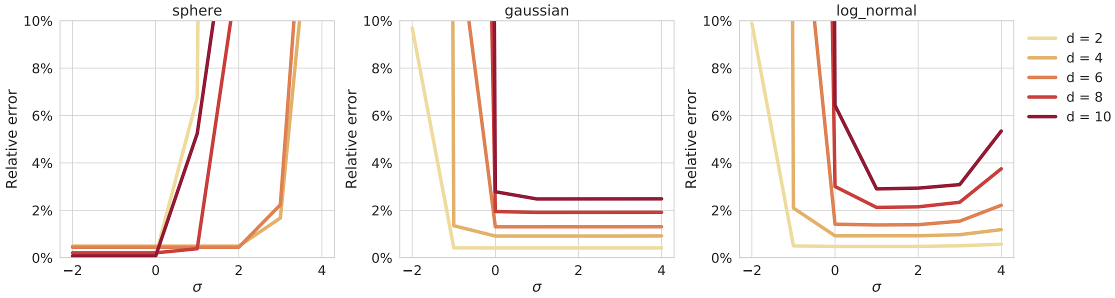
Experimental evaluation: Synthetic models

Gaussians: $X = \mu + \sigma^{\frac{1}{2}} Z, \qquad Z \sim \mathcal{N}(0, I)$



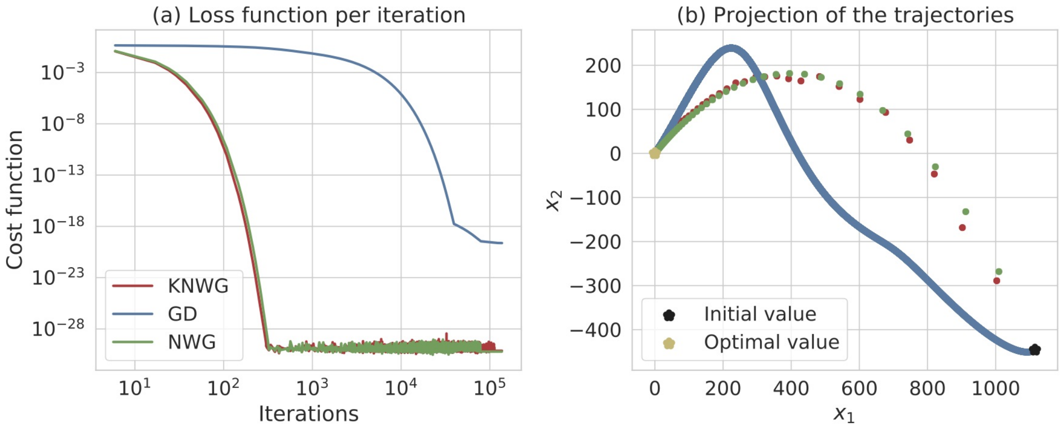
Experimental evaluation: Sensitivity to the choice of the kernel

- Gaussian kernel $k(x, y) = \exp(-\frac{\|x-y\|^2}{\sigma})$



Experimental evaluation: Optimization trajectory

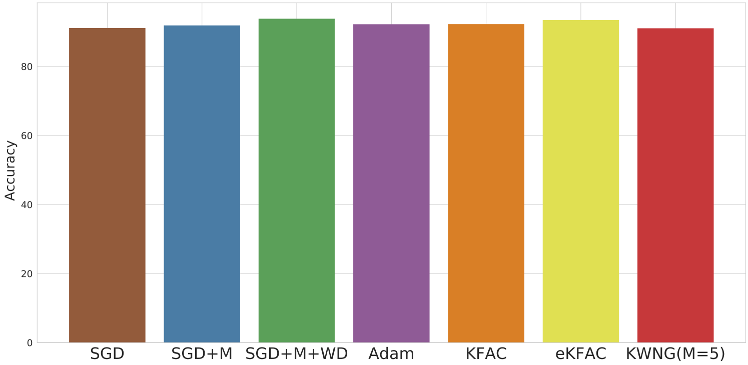
- Gaussian model for ρ_θ
- Loss functional $\mathcal{L}(\rho_\theta) = W_2^2(\rho_\theta, \rho_{\theta^*})$.



Experimental evaluation: Classification task

Well-conditioned problem:

$$\min_{\theta} \mathcal{L}(\rho_\theta) := \int \ell(h_\theta(Z), Y)\, \mathrm{d}\nu(Z, Y)$$

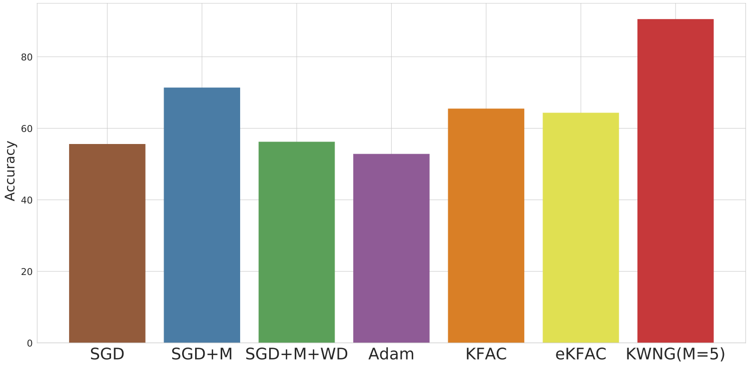


Experimental evaluation: Classification task

Ill-conditioned problem:

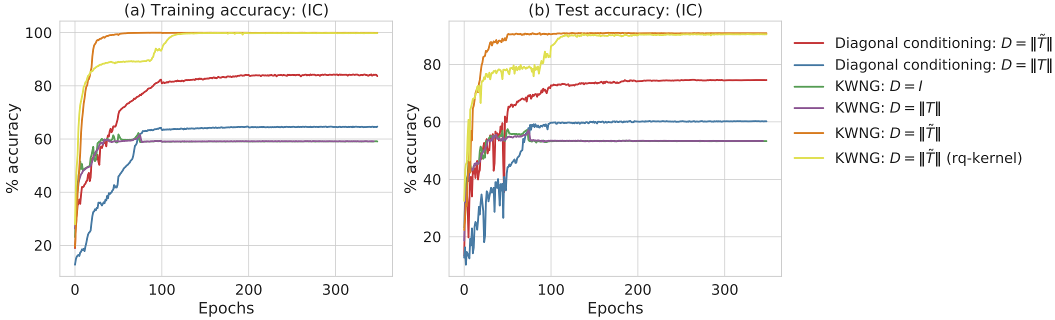
$$\min_{\theta} \mathcal{L}(\rho_\theta) := \int \ell(Uh_\theta(Z), Y)\, \mathrm{d}\nu(Z, Y)$$

U is a diagonal matrix with $\kappa = 10^7$



Ablation study

- Choice of the damping matrix $D(\theta)$
- Choice of the kernel (gaussian vs rational quadratic)



Conclusion

Summary of contributions

- Proposed to use Wasserstein natural gradient for ill-conditioned problems.
- A new algorithm to estimate the Wasserstein natural gradient
- Convergence rate: trade-off between computational complexity and statistical accuracy

Conclusion

Summary of contributions

- Proposed to use Wasserstein natural gradient for ill-conditioned problems.
- A new algorithm to estimate the Wasserstein natural gradient
- Convergence rate: trade-off between computational complexity and statistical accuracy

Limitation:

- Sensitive to the choice of the damping/regularization.
- Additional hyper-parameters to tune (kernel, basis points,...)
- Accuracy of the estimation quickly degrades with the dimension.
- Ridgeless estimator seems much more accurate in practice but no guarantees yet.

Future work:

- When can one clearly benefit from WNG: Natural Evolution Strategies [Wierstra et al., 2011]?
- Application to meta-learning: Can the Wasserstein be a good proximity measure between several tasks.
- Implicit Policy Optimization:
  - Useful for more complex action space [Tang and Agrawal, 2019]) (sequence of actions).
  - TRPO [Schulman et al., 2015] can't be used in this case, but WNG can.

Thank you !

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \lambda\epsilon K + \epsilon CC^\top)^\dagger T\right)\widehat{\nabla \mathcal{L}(\theta)}$$

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \epsilon CC^\top)^\dagger T\right)\widehat{\nabla \mathcal{L}(\theta)}$$

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \epsilon CC^\top)^\dagger T\right)\widehat{\nabla \mathcal{L}(\theta)}$$

- Chain rule for T:

$$T_m = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \partial_{i_m} k(Y_m, h_\theta(Z_n)) \implies T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - B^\top C^\top (CBB^\top C^\top + \epsilon CC^\top)^\dagger CB\right)\widehat{\nabla \mathcal{L}(\theta)}$$

- Chain rule for T:

$$T_m = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \partial_{i_m} k(Y_m, h_\theta(Z_n)) \implies T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - B^\top C^\top (CBB^\top C^\top + \epsilon CC^\top)^\dagger CB\right)\widehat{\nabla \mathcal{L}(\theta)}$$

- Chain rule for T:

$$T_m = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \partial_{i_m} k(Y_m, h_\theta(Z_n)) \implies T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

- 'Simplify' C by computing an SVD : $CC^\top = USU^\top$

$$\widetilde{T} = S^\dagger U^\top CB, \qquad P = S^\dagger S$$

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - \widetilde{T}^\top (\widetilde{T}\widetilde{T}^\top + \epsilon P)^\dagger \widetilde{T}\right) \widehat{\nabla \mathcal{L}(\theta)}$$

- Chain rule for T:

$$T_m = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \partial_{i_m} k(Y_m, h_\theta(Z_n)) \implies T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

- 'Simplify' C by computing an SVD : $CC^\top = USU^\top$

$$\widetilde{T} = S^\dagger U^\top CB, \qquad P = S^\dagger S$$

KWNG: Ridgeless version

- Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - \widetilde{T}^\top (\widetilde{T}\widetilde{T}^\top + \epsilon P)^\dagger \widetilde{T}\right) \widehat{\nabla \mathcal{L}(\theta)}$$

- Chain rule for T:

$$T_m = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \partial_{i_m} k(Y_m, h_\theta(Z_n)) \implies T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

- 'Simplify' C by computing an SVD : $CC^\top = USU^\top$

$$\widetilde{T} = S^\dagger U^\top CB, \qquad P = S^\dagger S$$

- No consistency result for the Ridgeless estimator yet.

KWNG: Ridgeless version

Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon} \left(I - T^\top (TT^\top + \epsilon CC^\top)^\dagger T \right) \widehat{\nabla \mathcal{L}(\theta)}$$

KWNG: Ridgeless version

Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \epsilon CC^\top)^\dagger T\right)\widehat{\nabla \mathcal{L}(\theta)}$$

$$T_m = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \partial_{i_m} k(Y_m, h_\theta(Z_n))$$

KWNG: Ridgeless version

Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - T^\top (TT^\top + \epsilon CC^\top)^\dagger T\right)\widehat{\nabla \mathcal{L}(\theta)}$$

$$T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

KWNG: Ridgeless version

Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - B^\top C^\top (CBB^\top C^\top + \epsilon CC^\top)^\dagger CB\right) \widehat{\nabla \mathcal{L}(\theta)}$$

$$T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

KWNG: Ridgeless version

Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon} \left(I - B^\top C^\top (CBB^\top C^\top + \epsilon CC^\top)^\dagger CB \right) \widehat{\nabla \mathcal{L}(\theta)}$$

$$T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

'Simplify' C:

$$\widetilde{T} = S^\dagger U^\top T, \qquad P = S^\dagger S$$

where $CC^\top = USU^\top$

KWNG: Ridgeless version

Additional structure when $\lambda = 0$:

$$\widehat{\nabla^W \mathcal{L}(\theta)} = \frac{1}{\epsilon}\left(I - \widetilde{T}^\top (\widetilde{T}\widetilde{T}^\top + \epsilon P)^\dagger \widetilde{T}\right)\widehat{\nabla \mathcal{L}(\theta)}$$

$$T = CB, \qquad B_n = \nabla_\theta h_\theta(Z_n)$$

'Simplify' C:

$$\widetilde{T} = S^\dagger U^\top T, \qquad P = S^\dagger S$$

where $CC^\top = USU^\top$

Benamou, J.-D. and Brenier, Y. (2000).
A computational fluid mechanics solution to the monge-kantorovich mass transfer problem.
Numerische Mathematik, 84(3):375–393.

Grosse, R. and Martens, J. (2016).
A Kronecker-factored Approximate Fisher Matrix for Convolution Layers.
In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, pages 573–582. JMLR.org.
event-place: New York, NY, USA.

Li, W. and Montufar, G. (2018).
Natural gradient via optimal transport.
arXiv:1803.07033 [cs, math].
arXiv: 1803.07033.

Martens, J. and Grosse, R. (2015).
Optimizing Neural Networks with Kronecker-factored Approximate Curvature.
arXiv:1503.05671 [cs, stat]