Variational Optimization

James Gleeson Eric Langlois
William Saunders

January 26, 2018

Variational Optimization

- Goal: Maximize a function.

$$\max_x f(z)$$

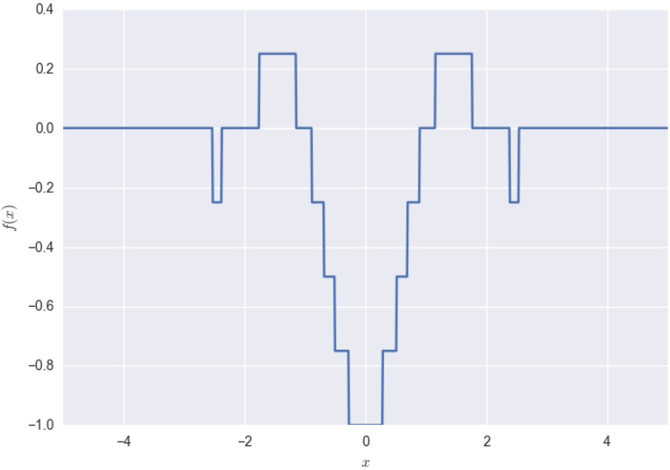
- Instead: Put a parameterized distribution $\pi(z|\theta)$ over z and maximize the expectation.

$$\max_\theta \mathbb{E}_{\pi(z|\theta)}[f(z)]$$

- This is a lower bound on the pointwise maximum:

$$J(\theta) = \mathbb{E}_{\pi(z|\theta)}[f(z)] \leq \max_x f(z)$$

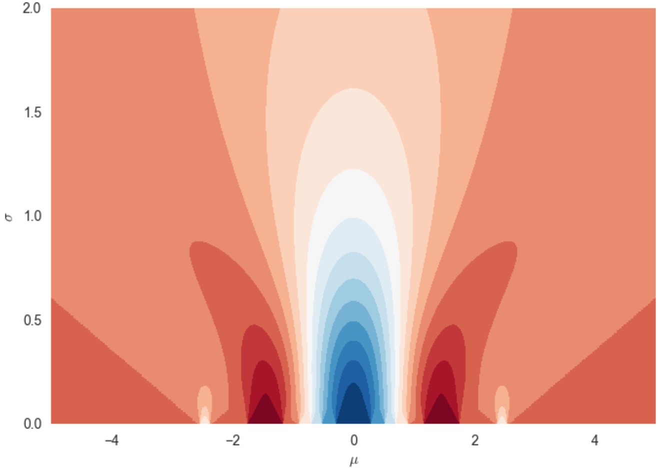
Example: Discrete Sinc



Original objective function.

Ferenc Huszár. http://www.inference.vc/evolution-strategies-variational-optimisation-and-natural-es-2/

Example: Discrete Sinc



Variational optimization landscape: differnetiable and in this case no local optima.

Ferenc Huszár. http://www.inference.vc/evolution-strategies-variational-optimisation-and-natural-es-2/

Natural Evolution Strategies

Wierstra, Schaul, Glasmachers, Sun, Peters, Schmidhuber

Variational Optimization using REINFORCE

- The objective function:

$$J(\theta) = \mathbb{E}_{\pi(z|\theta)}[f(z)]$$

- REINFORCE gradient:

$$\begin{aligned} \nabla_\theta J &= \mathbb{E}_{\pi(z|\theta)}[f(z)\nabla_\theta \log \pi(z|\theta)] \\ &\approx \frac{1}{N}\sum_{k=1}^{N} f(z_k)\nabla_\theta \log \pi(z_k|\theta) \end{aligned}$$

- Perform gradient ascent:

$$\theta \leftarrow \theta + \eta \nabla_\theta J$$

Canonical Search Gradient Algorithm

input : f, θ_{init}
repeat
  for $k = 1, \ldots, N$ **do**
    draw samples $z_k \sim \pi(\cdot, \theta)$
    evaluate the fitness $f(z_k)$
    calculate log-derivatives $\nabla_\theta \log \pi(z_k|\theta)$
  end
  $\nabla_\theta J \leftarrow \frac{1}{N} \sum_{k=1}^{N} \nabla_\theta \log \pi(z_k|\theta) \cdot f(z_k)$
  $\theta \leftarrow \theta + \eta \cdot \nabla_\theta J$
until *stopping condition is met*;

Multivariate Random Normal Distribution

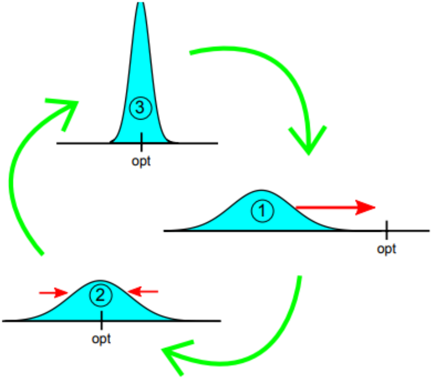
- Gradients:

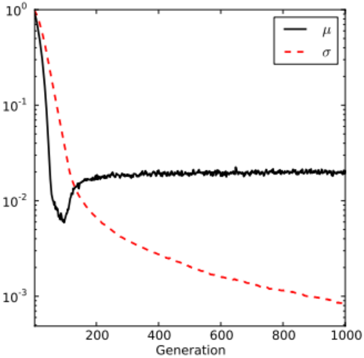
$$\nabla_\mu \log \pi(z|\theta) = \Sigma^{-1}(z-\mu)$$
$$\nabla_\Sigma \log \pi(z|\theta) = \frac{1}{2}\Sigma^{-1}(z-\mu)(z-\mu)^T\Sigma^T - \frac{1}{2}\Sigma^{-1}$$

- Updates for 1D Case:

$$\nabla_\mu J = \frac{z-\mu}{\sigma^2} \qquad \propto \frac{1}{\sigma}$$
$$\nabla_\sigma J = \frac{(z-\mu)^2-\sigma^2}{\sigma^3} \qquad \propto \frac{1}{\sigma}$$

Canonical Search Gradient Instability

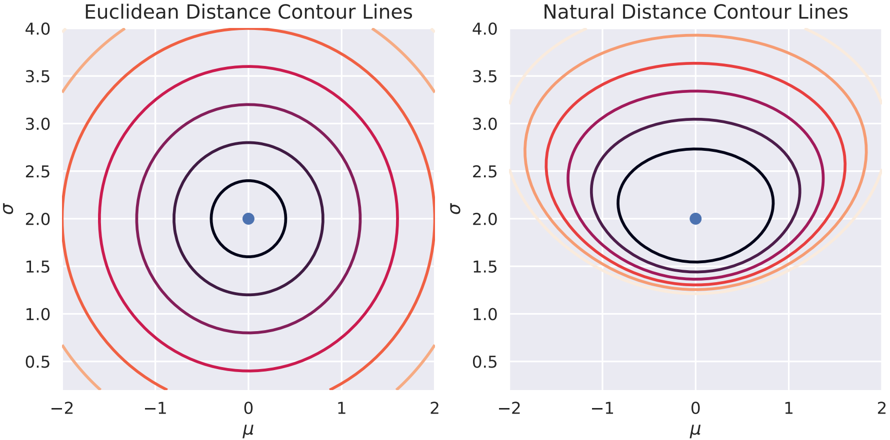




Using canonical search gradient algorithm to maximize $f(z) = z^2$.
Unstable; small σ leads to large updates of μ.

Wierstra, Daan, et al. "Natural evolution strategies." 2014.

Natural Gradient



- Gradient update steps to the contour of a Euclidean ball.
- Natural gradient: measure distance between distributions instead.

Natural Evolution Strategies (NES)

- Remove the dependence on parameterization.
- Distance is Kullback-Lieber divergence between $\pi(\cdot|\theta_t)$ and $\pi(\cdot|\theta_{t+1})$
- Update rule becomes

$$\tilde{\nabla}_\theta J = \mathbf{F}^{-1} \nabla_\theta J$$

- Fisher information matrix

$$\mathbf{F} = \mathbb{E}\left[\nabla_\theta \log \pi(\mathbf{z}|\theta) \nabla_\theta \log \pi(\mathbf{z}|\theta)^T\right]$$

Canonical Natural Evolution Strategies Algorithm

input : f, θ_{init}

repeat

  for $k = 1, \ldots, N$ **do**

    draw samples $z_k \sim \pi(\cdot, \theta)$

    evaluate the fitness $f(z_k)$

    calculate log-derivatives $\nabla_\theta \log \pi(z_k|\theta)$

  end

  $\nabla_\theta J \leftarrow \frac{1}{N} \sum_{k=1}^{N} \nabla_\theta \log \pi(z_k|\theta) \cdot f(z_k)$

  $\mathbf{F} \leftarrow \frac{1}{N} \sum_{k=1}^{N} \nabla_\theta \log \pi(z_k|\theta) \nabla_\theta \log \pi(z_k|\theta)^T$

  $\theta \leftarrow \theta + \eta \cdot \mathbf{F}^{-1} \nabla_\theta J$

until *stopping condition is met*;

Rotationally Symmetric Distributions

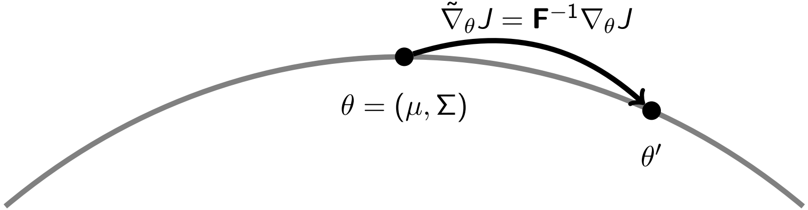
- Parameterize our distribution in terms of mean μ, covariance $\Sigma = A^T A$ and radial shape parameters τ.
- τ indexes a family of radially symmetric distributions with density $Q_\tau(\mathbf{z}) = q_\tau\left(\|\mathbf{z}\|^2\right)$.
- $\mathbf{z} = \mu + A^T \mathbf{s}$, $\mathbf{s} \sim Q_\tau(\cdot)$.
- The complete density is

$$\pi(\mathbf{z}|\mu, A, \tau) = \frac{1}{|\det(A)|} \cdot q_\tau\left(\left\|(A^{-1})^T(\mathbf{z} - \mu)\right\|^2\right)$$

- Problem: A has $\mathcal{O}(d^2)$ parameters, so F has $\mathcal{O}(d^4)$ parameters and inversion costs $\mathcal{O}(d^6)$.

Natural Gradient Update: Parameter Coordinates

$$\mathbf{F} = \mathbb{E}\left[\nabla_\theta \log \pi(\mathbf{z}|\theta) \nabla_\theta \log \pi(\mathbf{z}|\theta)^T\right]$$



Manifold of parameter values θ

Natural Gradient Update: Local Natural Coordinates

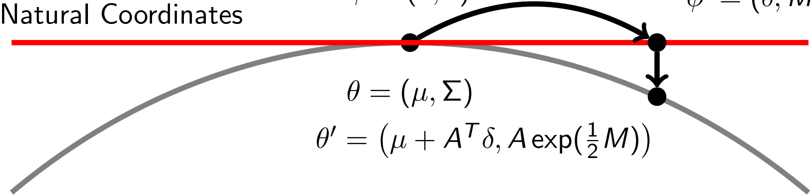
$$\tilde{\mathbf{F}} = \text{Identity matrix (if no } \tau)$$

$$\tilde{\nabla}_\phi J = \tilde{\mathbf{F}}^{-1} \nabla_\phi J$$

Local Tangent Plane
Natural Coordinates

$\phi = (0, 0)$

$\phi' = (\delta, M)$



$\theta = (\mu, \Sigma)$

$$\theta' = \left(\mu + A^T \delta, A \exp(\tfrac{1}{2} M)\right)$$

Manifold of parameter values θ

$$\nabla_\delta = \sum_{k=1}^{N} f(z_k) \cdot s_k$$

$$\nabla_M = \sum_{k=1}^{N} f(z_k) \cdot \left(s_k s_k^T - \mathrm{I}\right)$$

Exponential NES (xNES)

- Formulae on figures were for distributions like multivariate normal that lack a shape parameter τ.
- For general radially symmetric distributions, the Fisher matrix in natural coordinates is:

$$\mathbf{F} = \begin{pmatrix} \mathrm{I} & v \\ v^T & c \end{pmatrix}$$

$$v = \frac{\partial^2 \log \pi(z)}{\partial(\delta, M)\partial\tau} \qquad c = \frac{\partial^2 \log \pi(z)}{\partial\tau^2}$$

- The natural gradient can be computed in $\mathcal{O}(d^2)$ time.

Trick: Fitness Shaping

- Fix a set of *utility* values $u_1 > u_2 > \cdots > u_N$.
- Sort $\{z_k\}$ in descending order of $f(z_k)$
- Use u_k in place of $f(z_k)$ in the gradient calculation:

$$\nabla_\theta J(\theta) = \sum_{k=1}^{N} u_k \nabla_\theta \log \pi(z_k|\theta)$$

- Makes the algorithm invariant to monotonic increasing transformations of the fitness function.

Trick: Fitness Shaping

- Fix a set of *utility* values $u_1 > u_2 > \cdots > u_N$.
- Sort $\{z_k\}$ in descending order of $f(z_k)$
- Use u_k in place of $f(z_k)$ in the gradient calculation:

$$\nabla_\theta J(\theta) = \sum_{k=1}^{N} u_k \nabla_\theta \log \pi(z_k|\theta)$$

- Makes the algorithm invariant to monotonic increasing transformations of the fitness function.

$$u_k = \frac{\max\left(0, \log\left(\frac{N}{2}+1\right) - \log(k)\right)}{\sum_{j=1}^{N} \max\left(0, \log\left(\frac{N}{2}+1\right) - \log(j)\right)} - \frac{1}{N}$$