



**Genetic and Evolutionary  
Computation Conference**



**2014**



Vancouver, BC, Canada  
July 12-16, 2014

A recombination of the  
23rd International Conference on  
Genetic Algorithms (ICGA) and the  
19th Annual Genetic Programming Conference (GP)

**One Conference – Many Mini-Conferences**

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.



Copyright is held by the owner/author(s).

GECCO '14, Jul 12-16 2014, Vancouver, BC,  
Canada

ACM 978-1-4503-2881-4/14/07.

<http://dx.doi.org/10.1145/2598394.2605355>

## Information Geometry in Evolutionary Computation

Luigi Malagò<sup>1</sup> and Tobias Glasmachers<sup>2</sup>

<sup>1</sup> Shinshu University, <sup>2</sup> Ruhr-Universität Bochum

**GECCO Tutorial**

July 13, 2014

## Optimization by Population-based EC

In EC, a common approach to optimize a function is to **evolve** iteratively a population by applying different operators which ensures a tradeoff between

- **exploitation** (e.g., selective pressure)
- **exploration** (e.g., variation, genetic diversity)

## Optimization by Population-based EC

In EC, a common approach to optimize a function is to **evolve** iteratively a population by applying different operators which ensures a tradeoff between

- **exploitation** (e.g., selective pressure)
- **exploration** (e.g., variation, genetic diversity)

Many Evolutionary Algorithms (EAs) follow such paradigm, and can be defined as **population-based**, e.g.,

- Genetic Algorithms (GAs)
- Ant Colony Optimization (ACO)
- Particle Swarm Optimization (PSO)
- Evolution Strategies (ES)
- and many others...

## Population-based EC: Genetic Algorithms

Let us introduce some notation

- $\Omega$  the search space
- $f : \Omega \rightarrow \mathbb{R}$  the function to be optimized
- $\mathcal{P}_t = \{\mathbf{x} \in \Omega\}$  a population of individuals at time  $t$
- $\mathcal{P}_0$  the initial (e.g., random) population

## Population-based EC: Genetic Algorithms

Let us introduce some notation

- $\Omega$  the search space
- $f : \Omega \rightarrow \mathbb{R}$  the function to be optimized
- $\mathcal{P}_t = \{\mathbf{x} \in \Omega\}$  a population of individuals at time  $t$
- $\mathcal{P}_0$  the initial (e.g., random) population

The basic iteration of a naïve GA can be described as

$$\mathcal{P}_t \xrightarrow{\text{selection}} \mathcal{P}_t^s \xrightarrow{\text{crossover}} \mathcal{P}_t^c \xrightarrow{\text{mutation}} \mathcal{P}_{t+1}$$

## A Toy Example with 2 Binary Variables

Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$

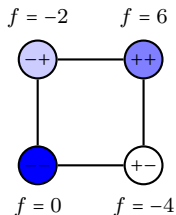
## A Toy Example with 2 Binary Variables

Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$

$\mathcal{P}_0$

1	-1
-1	1
1	-1
-1	-1
1	1
-1	1
-1	1
1	1

Hypercube



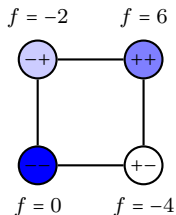
GA:  $\mathcal{P}_t \xrightarrow{\text{truncation selection}} \mathcal{P}_t^s \xrightarrow{\text{1-point crossover}} \mathcal{P}_t^c \xrightarrow{\text{mutation}} \mathcal{P}_{t+1}$

## A Toy Example with 2 Binary Variables

Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$

$\mathcal{P}_0$		$f(\mathbf{x})$
1	-1	-4
-1	1	-2
1	-1	-4
-1	-1	0
1	1	6
-1	1	-2
-1	1	-2
1	1	6

Hypercube



GA:  $\mathcal{P}_t \xrightarrow{\text{truncation selection}} \mathcal{P}_t^s \xrightarrow{\text{1-point crossover}} \mathcal{P}_t^c \xrightarrow{\text{mutation}} \mathcal{P}_{t+1}$

## From Populations to Probability Distributions

A population  $\mathcal{P}$  can be seen as a **sample** i. i. d.  $\sim p$ ,  $p$  probability distribution in the simplex  $\Delta$  for discrete  $\Omega$ , and  $p$  probability density for continuous  $\Omega$

Let  $N$  denote the sample size

$$\mathcal{P} \xrightarrow{\text{estimation}} \hat{p} \qquad \mathcal{P} \xleftarrow{\text{sampling}} p$$

For unbiased estimators and  $N \rightarrow \infty$  (infinite population size analysis)

$$\mathcal{P} \xrightarrow{\text{estimation}} p$$

Such approach is at the basis of the theoretical analysis of Vose (1999) on SGA

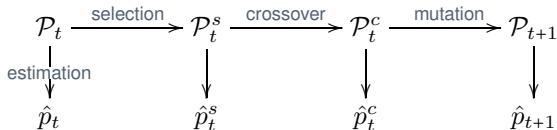
We can describe genetic operators as **maps** from the probability simplex to the the probability simplex itself, e.g.,

$$\text{selection} : \Delta \ni p \mapsto p^s \in \Delta$$

## From Hypercubes to Probability Simplices

A run of a population-based EA identifies a sequence of points in  $\Delta$

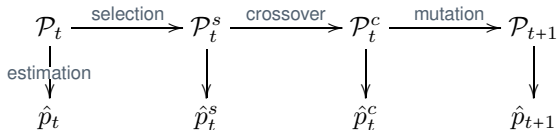
Single run of the  
GA:



# From Hypercubes to Probability Simplices

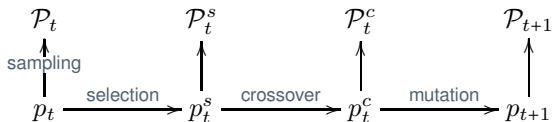
A run of a population-based EA identifies a **sequence of points** in  $\Delta$

Single run of the  
GA:



A run can be seen as a **realization** of the expected behavior of the algorithm

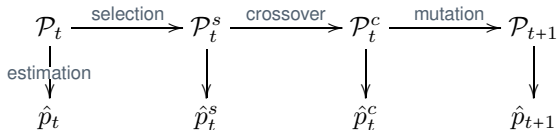
Expected behavior  
of the GA:



# From Hypercubes to Probability Simplices

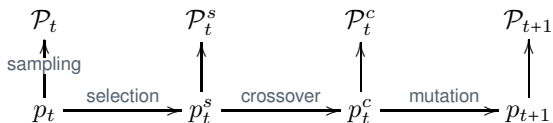
A run of a population-based EA identifies a **sequence of points** in  $\Delta$

Single run of the  
GA:



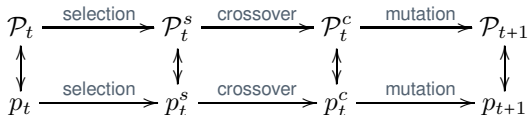
A run can be seen as a **realization** of the expected behavior of the algorithm

Expected behavior  
of the GA:



For unbiased estimators and  $N \rightarrow \infty$ , the map is one-to-one

Infinite population  
size analysis of  
the GA:

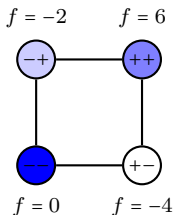


## A Toy Example with 2 Binary Variables (cont.)

Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$

$\mathcal{P}_0$		$f(\mathbf{x})$
1	-1	-4
-1	1	-2
1	-1	-4
-1	-1	0
1	1	6
-1	1	-2
-1	1	-2
1	1	6

Hypercube



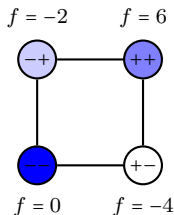
GA:  $p_t \xrightarrow{\text{truncation selection}} p_t^s \xrightarrow{\text{1-point crossover}} p_t^c \xrightarrow{\text{mutation}} p_{t+1}$

## A Toy Example with 2 Binary Variables (cont.)

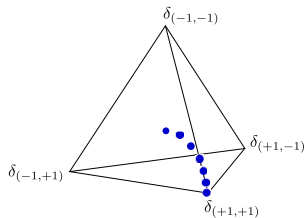
Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$

$\mathcal{P}_0$		$f(\mathbf{x})$
1	-1	-4
-1	1	-2
1	-1	-4
-1	-1	0
1	1	6
-1	1	-2
-1	1	-2
1	1	6

Hypercube



Probability simplex  $\Delta$



GA:  $p_t \xrightarrow{\text{truncation selection}} p_t^s \xrightarrow{\text{1-point crossover}} p_t^c \xrightarrow{\text{mutation}} p_{t+1}$

## Model-Based Optimization

In model-based optimization, the search for the optimum of  $f$  is performed explicitly in the space of probability distributions.

## Model-Based Optimization

In model-based optimization, the search for the optimum of  $f$  is performed explicitly in the space of probability distributions.

By updating the parameters of a probability distribution, iterative algorithms generate sequences of distributions.

## Model-Based Optimization

In model-based optimization, the search for the optimum of  $f$  is performed explicitly in the space of probability distributions.

By updating the parameters of a probability distribution, iterative algorithms generate sequences of distributions.

Candidate solutions for the optimum of  $f$  can be obtained by sampling.

## Model-Based Optimization

In model-based optimization, the search for the optimum of  $f$  is performed explicitly in the space of probability distributions.

By updating the parameters of a probability distribution, iterative algorithms generate sequences of distributions.

Candidate solutions for the optimum of  $f$  can be obtained by sampling.

A model-based algorithm is expected to produce a converging and minimizing sequence, however

- Which statistical model to choose?
- How to generate such sequence?

## Examples of Model-based Algorithms

### Evolutionary computation

- EDAs (Larrañaga and Lozano, 2002), DEUM framework (Shakya et al., 2005)

### Gradient descent

- SGD (Robbins and Monro, 1951), CMA-ES (Hansen and Ostermeier, 2001), NES (Wierstra et al., 2008), SNGD (M. et al., FOGA 2011), IGO (Ollivier et al., 2011),

Boltzmann distribution and Gibbs sampler (Geman and Geman, 1984)

Simulated Annealing and Boltzmann Machines (Aarts and Korst, 1989)

The Cross-Entropy method (Rubinstein, 1997)

*LP relaxation in pseudo-Boolean optimization (Boros and Hammer, 2001)*

*Methods of Moments (Meziat et al., 2001)*

## Model-based EC: Estimation of Distribution

In Estimation of Distribution Algorithms (EDAs) a statistical model is introduced to model interactions among variables of  $f$

Genetic operators (crossover and mutation in GAs) are replaced by statistical operators such as estimation and sampling

## Model-based EC: Estimation of Distribution

In Estimation of Distribution Algorithms (EDAs) a statistical model is introduced to model interactions among variables of  $f$

Genetic operators (crossover and mutation in GAs) are replaced by statistical operators such as estimation and sampling

Let us introduce some more notation

- $p(\mathbf{x}, \boldsymbol{\theta})$  a probability distribution over  $\Omega$  parametrized by  $\boldsymbol{\theta}$
- $\mathcal{M} = \{p(\mathbf{x}, \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$  a parametric statistical model

The basic iteration of an EDA can be described as

$$\mathcal{P}_t \xrightarrow{\text{selection}} \mathcal{P}_t^s \xrightarrow[\text{(model selection)}]{\text{estimation}} p_t \xrightarrow{\text{sampling}} \mathcal{P}_{t+1} \quad p_t \in \mathcal{M}$$

## Model-based EC: Estimation of Distribution

In Estimation of Distribution Algorithms (EDAs) a statistical model is introduced to model interactions among variables of  $f$

Genetic operators (crossover and mutation in GAs) are replaced by statistical operators such as estimation and sampling

Let us introduce some more notation

- $p(\mathbf{x}, \boldsymbol{\theta})$  a probability distribution over  $\Omega$  parametrized by  $\boldsymbol{\theta}$
- $\mathcal{M} = \{p(\mathbf{x}, \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$  a parametric statistical model

The basic iteration of an EDA can be described as

$$\mathcal{P}_t \xrightarrow{\text{selection}} \mathcal{P}_t^s \xrightarrow[\text{(model selection)}]{\text{estimation}} p_t \xrightarrow{\text{sampling}} \mathcal{P}_{t+1} \quad p_t \in \mathcal{M}$$

From a model-based perspective, we have

$$p_t \xrightarrow{\text{sampling}} \mathcal{P}_{t+1} \xrightarrow{\text{selection}} \mathcal{P}_{t+1}^s \xrightarrow[\text{(model selection)}]{\text{estimation}} p_{t+1}$$

## Estimation of Distribution Algorithms

Let  $\mathcal{M}$  to be the independence model for  $\mathbf{x} = (x_1, x_2)$

$$\mathcal{M} = \{p : p(\mathbf{x}) = p_1(x_1)p_2(x_2)\},$$

with  $p_i(x_i) = \mathbb{P}(X_i = x_i)$

We parametrize  $\mathcal{M}$  using marginal probabilities  $\mu_i = p_i(1)$ ,  $\boldsymbol{\mu} = [0, 1]^2$

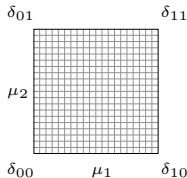
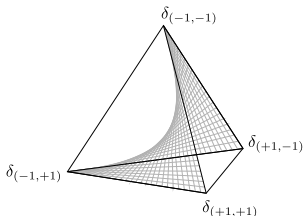
# Estimation of Distribution Algorithms

Let  $\mathcal{M}$  to be the independence model for  $\mathbf{x} = (x_1, x_2)$

$$\mathcal{M} = \{p : p(\mathbf{x}) = p_1(x_1)p_2(x_2)\},$$

with  $p_i(x_i) = \mathbb{P}(X_i = x_i)$

We parametrize  $\mathcal{M}$  using marginal probabilities  $\mu_i = p_i(1)$ ,  $\boldsymbol{\mu} = [0, 1]^2$



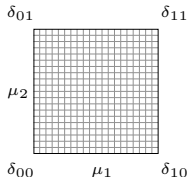
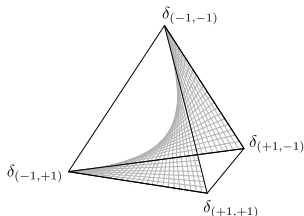
## Estimation of Distribution Algorithms

Let  $\mathcal{M}$  to be the independence model for  $\mathbf{x} = (x_1, x_2)$

$$\mathcal{M} = \{p : p(\mathbf{x}) = p_1(x_1)p_2(x_2)\},$$

with  $p_i(x_i) = \mathbb{P}(X_i = x_i)$

We parametrize  $\mathcal{M}$  using marginal probabilities  $\mu_i = p_i(1)$ ,  $\boldsymbol{\mu} = [0, 1]^2$



$\mathcal{M}$  identifies a 2-dimensional surface in  $\Delta$

Estimation of the parameters given a sample is obtained with a maximum likelihood estimator, i.e., we count occurrences

## Back to the Toy Example with 2 Binary Variables

Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$ , independence model

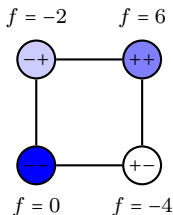
## Back to the Toy Example with 2 Binary Variables

Example:  $\Omega = \{-1, 1\}^2$ ,  $f(\mathbf{x}) = x_1 + 2x_2 + 3x_1x_2$ , independence model

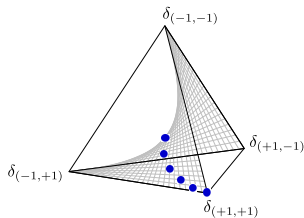
$\mathcal{P}_0$

1	-1
-1	1
1	-1
-1	-1
1	1
-1	1
-1	1
-1	1

Hypercube



Probability simplex



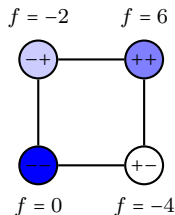
$$\text{EDA: } \mathcal{P}_t \xrightarrow{\text{truncation selection}} \mathcal{P}_t^s \xrightarrow{\text{estimation}} \mathcal{P}_t^c \xrightarrow{\text{sampling}} \mathcal{P}_{t+1}$$

## Back to the Toy Example with 2 Binary Variables

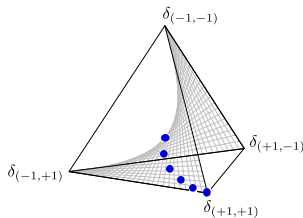
Example:  $\Omega = \{-1, 1\}^2$ ,  $f(x) = x_1 + 2x_2 + 3x_1x_2$ , independence model

$\mathcal{P}_0$	$f(x)$		
<table border="1"><tr><td>1</td><td>-1</td></tr></table>	1	-1	-4
1	-1		
<table border="1"><tr><td>-1</td><td>1</td></tr></table>	-1	1	-2
-1	1		
<table border="1"><tr><td>1</td><td>-1</td></tr></table>	1	-1	-4
1	-1		
<table border="1"><tr><td>-1</td><td>-1</td></tr></table>	-1	-1	0
-1	-1		
<table border="1"><tr><td>1</td><td>1</td></tr></table>	1	1	6
1	1		
<table border="1"><tr><td>-1</td><td>1</td></tr></table>	-1	1	-2
-1	1		
<table border="1"><tr><td>-1</td><td>1</td></tr></table>	-1	1	-2
-1	1		
<table border="1"><tr><td>-1</td><td>1</td></tr></table>	-1	1	6
-1	1		

Hypercube



Probability simplex



$$\text{EDA: } \mathcal{P}_t \xrightarrow{\text{truncation selection}} \mathcal{P}_t^s \xrightarrow{\text{estimation}} \mathcal{P}_t^c \xrightarrow{\text{sampling}} \mathcal{P}_{t+1}$$

## Expected Fitness Landscape

In model-based optimization, the search for the optimum in  $\Omega$  is guided by a search in the space of the probability distributions.

## Expected Fitness Landscape

In model-based optimization, the search for the optimum in  $\Omega$  is guided by a search in the space of the probability distributions.

A natural choice is to optimize the **expected value** of  $f$  over  $\mathcal{M}$ ,

$$\mathbb{E}_p[f] : \mathcal{M} \rightarrow \mathbb{R}$$

which can be expressed as a function of  $\xi$ , given a parameterization for  $p(x, \xi) \in \mathcal{M}$ , i.e.,

$$\xi \mapsto \mathbb{E}_\xi[f]$$

## Equivalent Parameterizations for the Independence Model $p(\mathbf{x}) = p_1(x_1)p_2(x_2)$

Marginal probabilities  $\boldsymbol{\mu} = (\mu_1, \mu_2) \in [0, 1]^2$

$$p_i(x_i) = \mathbb{P}(X_i = x_i) \quad p_i(1) = \mu_i \quad p_i(-1) = 1 - \mu_i$$

$$p_i(x_i) = (2\mu_i x_i - x_i + 1) / 2$$

$$\mathbb{E}_{\boldsymbol{\mu}}[f] = \sum_{\mathbf{x} \in \Omega} f(\mathbf{x}) p_1(x_1) p_2(x_2) = -4\mu_1 - 2\mu_2 + 12\mu_1\mu_2$$

Natural parameters  $\boldsymbol{\theta} = (\theta_1, \theta_2) \in \mathbb{R}^2$  of the Exponential Family

$$p(\mathbf{x}) = \exp\{\theta_1 x_1 + \theta_2 x_2 - \psi(\boldsymbol{\theta})\} \quad \psi(\boldsymbol{\theta}) = \ln \sum_{\mathbf{x} \in \Omega} \exp\{\theta_1 x_1 + \theta_2 x_2\} = \ln Z(\boldsymbol{\theta})$$

$$p_i(x_i) = \frac{e^{\theta_i x_i}}{e^{\theta_i} + e^{-\theta_i}}$$

$$\mathbb{E}_{\boldsymbol{\theta}}[f] = (-4e^{\theta_1 - \theta_2} - 2e^{-\theta_1 + \theta_2} + 6e^{\theta_1 + \theta_2}) / Z(\boldsymbol{\theta})$$

## Equivalent Parameterizations for the Independence Model $p(\mathbf{x}) = p_1(x_1)p_2(x_2)$

Marginal probabilities  $\boldsymbol{\mu} = (\mu_1, \mu_2) \in [0, 1]^2$

$$p_i(x_i) = \mathbb{P}(X_i = x_i) \quad p_i(1) = \mu_i \quad p_i(-1) = 1 - \mu_i$$

$$p_i(x_i) = (2\mu_i x_i - x_i + 1) / 2$$

$$\mathbb{E}_{\boldsymbol{\mu}}[f] = \sum_{\mathbf{x} \in \Omega} f(\mathbf{x}) p_1(x_1) p_2(x_2) = -4\mu_1 - 2\mu_2 + 12\mu_1\mu_2$$

Natural parameters  $\boldsymbol{\theta} = (\theta_1, \theta_2) \in \mathbb{R}^2$  of the Exponential Family

$$p(\mathbf{x}) = \exp\{\theta_1 x_1 + \theta_2 x_2 - \psi(\boldsymbol{\theta})\} \quad \psi(\boldsymbol{\theta}) = \ln \sum_{\mathbf{x} \in \Omega} \exp\{\theta_1 x_1 + \theta_2 x_2\} = \ln Z(\boldsymbol{\theta})$$

$$p_i(x_i) = \frac{e^{\theta_i x_i}}{e^{\theta_i} + e^{-\theta_i}}$$

$$\mathbb{E}_{\boldsymbol{\theta}}[f] = (-4e^{\theta_1 - \theta_2} - 2e^{-\theta_1 + \theta_2} + 6e^{\theta_1 + \theta_2}) / Z(\boldsymbol{\theta})$$

## Equivalent Parameterizations for the Independence Model $p(\mathbf{x}) = p_1(x_1)p_2(x_2)$

Marginal probabilities  $\boldsymbol{\mu} = (\mu_1, \mu_2) \in [0, 1]^2$

$$p_i(x_i) = \mathbb{P}(X_i = x_i) \quad p_i(1) = \mu_i \quad p_i(-1) = 1 - \mu_i$$

$$p_i(x_i) = (2\mu_i x_i - x_i + 1) / 2$$

$$\mathbb{E}_{\boldsymbol{\mu}}[f] = \sum_{\mathbf{x} \in \Omega} f(\mathbf{x}) p_1(x_1) p_2(x_2) = -4\mu_1 - 2\mu_2 + 12\mu_1\mu_2$$

Natural parameters  $\boldsymbol{\theta} = (\theta_1, \theta_2) \in \mathbb{R}^2$  of the Exponential Family

$$p(\mathbf{x}) = \exp\{\theta_1 x_1 + \theta_2 x_2 - \psi(\boldsymbol{\theta})\} \quad \psi(\boldsymbol{\theta}) = \ln \sum_{\mathbf{x} \in \Omega} \exp\{\theta_1 x_1 + \theta_2 x_2\} = \ln Z(\boldsymbol{\theta})$$

$$p_i(x_i) = \frac{e^{\theta_i x_i}}{e^{\theta_i} + e^{-\theta_i}}$$

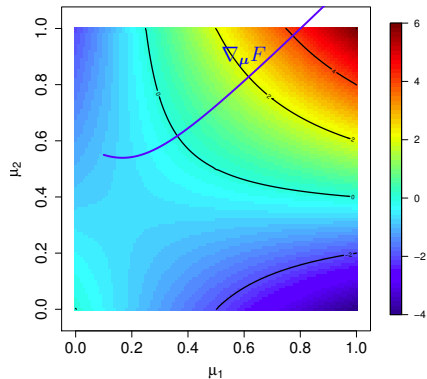
$$\mathbb{E}_{\boldsymbol{\theta}}[f] = (-4e^{\theta_1 - \theta_2} - 2e^{-\theta_1 + \theta_2} + 6e^{\theta_1 + \theta_2}) / Z(\boldsymbol{\theta})$$

The mapping between the two parameterizations is one-to-one for  $p(\mathbf{x}) > 0$

$$\theta_i = (\ln(\mu_i) - \ln(1 - \mu_i)) / 2 \qquad \mu_i = \frac{e^{\theta_i}}{e^{\theta_i} + e^{-\theta_i}}$$

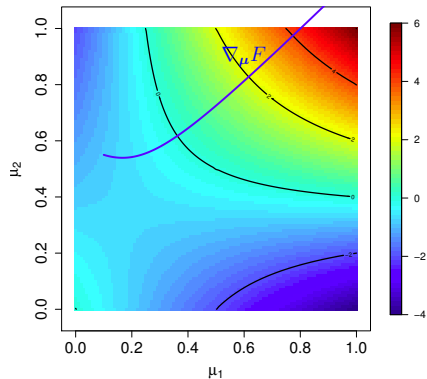
# Gradient Flows on the Independence Model

Marginal probabilities  $\mu$

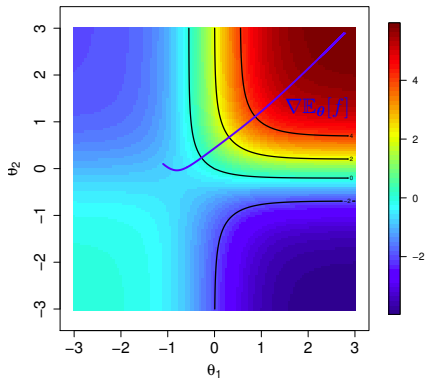


# Gradient Flows on the Independence Model

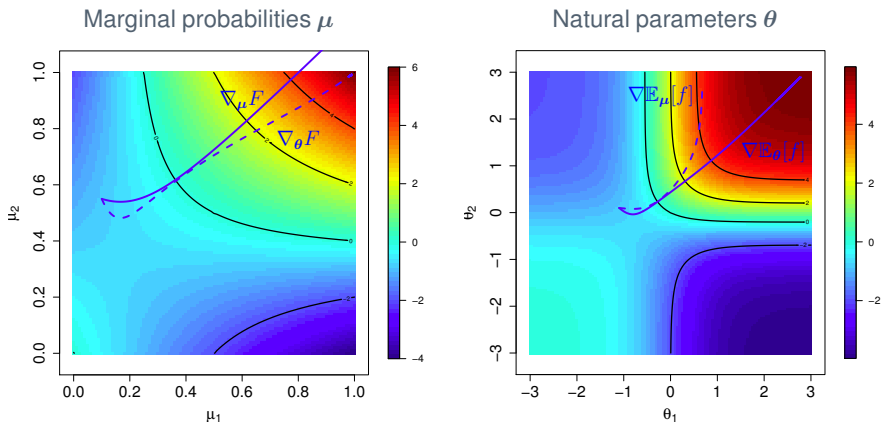
Marginal probabilities  $\mu$



Natural parameters  $\theta$



# Gradient Flows on the Independence Model

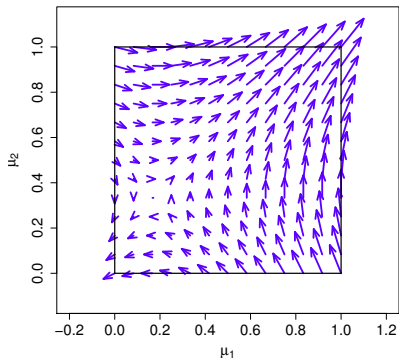


Gradient flows  $\nabla_{\xi}[f]$  depend on the parameterization

In the  $\eta$  parameters,  $\nabla_{\eta}[f]$  does not convergence to the expected distribution  $\delta_x$  over an optimum

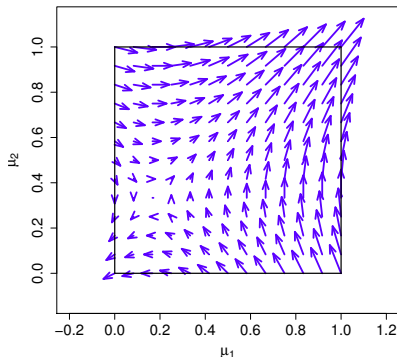
# Gradient Flows on the Independence Model

Marginal probabilities  $\mu$ ,  $\lambda = 0.025$

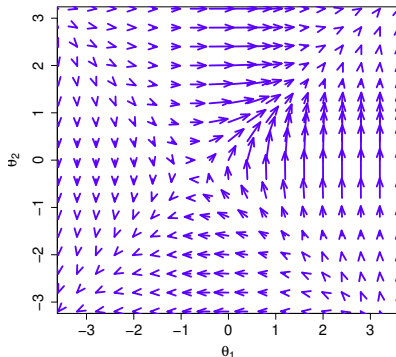


# Gradient Flows on the Independence Model

Marginal probabilities  $\mu$ ,  $\lambda = 0.025$



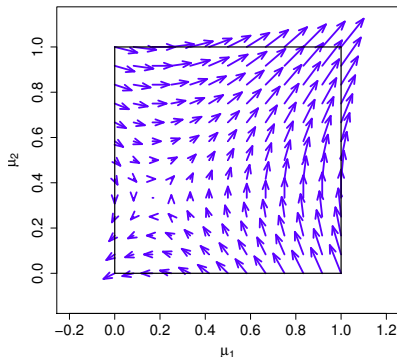
Natural parameters  $\theta$ ,  $\lambda = 0.15$



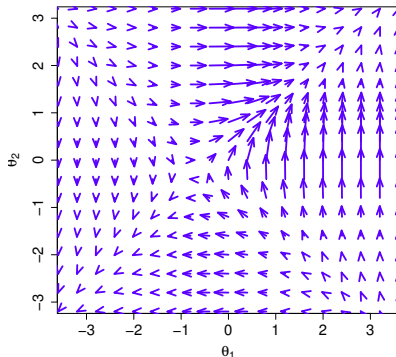
In the  $\theta$  parameters,  $\nabla \mathbb{E}_{\theta}[f]$  vanishes over plateaux

# Gradient Flows on the Independence Model

Marginal probabilities  $\mu$ ,  $\lambda = 0.025$



Natural parameters  $\theta$ ,  $\lambda = 0.15$



In the  $\theta$  parameters,  $\nabla \mathbb{E}_{\theta}[f]$  vanishes over plateaux

We didn't take into account the non-Euclidean geometry of  $\mathcal{M}$

## Summary of the Intro

- There exists a common geometric framework to describe population- and model-based EAs

## Summary of the Intro

- There exists a common geometric framework to describe population- and model-based EAs
- Iterative algorithm generate sequences of distributions which can be compared to the gradient flow of  $\mathbb{E}_p[f]$

## Summary of the Intro

- There exists a common geometric framework to describe population- and model-based EAs
- Iterative algorithm generate sequences of distributions which can be compared to the gradient flow of  $\mathbb{E}_p[f]$
- The choice of the statistical model and of the parameterization plays an important role

## Summary of the Intro

- There exists a common geometric framework to describe population- and model-based EAs
- Iterative algorithm generate sequences of distributions which can be compared to the gradient flow of  $\mathbb{E}_p[f]$
- The choice of the statistical model and of the parameterization plays an important role
- Euclidean geometry does not appear to be the proper geometry for  $\mathcal{M}$

## Summary of the Intro

- There exists a common geometric framework to describe population- and model-based EAs
- Iterative algorithm generate sequences of distributions which can be compared to the gradient flow of  $\mathbb{E}_p[f]$
- The choice of the statistical model and of the parameterization plays an important role
- Euclidean geometry does not appear to be the proper geometry for  $\mathcal{M}$

We need a more general mathematical framework, able to deal with non-Euclidean geometries, to define a unifying perspective on model-based optimization

## Part I

- Stochastic relaxation of the fitness functions
- Introduction to the Information Geometry of statistical models
- Natural Gradient
- Fitness landscape and model selection

## Part II

- Natural Evolution Strategies
- Stochastic Natural Gradient Descent
- Information Geometric Optimization
- Convergence theorems
- Practical performance



# Part I

## Stochastic Relaxation of $f$

Consider the following optimization problem

$$(P) \quad \min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

We define **Stochastic Relaxation (SR)** of  $f$  the function

$$F : p \mapsto \mathbb{E}_p[f]$$

Given a statistical model  $\mathcal{M} = \{p(\mathbf{x})\}$ , we look for the solution of (P) by generating **minimizing sequences**  $\{p_t\}$  in  $\mathcal{M}$  for  $F(p)$

Let  $\xi$  be a parameterization for  $\mathcal{M}$ , i.e.,  $\mathcal{M} = \{p(\mathbf{x}; \xi) : \xi \in \Xi\}$ , the SR can be expressed as

$$(SR) \quad \min_{\xi \in \Xi} F(\xi)$$

We move the search to the **space of probability distribution**

The parameters  $\xi \in \Xi$  become the variables of the relaxed problem

## Equivalence of (P) and (SR)

Let us introduce some notation

- $\mathbf{x}^* \in \Omega^* = \arg \min_{\mathbf{x} \in \Omega} f(\mathbf{x})$  the global optima of  $f$
- $p_* \in \mathcal{M}^* = \arg \min_{p \in \overline{\mathcal{M}}} F(\xi)$  the global optima of  $F$
- $\overline{\mathcal{M}}$  the **topological closure** of  $\mathcal{M}$ , i.e.,  $\mathcal{M}$  together all limit distributions of sequences  $\{p_t\} \in \mathcal{M}$

Candidate solutions for (P) can be **sampled** by solutions of the (SR)

Distributions in  $\mathcal{M}^*$  have reduced support and for discrete  $\Omega$  corresponds to faces of  $\Delta$

(P) and (SR) are **equivalent** if and only if from a solution of (SR) we can sample points in  $\Omega^*$  with  $\mathbb{P}(X = \mathbf{x}^*) = 1$

A sufficient condition is the inclusion of the Dirac distributions  $\delta_{\mathbf{x}^*}$  in  $\overline{\mathcal{M}}$ , i.e., there exists a sequence  $\{p_t\} \in \mathcal{M}$  such that

$$\lim_{t \rightarrow \infty} F(p_t) = \min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

## The Exponential Family

In the following, we consider models in the exponential family  $\mathcal{E}$

$$p(\mathbf{x}, \boldsymbol{\theta}) = \exp\left(\sum_{i=1}^m \theta_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta})\right)$$

- sufficient statistics  $T = (T_1(\mathbf{x}), \dots, T_m(\mathbf{x}))$
- natural parameters  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m) \in \boldsymbol{\theta}$
- log-partition function  $\psi(\boldsymbol{\theta})$

## The Exponential Family

In the following, we consider models in the exponential family  $\mathcal{E}$

$$p(\mathbf{x}, \boldsymbol{\theta}) = \exp\left(\sum_{i=1}^m \theta_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta})\right)$$

- sufficient statistics  $T = (T_1(\mathbf{x}), \dots, T_m(\mathbf{x}))$
- natural parameters  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m) \in \boldsymbol{\theta}$
- log-partition function  $\psi(\boldsymbol{\theta})$

Several statistical models belong to the exponential family (or its closure), both in the continuous and discrete case

- independence model
- tree models
- log-linear models, i.e., Markov random fields
- multivariate Gaussians
- and many others...

(Hwang, 1980; Geman and Geman, 1984)

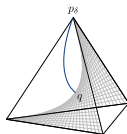
- The Gibbs or Boltzmann distribution is the one dimensional exponential family

$$p(\mathbf{x}; \beta) = \frac{q e^{-\beta f}}{\mathbb{E}_q[e^{-\beta f}]}, \quad \beta > 0$$

- The set of distributions is not weakly closed

$$\lim_{\beta \rightarrow 0} p(\mathbf{x}; \beta) = q$$

$$\lim_{\beta \rightarrow \infty} p(\mathbf{x}; \beta) = \delta_{\Omega^*}$$



(Hwang, 1980; Geman and Geman, 1984)

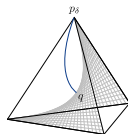
- The Gibbs or Boltzmann distribution is the one dimensional exponential family

$$p(\mathbf{x}; \beta) = \frac{q e^{-\beta f}}{\mathbb{E}_q[e^{-\beta f}]}, \quad \beta > 0$$

- The set of distributions is **not** weakly closed

$$\lim_{\beta \rightarrow 0} p(\mathbf{x}; \beta) = q$$

$$\lim_{\beta \rightarrow \infty} p(\mathbf{x}; \beta) = \delta_{\Omega^*}$$



- The limit  $p_\delta$  is the uniform distribution over the minima of  $f$  and since  $\nabla \mathbb{E}_\beta[f] = -\text{Var}_\beta(f) < 0$ ,  $\mathbb{E}_\beta[f]$  decreases monotonically

(Hwang, 1980; Geman and Geman, 1984)

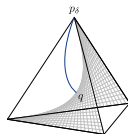
- The Gibbs or Boltzmann distribution is the one dimensional exponential family

$$p(\mathbf{x}; \beta) = \frac{q e^{-\beta f}}{\mathbb{E}_q[e^{-\beta f}]}, \quad \beta > 0$$

- The set of distributions is **not** weakly closed

$$\lim_{\beta \rightarrow 0} p(\mathbf{x}; \beta) = q$$

$$\lim_{\beta \rightarrow \infty} p(\mathbf{x}; \beta) = \delta_{\Omega^*}$$



- The limit  $p_\delta$  is the uniform distribution over the minima of  $f$  and since  $\nabla \mathbb{E}_\beta[f] = -\text{Var}_\beta(f) < 0$ ,  $\mathbb{E}_\beta[f]$  decreases monotonically

Evaluating the partition function is computationally infeasible

The geometry of statistical models is not Euclidean

We need tools from differential geometry to define notions such as tangent vectors, shortest paths and distances between distributions

## Information Geometry

The geometry of statistical models is not Euclidean

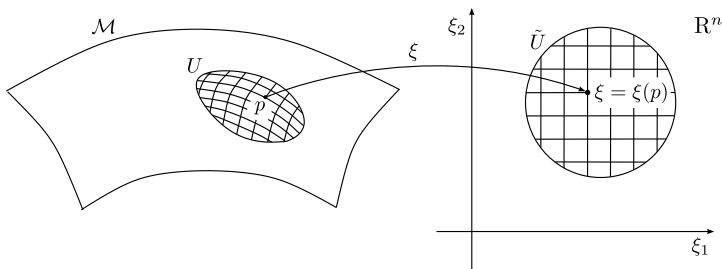
We need tools from differential geometry to define notions such as tangent vectors, shortest paths and distances between distributions

Information Geometry (IG) consists of the study of statistical models as manifolds of distributions endowed with the Fisher information metric  $I$  (Amari 1982, 2001)

The geometry of statistical models is not Euclidean

We need tools from differential geometry to define notions such as tangent vectors, shortest paths and distances between distributions

Information Geometry (IG) consists of the study of statistical models as manifolds of distributions endowed with the Fisher information metric  $I$  (Amari 1982, 2001)



## Characterization of the Tangent Space of $\mathcal{E}$

Over the manifold of distributions we introduce an affine chart in  $p$  such that any density  $q$  is locally represented w.r.t. to the reference measure  $p$ , i.e.,  $\frac{q}{p} - 1$

The tangent space at each point  $p$  is defined by

$$\mathbf{T}_p = \{v : \mathbb{E}_p[v] = 0\}$$

## Characterization of the Tangent Space of $\mathcal{E}$

Over the manifold of distributions we introduce an affine chart in  $p$  such that any density  $q$  is locally represented w.r.t. to the reference measure  $p$ , i.e.,  $\frac{q}{p} - 1$

The tangent space at each point  $p$  is defined by

$$\mathcal{T}_p = \{v : \mathbb{E}_p[v] = 0\}$$

Consider a curve  $p(\theta)$  such that  $p(0) = p$ , then  $\dot{p} \in \mathcal{T}_p$

In a moving coordinate system, tangent (velocity) vectors in  $\mathcal{T}_{p(\theta)}$  to the curve are given by logarithmic derivatives

$$\frac{\dot{p}(\theta)}{p(\theta)} = \frac{d}{d\theta} \log p(\theta)$$

## Characterization of the Tangent Space of $\mathcal{E}$

The one dimensional model

$$p(\theta) = \exp\{\theta T - \psi(\theta)\}$$

is a curve in the manifold, with tangent vector

$$\frac{\dot{p}(\theta)}{p(\theta)} = T - \frac{d}{d\theta}\psi(\theta)$$

On the other side, given a vector field, at each  $p$  we have a vector  $U(p)$  tangent to some curve, we obtain a differential equation

$$\frac{d}{d\theta} \log p(\theta) = U(p),$$

whose solution is a one dimensional model in  $\mathcal{E}$

## (Natural) Gradient

Let  $(\mathcal{M}, I)$  be a statistical manifold endowed with a metric  $I = [g_{ij}]$ , and let  $F(p) : \mathcal{M} \mapsto \mathbb{R}$  be a smooth function defined over  $\mathcal{M}$

## (Natural) Gradient

Let  $(\mathcal{M}, I)$  be a statistical manifold endowed with a metric  $I = [g_{ij}]$ , and let  $F(p) : \mathcal{M} \mapsto \mathbb{R}$  be a smooth function defined over  $\mathcal{M}$

For each vector field  $U$  over  $\mathcal{M}$ , the (natural) gradient of  $F$ , i.e., the direction of steepest descent of  $F$ , denoted by  $\tilde{\nabla} F$ , satisfies

$$g(\tilde{\nabla} F, U) = D_U F,$$

where  $D_U F$  is the directional derivative of  $F$  in the direction  $U$

## (Natural) Gradient

Let  $(\mathcal{M}, I)$  be a statistical manifold endowed with a metric  $I = [g_{ij}]$ , and let  $F(p) : \mathcal{M} \mapsto \mathbb{R}$  be a smooth function defined over  $\mathcal{M}$

For each vector field  $U$  over  $\mathcal{M}$ , the (natural) gradient of  $F$ , i.e., the direction of steepest descent of  $F$ , denoted by  $\tilde{\nabla} F$ , satisfies

$$g(\tilde{\nabla} F, U) = D_U F,$$

where  $D_U F$  is the directional derivative of  $F$  in the direction  $U$

In coordinates  $\xi$  we have

$$\tilde{\nabla}_{\xi} F = \sum_{i=1}^k \sum_{j=1}^k g^{ij} \frac{\partial F}{\partial \xi_i} \frac{\partial}{\partial \xi_j} = I(\xi)^{-1} \nabla_{\xi} F$$

## (Natural) Gradient

Let  $(\mathcal{M}, I)$  be a statistical manifold endowed with a metric  $I = [g_{ij}]$ , and let  $F(p) : \mathcal{M} \mapsto \mathbb{R}$  be a smooth function defined over  $\mathcal{M}$

For each vector field  $U$  over  $\mathcal{M}$ , the (natural) gradient of  $F$ , i.e., the direction of steepest descent of  $F$ , denoted by  $\tilde{\nabla} F$ , satisfies

$$g(\tilde{\nabla} F, U) = D_U F,$$

where  $D_U F$  is the directional derivative of  $F$  in the direction  $U$

In coordinates  $\xi$  we have

$$\tilde{\nabla}_{\xi} F = \sum_{i=1}^k \sum_{j=1}^k g^{ij} \frac{\partial F}{\partial \xi_i} \frac{\partial}{\partial \xi_j} = I(\xi)^{-1} \nabla_{\xi} F$$

There is only one (natural) gradient of  $F$  given by the geometry of  $\mathcal{M}$

We use  $\tilde{\nabla}_{\xi} F$  to distinguish the natural gradient from the vanilla gradient  $\nabla_{\xi} F$ , i.e., the vector of partial derivatives of  $F$  w. r. t.  $\xi$

## Geometry of the Exponential Family

In case of a finite sample space  $\mathcal{X}$ , we have

$$p(\mathbf{x}; \boldsymbol{\theta}) = \exp \left( \sum_{i=1}^k \theta_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta}) \right) \quad \boldsymbol{\theta} \in \mathbb{R}^k$$

and

$$\mathbf{T}_{\boldsymbol{\theta}} = \left\{ v : v = \sum_{i=1}^k a_i (T_i(\mathbf{x}) - \mathbb{E}_{\boldsymbol{\theta}}[T_i]), a_i \in \mathbb{R} \right\}$$

## Geometry of the Exponential Family

In case of a finite sample space  $\mathcal{X}$ , we have

$$p(\mathbf{x}; \boldsymbol{\theta}) = \exp\left(\sum_{i=1}^k \theta_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta})\right) \quad \boldsymbol{\theta} \in \mathbb{R}^k$$

and

$$\mathbf{T}_{\boldsymbol{\theta}} = \left\{ v : v = \sum_{i=1}^k a_i (T_i(\mathbf{x}) - \mathbb{E}_{\boldsymbol{\theta}}[T_i]), a_i \in \mathbb{R} \right\}$$

Since  $\nabla_{\boldsymbol{\theta}} F = \text{Cov}_{\boldsymbol{\theta}}(f, T)$ , if  $f \in \mathbf{T}_p$ , the steepest direction is given by  $f - \mathbb{E}_{\boldsymbol{\theta}}[f]$ , otherwise we take the projection of  $f$  onto  $\mathbf{T}_p$

$$\hat{f} = \sum_{i=1}^k \hat{a}_i (T_i(\mathbf{x}) - \mathbb{E}_{\boldsymbol{\theta}}[T_i]),$$

and obtain  $\hat{f}$  by solving a system of linear equations

Since  $f - \hat{f}$  is orthogonal to  $T_p$

$$\mathbb{E}_{\boldsymbol{\theta}}[(f - \hat{f}_{\boldsymbol{\theta}})(T - \mathbb{E}_{\boldsymbol{\theta}}[T])] = \text{Cov}_{\boldsymbol{\theta}}(f - \hat{f}_{\boldsymbol{\theta}}, T) = 0,$$

from which we obtain, for  $i = 1, \dots, k$ ,

$$\text{Cov}_{\boldsymbol{\theta}}(f, T_i) = \text{Cov}_{\boldsymbol{\theta}}(\hat{f}_{\boldsymbol{\theta}}, T_i) = \sum_{j=1}^k \hat{a}_j \text{Cov}_{\boldsymbol{\theta}}(T_j, T_i)$$

## Geometry of Statistical Models

Since  $f - \hat{f}$  is orthogonal to  $T_p$

$$\mathbb{E}_{\theta}[(f - \hat{f}_{\theta})(T - \mathbb{E}_{\theta}[T])] = \text{Cov}_{\theta}(f - \hat{f}_{\theta}, T) = 0,$$

from which we obtain, for  $i = 1, \dots, k$ ,

$$\text{Cov}_{\theta}(f, T_i) = \text{Cov}_{\theta}(\hat{f}_{\theta}, T_i) = \sum_{j=1}^k \hat{a}_j \text{Cov}_{\theta}(T_j, T_i)$$

As the Hessian matrix of  $\psi(\theta)$  is invertible, we have

$$\hat{a} = [\text{Cov}_{\theta}(T_i, T_j)]^{-1} \text{Cov}_{\theta}(f, T) = I(\theta)^{-1} \nabla F(\theta)$$

In case  $f \in \text{Span}\{T_1, \dots, T_k\}$ , then  $\hat{f}_{\theta} = f$

By taking projection of  $f$  to  $T_p$ , we obtained the **natural gradient**  $\tilde{\nabla} F$ , i.e., the gradient evaluated w.r.t. the Fisher information metric  $I$

## Theorem 1 (M. et al., CEC 2013)

If the sufficient statistics  $\{T_i\}$  of  $p(x; \theta) \in \mathcal{E}$  are centered in  $\theta$ , i.e.,  $\mathbb{E}_\theta[T_i] = 0$ , then the least squares estimator  $\hat{c}_N$  with respect to an i.i.d. sample  $\mathcal{P}$  from  $p$  of the regression model

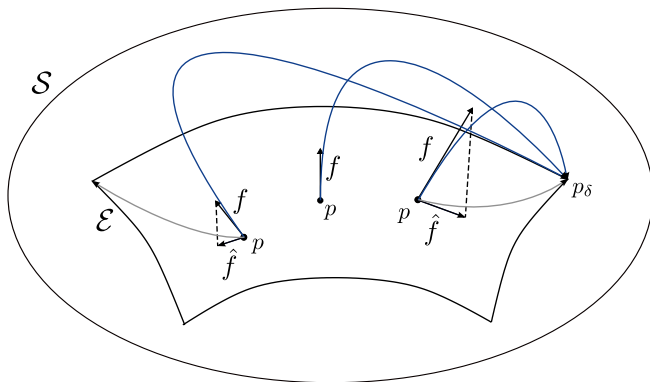
$$\hat{f}(x) = \sum_i a_i T_i(x)$$

converges to the natural gradient  $\tilde{\nabla} \mathbb{E}_\theta[f]$ , as  $N \rightarrow \infty$

*Proof.*

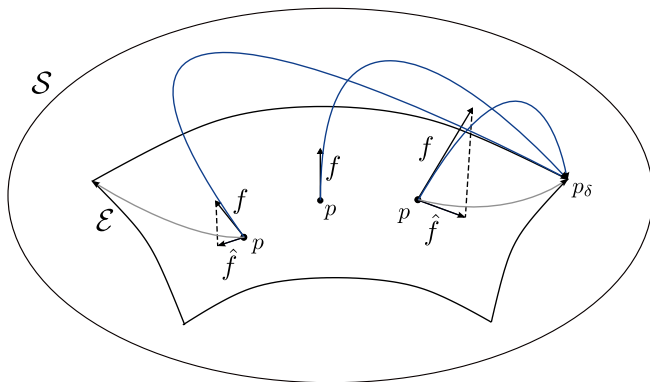
$$\begin{aligned} \hat{a}_N &= (A^\top A)^{-1} A^\top \mathbf{y} \\ &= \left[ \frac{1}{N} \sum_{x \in \mathcal{P}} T_i(x) T_j(x) \right]_{x,i}^{-1} \left( \frac{1}{N} \sum_{x \in \mathcal{P}} f(x) T_i(x) \right)_i \\ &= [\widehat{\text{Cov}}(T_i, T_j) + \widehat{\mathbb{E}}[T_i] \widehat{\mathbb{E}}[T_j]]_{x,i}^{-1} (\widehat{\text{Cov}}(f, T_i) + \widehat{\mathbb{E}}[f] \widehat{\mathbb{E}}[T_i])_i \end{aligned}$$

If  $f \notin T_p$ , the projection  $\hat{f}$  may vanish, and local minima may appear



# The Big Picture

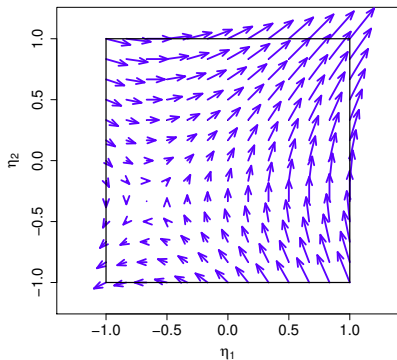
If  $f \notin T_p$ , the projection  $\hat{f}$  may vanish, and local minima may appear



For finite  $\Omega$ ,  $f = \sum_{\alpha \in L} c_\alpha \mathbf{x}^\alpha$  with  $\mathbf{x} = \prod_i x_i^{\alpha_i}$  and  $L = \{0, 1\}^n$ , consider the exponential family  $\mathcal{E}$  with sufficient statistics  $T_\beta(\mathbf{x}) = \mathbf{x}^\beta$ , with  $\beta \in M = \{0, 1\}^n \setminus 0$ , then  $f \in T_p$  iff  $L \setminus M \cup 0$

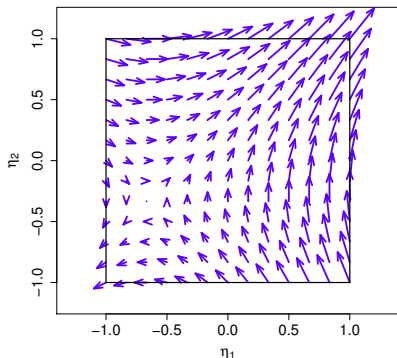
# Vanilla vs Natural Gradient: $\eta, \lambda = 0.05$

Vanilla gradient  $\nabla F$

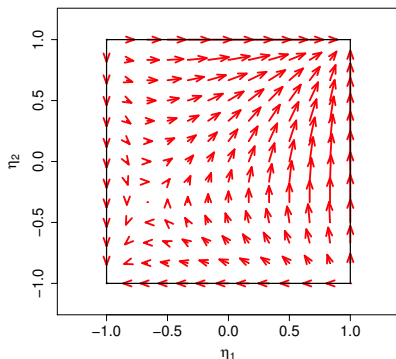


# Vanilla vs Natural Gradient: $\eta, \lambda = 0.05$

Vanilla gradient  $\nabla F$



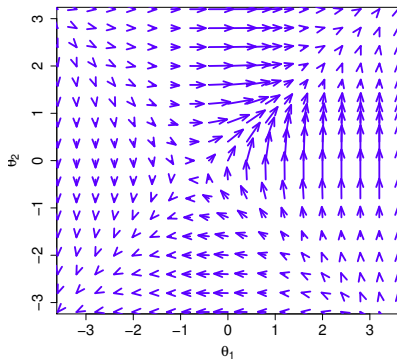
Natural gradient  $\tilde{\nabla} F$



There exist two basins of attraction,  $\tilde{\nabla} \mathbb{E}_{\eta}[f]$  convergences to  $\delta_x$  distributions, which are local optima for  $F$ , i.e.,  $\tilde{\nabla} \mathbb{E}_{\delta_x}[f] = 0$

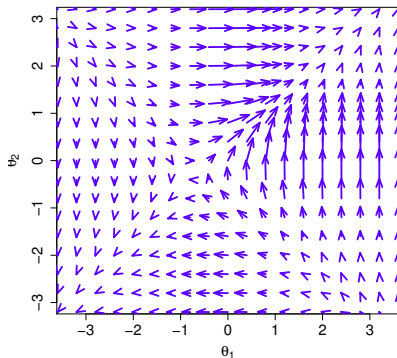
# Vanilla vs Natural Gradient: $\theta, \lambda = 0.15$

Vanilla gradient  $\nabla F$

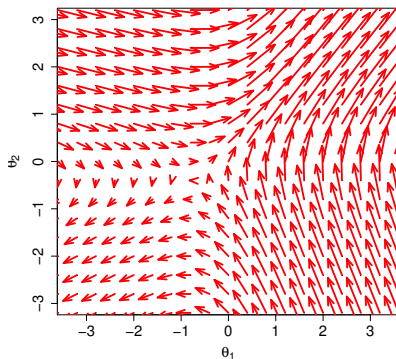


# Vanilla vs Natural Gradient: $\theta, \lambda = 0.15$

Vanilla gradient  $\nabla F$



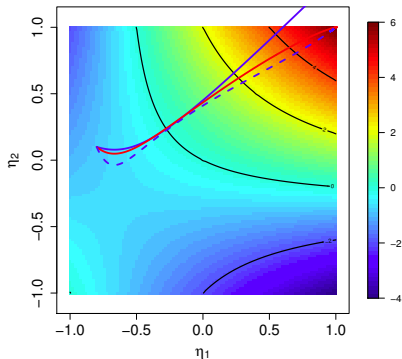
Natural gradient  $\tilde{\nabla} F$



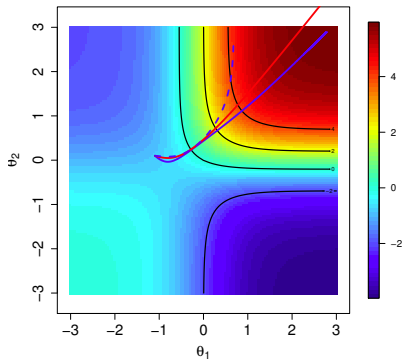
In the natural parameters  $\tilde{\nabla} \mathbb{E}_{\theta}[f]$  speeds up over the plateaux

# Vanilla vs Natural Gradient

Expectation parameters  $\eta$



Natural parameters  $\theta$



Vanilla gradient  $\nabla F$  vs Natural gradient  $\tilde{\nabla} F$

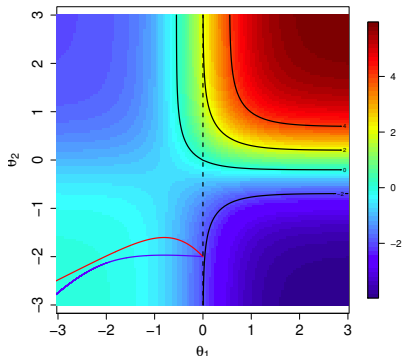
For infinitesimal step size ( $\lambda \rightarrow 0$ ), the gradient flow is invariant to parameterization

## Choice of $\mathcal{M}$

The choice of the statistical model  $\mathcal{M}$  determines the landscape of  $F$

Independence model,  $\theta = (\theta_1, \theta_2, 0)$

$$p(\mathbf{x}) = \exp\{\theta_1 x_1 + \theta_2 x_2 - \psi(\theta)\}$$

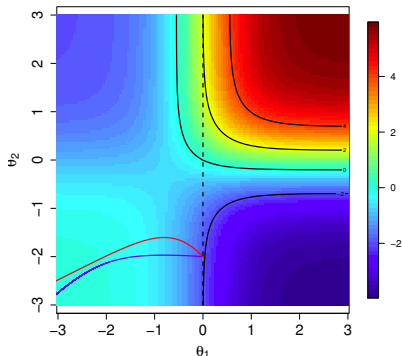


# Choice of $\mathcal{M}$

The choice of the statistical model  $\mathcal{M}$  determines the landscape of  $F$

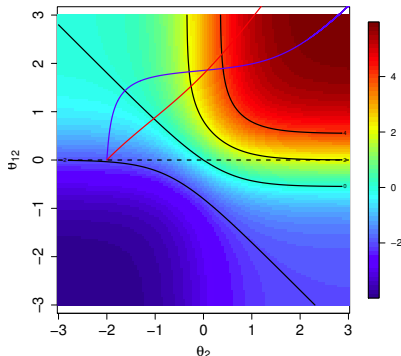
Independence model,  $\theta = (\theta_1, \theta_2, 0)$

$$p(\mathbf{x}) = \exp\{\theta_1 x_1 + \theta_2 x_2 - \psi(\boldsymbol{\theta})\}$$



Exponential family,  $\theta = (0, \theta_2, \theta_{12})$

$$p(\mathbf{x}) = \exp\{\theta_2 x_2 + \theta_{12} x_1 x_2 - \psi(\boldsymbol{\theta})\}$$



Vanilla gradient  $\nabla F$  vs Natural gradient  $\tilde{\nabla} F$

## Generating minimizing sequences $\{p_t\}$

In model-based optimization, the relaxed problem (SR) can be approached with different techniques, among the other we have

- Estimation of distribution EDAs, see Larrañaga and Lozano (2002) for a review
- Covariance Matrix Adaptation CMA-ES (Hansen and Ostermeier, 2001)
- Fitness modelling DEUM framework (Shakya et al., 2005)
- Gradient descent NES (Wierstra et al., 2008), SNGD (M. et al., FOGA 2011), IGO (Arnold et al., 2011)

In the following we will show how a geometrical framework based on Information Geometry can be exploited to relate these different approaches

# Part II



## A General Framework for Algorithms

- In the first part we have seen natural gradients on distributions.

## A General Framework for Algorithms

- In the first part we have seen natural gradients on distributions.
- Now we will derive concrete algorithms from this general framework.

## A General Framework for Algorithms

- In the first part we have seen natural gradients on distributions.
- Now we will derive concrete algorithms from this general framework.
- Design choices:
  - search space: discrete or continuous, structure?
  - statistical model?
  - stochastic relaxation?
  - efficient computation/estimation of the natural gradient?

## The Log-Likelihood Trick

Assume the objective  $W_f(\xi) = \mathbb{E}_\xi[w(f(\mathbf{x}))]$ .

## The Log-Likelihood Trick

Assume the objective  $W_f(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{\xi}}[w(f(\boldsymbol{x}))]$ .

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} W_f(\boldsymbol{\xi}) &= \nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\xi}}[w(f(\boldsymbol{x}))] \\&= \nabla_{\boldsymbol{\xi}} \int_{\Omega} w(f(\boldsymbol{x})) \cdot p(\boldsymbol{x}|\boldsymbol{\xi}) \, d\boldsymbol{x} \\&= \int_{\Omega} w(f(\boldsymbol{x})) \cdot \nabla_{\boldsymbol{\xi}} p(\boldsymbol{x}|\boldsymbol{\xi}) \, d\boldsymbol{x} \\&= \int_{\Omega} w(f(\boldsymbol{x})) \cdot \nabla_{\boldsymbol{\xi}} p(\boldsymbol{x}|\boldsymbol{\xi}) \cdot \frac{p(\boldsymbol{x}|\boldsymbol{\xi})}{p(\boldsymbol{x}|\boldsymbol{\xi})} \, d\boldsymbol{x} \\&= \int_{\Omega} w(f(\boldsymbol{x})) \cdot \frac{\nabla_{\boldsymbol{\xi}} p(\boldsymbol{x}|\boldsymbol{\xi})}{p(\boldsymbol{x}|\boldsymbol{\xi})} \cdot p(\boldsymbol{x}|\boldsymbol{\xi}) \, d\boldsymbol{x} \\&= \mathbb{E}_{\boldsymbol{\xi}}[w(f(\boldsymbol{x})) \cdot \nabla_{\boldsymbol{\xi}} \log(p(\boldsymbol{x}|\boldsymbol{\xi}))]\end{aligned}$$

## The Log-Likelihood Trick

Assume the objective  $W_f(\xi) = \mathbb{E}_\xi[w(f(x))]$ .

$$\begin{aligned}\nabla_\xi W_f(\xi) &= \nabla_\xi \mathbb{E}_\xi[w(f(x))] \\&= \nabla_\xi \int_{\Omega} w(f(x)) \cdot p(x|\xi) \, dx \\&= \int_{\Omega} w(f(x)) \cdot \nabla_\xi p(x|\xi) \, dx \\&= \int_{\Omega} w(f(x)) \cdot \nabla_\xi p(x|\xi) \cdot \frac{p(x|\xi)}{p(x|\xi)} \, dx \\&= \int_{\Omega} w(f(x)) \cdot \frac{\nabla_\xi p(x|\xi)}{p(x|\xi)} \cdot p(x|\xi) \, dx \\&= \mathbb{E}_\xi[w(f(x)) \cdot \nabla_\xi \log(p(x|\xi))]\end{aligned}$$

The gradient of the expectation can be written as the expectation of a weighted gradient of the log likelihood.

## The Log-Likelihood Trick

$$\nabla_{\boldsymbol{\xi}} W_f(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{\xi}} \left[ w(f(\boldsymbol{x})) \cdot \nabla_{\boldsymbol{\xi}} \log(p(\boldsymbol{x}|\boldsymbol{\xi})) \right]$$

## The Log-Likelihood Trick

$$\nabla_{\boldsymbol{\xi}} W_f(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{\xi}} \left[ w(f(\boldsymbol{x})) \cdot \nabla_{\boldsymbol{\xi}} \log(p(\boldsymbol{x}|\boldsymbol{\xi})) \right]$$

- The expected value can be estimated efficiently.

## The Log-Likelihood Trick

$$\nabla_{\xi} W_f(\xi) = \mathbb{E}_{\xi} \left[ w(f(\mathbf{x})) \cdot \nabla_{\xi} \log(p(\mathbf{x}|\xi)) \right]$$

- The expected value can be estimated efficiently.
- Its Monte Carlo estimate reads:

$$\nabla_{\xi} W_f(\xi) \approx \frac{1}{N} \sum_{\mathbf{x}_1, \dots, \mathbf{x}_N \sim P_{\xi}} w(f(\mathbf{x}_i)) \cdot \nabla_{\xi} \log(p(\mathbf{x}_i|\xi))$$

## The Log-Likelihood Trick

$$\nabla_{\xi} W_f(\xi) = \mathbb{E}_{\xi} \left[ w(f(\mathbf{x})) \cdot \nabla_{\xi} \log(p(\mathbf{x}|\xi)) \right]$$

- The expected value can be estimated efficiently.
- Its Monte Carlo estimate reads:

$$\nabla_{\xi} W_f(\xi) \approx \frac{1}{N} \sum_{\mathbf{x}_1, \dots, \mathbf{x}_N \sim P_{\xi}} w(f(\mathbf{x}_i)) \cdot \nabla_{\xi} \log(p(\mathbf{x}_i|\xi))$$

- Note: neither the gradient of  $W_f$  nor its approximation require the gradient of  $f$ .

## Stochastic Gradient Descent (SGD)

- **Gradient Descent (GD):**

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} W_f(\xi) \ .$$

## Stochastic Gradient Descent (SGD)

- **Gradient Descent (GD):**

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} W_f(\xi) \ .$$

- The parameter  $\gamma > 0$  is called learning rate.

# Stochastic Gradient Descent (SGD)

- **Gradient Descent (GD):**

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} W_f(\xi) \ .$$

- The parameter  $\gamma > 0$  is called learning rate.
- Following an unbiased gradient estimate  $G(\xi)$  (with  $\mathbb{E}[G(\xi)] = \nabla_{\xi} W_f(\xi)$ ) is known as **stochastic gradient descent (SGD)**:

$$\xi \leftarrow \xi - \gamma \cdot G(\xi) \ .$$

## Stochastic Gradient Descent (SGD)

- **Gradient Descent (GD):**

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} W_f(\xi) \ .$$

- The parameter  $\gamma > 0$  is called learning rate.
- Following an unbiased gradient estimate  $G(\xi)$  (with  $\mathbb{E}[G(\xi)] = \nabla_{\xi} W_f(\xi)$ ) is known as **stochastic gradient descent (SGD)**:

$$\xi \leftarrow \xi - \gamma \cdot G(\xi) \ .$$

- This is a well-established optimization algorithm with many applications, e.g., in machine learning.

## Stochastic Gradient Descent (SGD)

- **Gradient Descent (GD):**

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} W_f(\xi) \ .$$

- The parameter  $\gamma > 0$  is called learning rate.
- Following an unbiased gradient estimate  $G(\xi)$  (with  $\mathbb{E}[G(\xi)] = \nabla_{\xi} W_f(\xi)$ ) is known as **stochastic gradient descent (SGD)**:

$$\xi \leftarrow \xi - \gamma \cdot G(\xi) \ .$$

- This is a well-established optimization algorithm with many applications, e.g., in machine learning.
- Replacing  $\nabla W_f(\xi)$  with  $\tilde{\nabla} W_f(\xi)$  results in stochastic **natural** gradient descent (SNGD).

## Stochastic Gradient Descent (SGD)

- **Gradient Descent (GD):**

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} W_f(\xi) \ .$$

- The parameter  $\gamma > 0$  is called learning rate.
- Following an unbiased gradient estimate  $G(\xi)$  (with  $\mathbb{E}[G(\xi)] = \nabla_{\xi} W_f(\xi)$ ) is known as **stochastic gradient descent (SGD)**:

$$\xi \leftarrow \xi - \gamma \cdot G(\xi) \ .$$

- This is a well-established optimization algorithm with many applications, e.g., in machine learning.
- Replacing  $\nabla W_f(\xi)$  with  $\tilde{\nabla} W_f(\xi)$  results in stochastic **natural** gradient descent (SNGD).
- Concrete scheme for iterative optimization of  $W_f(\xi)$ .

## Continuous Optimization with NES

- Such a scheme was first proposed by Wierstra et al. in 2008.

## Continuous Optimization with NES

- Such a scheme was first proposed by Wierstra et al. in 2008.
- Original **Natural Evolution Strategies (NES)** approach: SNGD on expected fitness  $W_f(\xi) = \mathbb{E}_{\xi}[f(x)]$  with multi-variate Gaussians  $\mathcal{N}(\mathbf{m}, \mathbf{C})$ .

## Continuous Optimization with NES

- Such a scheme was first proposed by Wierstra et al. in 2008.
- Original **Natural Evolution Strategies (NES)** approach: SNGD on expected fitness  $W_f(\xi) = \mathbb{E}_{\xi}[f(x)]$  with multi-variate Gaussians  $\mathcal{N}(\mathbf{m}, \mathbf{C})$ .
- Closed form Fisher matrix:

$$\mathcal{I}_{i,j} = \frac{\partial \mathbf{m}^T}{\partial \xi_i} \mathbf{C}^{-1} \frac{\partial \mathbf{m}}{\partial \xi_j} + \frac{1}{2} \text{tr} \left( \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \xi_i} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \xi_j} \right)$$

## Continuous Optimization with NES

- Such a scheme was first proposed by Wierstra et al. in 2008.
- Original **Natural Evolution Strategies (NES)** approach: SNGD on expected fitness  $W_f(\xi) = \mathbb{E}_{\xi}[f(x)]$  with multi-variate Gaussians  $\mathcal{N}(\mathbf{m}, \mathbf{C})$ .
- Closed form Fisher matrix:

$$\mathcal{I}_{i,j} = \frac{\partial \mathbf{m}^T}{\partial \xi_i} \mathbf{C}^{-1} \frac{\partial \mathbf{m}}{\partial \xi_j} + \frac{1}{2} \text{tr} \left( \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \xi_i} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \xi_j} \right)$$

- Practical versions of NES apply many performance enhancing techniques like rank-based utilities and non-uniform learning rates that complement the SNGD approach.

## Natural Gradients for Gaussian Distributions

- $\Omega = \mathbb{R}^d$ ,  $\xi = (\mathbf{m}, \mathbf{C})$ , Gaussian density:

$$p(\mathbf{x}|\xi) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})\right) .$$

## Natural Gradients for Gaussian Distributions

- $\Omega = \mathbb{R}^d$ ,  $\xi = (\mathbf{m}, \mathbf{C})$ , Gaussian density:

$$p(\mathbf{x}|\xi) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})\right) .$$

Its natural logarithm is

$$\log(p(\mathbf{x}|\xi)) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \text{tr}(\log(\mathbf{C})) - \frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}) .$$

## Natural Gradients for Gaussian Distributions

- $\Omega = \mathbb{R}^d$ ,  $\xi = (\mathbf{m}, \mathbf{C})$ , Gaussian density:

$$p(\mathbf{x}|\xi) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})\right) .$$

Its natural logarithm is

$$\log(p(\mathbf{x}|\xi)) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \text{tr}(\log(\mathbf{C})) - \frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}) .$$

- Represent covariance matrix with a factor:  $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ .

## Natural Gradients for Gaussian Distributions

- $\Omega = \mathbb{R}^d$ ,  $\xi = (\mathbf{m}, \mathbf{C})$ , Gaussian density:

$$p(\mathbf{x}|\xi) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})\right) .$$

Its natural logarithm is

$$\log(p(\mathbf{x}|\xi)) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \text{tr}(\log(\mathbf{C})) - \frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}) .$$

- Represent covariance matrix with a factor:  $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ .
- For  $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$  introduce normalized sample  $\mathbf{z} = \mathbf{A}^{-1}(\mathbf{x} - \mathbf{m}) \sim \mathcal{N}(0, \mathbf{I})$ .

## Natural Gradients for Gaussian Distributions

- $\Omega = \mathbb{R}^d$ ,  $\xi = (\mathbf{m}, \mathbf{C})$ , Gaussian density:

$$p(\mathbf{x}|\xi) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})\right) .$$

Its natural logarithm is

$$\log(p(\mathbf{x}|\xi)) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \text{tr}(\log(\mathbf{C})) - \frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}) .$$

- Represent covariance matrix with a factor:  $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ .
- For  $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$  introduce normalized sample  $\mathbf{z} = \mathbf{A}^{-1}(\mathbf{x} - \mathbf{m}) \sim \mathcal{N}(0, \mathbf{I})$ .
- In tailored coordinates

$$(\mathbf{m}', \mathbf{A}') = \left( \mathbf{m} + \mathbf{A}\delta, \mathbf{A}\left(\mathbf{I} + \frac{1}{2}\mathbf{M}\right) \right)$$

centered to the current distribution  $(\mathbf{m}, \mathbf{A})$  the Fisher matrix w.r.t. the local parameters  $\xi = (\delta, \mathbf{M})$  becomes the identity.

## Natural Gradients for Gaussian Distributions

- The (natural) gradient of the log density at  $(\delta, \mathbf{M}) = 0$  is

$$\begin{aligned}\tilde{\nabla}_{\delta} \log(p(\mathbf{x}|\boldsymbol{\xi})) &= \mathbf{z} \\ \tilde{\nabla}_{\mathbf{M}} \log(p(\mathbf{x}|\boldsymbol{\xi})) &= \frac{1}{2}(\mathbf{z}\mathbf{z}^T - \mathbf{I})\end{aligned}$$

## Natural Gradients for Gaussian Distributions

- The (natural) gradient of the log density at  $(\delta, \mathbf{M}) = 0$  is

$$\begin{aligned}\tilde{\nabla}_{\delta} \log(p(\mathbf{x}|\boldsymbol{\xi})) &= \mathbf{z} \\ \tilde{\nabla}_{\mathbf{M}} \log(p(\mathbf{x}|\boldsymbol{\xi})) &= \frac{1}{2}(\mathbf{z}\mathbf{z}^T - \mathbf{I})\end{aligned}$$

- The stochastic (natural) gradient of  $\mathbb{E}[f]$  becomes

$$\begin{aligned}G_{\delta}(\boldsymbol{\xi}) &= \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \cdot \mathbf{z}_i \\ G_{\mathbf{M}}(\boldsymbol{\xi}) &= \frac{1}{2N} \sum_{i=1}^N f(\mathbf{x}_i) \cdot (\mathbf{z}_i \mathbf{z}_i^T - \mathbf{I})\end{aligned}$$

## Natural Gradients for Gaussian Distributions

- The (natural) gradient of the log density at  $(\delta, \mathbf{M}) = 0$  is

$$\begin{aligned}\tilde{\nabla}_{\delta} \log(p(\mathbf{x}|\boldsymbol{\xi})) &= \mathbf{z} \\ \tilde{\nabla}_{\mathbf{M}} \log(p(\mathbf{x}|\boldsymbol{\xi})) &= \frac{1}{2}(\mathbf{z}\mathbf{z}^T - \mathbf{I})\end{aligned}$$

- The stochastic (natural) gradient of  $\mathbb{E}[f]$  becomes

$$\begin{aligned}G_{\delta}(\boldsymbol{\xi}) &= \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \cdot \mathbf{z}_i \\ G_{\mathbf{M}}(\boldsymbol{\xi}) &= \frac{1}{2N} \sum_{i=1}^N f(\mathbf{x}_i) \cdot (\mathbf{z}_i \mathbf{z}_i^T - \mathbf{I})\end{aligned}$$

- Tricks of the trade: replace “raw fitness” with “rank-based utility weights”

$$f(\mathbf{x}_1) \leq \dots \leq f(\mathbf{x}_N) \quad \rightarrow \quad u_1 \geq \dots \geq u_N$$

to achieve better invariance and faster convergence.

## Natural Evolution Strategies (NES)

Canonical NES algorithm with Gaussians  $\mathcal{N}(\mathbf{m}, \mathbf{C} = \mathbf{A}\mathbf{A}^T)$

while stopping criterion not fulfilled do

  // sample offspring

  for  $i \in \{1, \dots, N\}$  do

$\mathbf{z}_i \leftarrow \mathcal{N}(0, \mathbf{I})$

$\mathbf{x}_i \leftarrow \mathbf{m} + \mathbf{A} \cdot \mathbf{z}_i$

  sort  $\{(\mathbf{z}_i, \mathbf{x}_i)\}$  w.r.t.  $f(\mathbf{x}_i)$

  // compute stochastic natural gradient

$G_\delta \leftarrow \frac{1}{N} \sum_{i=1}^N u_i \cdot \mathbf{z}_i$

$G_M \leftarrow \frac{1}{2N} \sum_{i=1}^N u_i \cdot (\mathbf{z}_i \mathbf{z}_i^T - \mathbf{I})$

  // apply update

$\mathbf{m} \leftarrow \mathbf{m} + \gamma_m \cdot \mathbf{A} \cdot G_\delta$

$\mathbf{A} \leftarrow \mathbf{A} \cdot (\mathbf{I} + \gamma_A \cdot G_M)$

loop

## Natural Evolution Strategies (NES)

- NES (Wierstra et al., 2008) is a CMA-ES-like algorithm from “first principles”. It “explains” three aspects of ES from a single principle:
  - optimization – update of  $\mathbf{m}$
  - step size control – update of  $\sigma = \sqrt[d]{\det(\mathbf{A})}$
  - shape control (CMA) – update of  $\mathbf{A}$  (or of  $\mathbf{A}/\sigma$ )

## Natural Evolution Strategies (NES)

- NES (Wierstra et al., 2008) is a CMA-ES-like algorithm from “first principles”. It “explains” three aspects of ES from a single principle:
  - optimization – update of  $\mathbf{m}$
  - step size control – update of  $\sigma = \sqrt[d]{\det(\mathbf{A})}$
  - shape control (CMA) – update of  $\mathbf{A}$  (or of  $\mathbf{A}/\sigma$ )
- However, it does not cover all aspects of CMA-ES:
  - noise-counteracting techniques such as cumulation

## Natural Evolution Strategies (NES)

- NES (Wierstra et al., 2008) is a CMA-ES-like algorithm from “first principles”. It “explains” three aspects of ES from a single principle:
  - optimization – update of  $\mathbf{m}$
  - step size control – update of  $\sigma = \sqrt[d]{\det(\mathbf{A})}$
  - shape control (CMA) – update of  $\mathbf{A}$  (or of  $\mathbf{A}/\sigma$ )
- However, it does not cover all aspects of CMA-ES:
  - noise-counteracting techniques such as cumulation
- A few more tricks are required to make it fly:
  - rank-based utilities replace fitness values
  - different learning rates for mean and covariance

## SNGD with Exponential Family

- Consider an exponential family

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^k \boldsymbol{\theta}_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta})\right)$$

with sufficient statistics  $\{T_i\}$ .

## SNGD with Exponential Family

- Consider an exponential family

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^k \boldsymbol{\theta}_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta})\right)$$

with sufficient statistics  $\{T_i\}$ .

- The derivative of the log density is simply

$$\frac{\partial \log(p(\mathbf{x}|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_i} = T_i(\mathbf{x}) - \mathbb{E}[T_i(\mathbf{x})] \ .$$

## SNGD with Exponential Family

- Consider an exponential family

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^k \boldsymbol{\theta}_i T_i(\mathbf{x}) - \psi(\boldsymbol{\theta})\right)$$

with sufficient statistics  $\{T_i\}$ .

- The derivative of the log density is simply

$$\frac{\partial \log(p(\mathbf{x}|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_i} = T_i(\mathbf{x}) - \mathbb{E}[T_i(\mathbf{x})] \ .$$

- Hence also the gradient has a simple form:

$$\text{Cov}_{\boldsymbol{\theta}}\left(\mathbf{T}(\mathbf{x}), W_f(\mathbf{x})\right) \ .$$

## SNGD with Exponential Family

- The Fisher matrix has entries

$$\mathcal{I}_{ij}(\boldsymbol{\theta}) = \text{Cov}_{\boldsymbol{\theta}} \left( T_i(\boldsymbol{x}), T_j(\boldsymbol{x}) \right) .$$

## SNGD with Exponential Family

- The Fisher matrix has entries

$$\mathcal{I}_{ij}(\boldsymbol{\theta}) = \text{Cov}_{\boldsymbol{\theta}} \left( \mathbf{T}_i(\mathbf{x}), \mathbf{T}_j(\mathbf{x}) \right) .$$

- The natural gradient can be expressed solely in terms of covariances:

$$\tilde{\nabla} W_f(\boldsymbol{\theta}) = \text{Cov}_{\boldsymbol{\theta}} \left( \mathbf{T}(\mathbf{x}), \mathbf{T}(\mathbf{x}) \right)^{-1} \text{Cov}_{\boldsymbol{\theta}} \left( \mathbf{T}(\mathbf{x}), W_f(\mathbf{x}) \right) .$$

(see Malagò et al., 2011)

## SNGD with Exponential Family

- Example: bitstrings  $\Omega = \{-1, +1\}^n$ .

## SNGD with Exponential Family

- Example: bitstrings  $\Omega = \{-1, +1\}^n$ .
- Then the probability simplex  $\Delta$  and hence the statistical manifold  $\mathcal{M}$  is finite dimensional.

## SNGD with Exponential Family

- Example: bitstrings  $\Omega = \{-1, +1\}^n$ .
- Then the probability simplex  $\Delta$  and hence the statistical manifold  $\mathcal{M}$  is finite dimensional.
- The sufficient statistics  $T_i(x)$  are square free monomials ( $x_i^2 = 1$ ).

## SNGD with Exponential Family

- Example: bitstrings  $\Omega = \{-1, +1\}^n$ .
- Then the probability simplex  $\Delta$  and hence the statistical manifold  $\mathcal{M}$  is finite dimensional.
- The sufficient statistics  $T_i(x)$  are square free monomials ( $x_i^2 = 1$ ).
- Each monomial characterizes a subset of bits the dependencies of which can be modeled.

## SNGD with Exponential Family

- Example: bitstrings  $\Omega = \{-1, +1\}^n$ .
- Then the probability simplex  $\Delta$  and hence the statistical manifold  $\mathcal{M}$  is finite dimensional.
- The sufficient statistics  $T_i(x)$  are square free monomials ( $x_i^2 = 1$ ).
- Each monomial characterizes a subset of bits the dependencies of which can be modeled.
- If the chosen model contains all interactions of variables in  $f$  then there is only one (global) optimum of  $W_f$ . The natural gradient will guide us there (see Malagò et al., 2011 for details).

## Information Geometric Optimization (IGO)

The **Information Geometric Optimization (IGO)** approach by Ollivier et al. introduces a unifying perspective:

The **Information Geometric Optimization (IGO)** approach by Ollivier et al. introduces a unifying perspective:

- emphasizes invariance properties as a means to reduce the number of arbitrary design choices,
- with a specific choice of  $W_f$  it explains the utility weights of NES from within the framework,
- it highlights the role of the gradient flow as the “pure form” of the EA, with the SNGD update being an approximation.

## Dynamic Stochastic Relaxation

- Expected fitness  $W_f(\xi) = \mathbb{E}_\xi[f(x)]$  is only one possible stochastic relaxation of  $f$ .

## Dynamic Stochastic Relaxation

- Expected fitness  $W_f(\xi) = \mathbb{E}_\xi[f(x)]$  is only one possible stochastic relaxation of  $f$ .
- We have already seen the generalization  $W_f(\xi) = \mathbb{E}_\xi[w(f(x))]$  with a transformation  $w$ .

## Dynamic Stochastic Relaxation

- Expected fitness  $W_f(\xi) = \mathbb{E}_\xi[f(x)]$  is only one possible stochastic relaxation of  $f$ .
- We have already seen the generalization  $W_f(\xi) = \mathbb{E}_\xi[w(f(x))]$  with a transformation  $w$ .
- In IGO the weight function depends on the  $f$ -quantile under the current distribution  $P_{\xi_0}$ :  $w(f(x)) = \tilde{w}(q_{\xi_0}^{-1}(f(x)))$ , where  $q_{\xi_0} : [0, 1] \rightarrow \mathbb{R}$  encodes the quantiles of the distribution of  $f(x)$ ,  $x \sim P_{\xi_0}$ .

## Dynamic Stochastic Relaxation

- Expected fitness  $W_f(\xi) = \mathbb{E}_\xi[f(x)]$  is only one possible stochastic relaxation of  $f$ .
- We have already seen the generalization  $W_f(\xi) = \mathbb{E}_\xi[w(f(x))]$  with a transformation  $w$ .
- In IGO the weight function depends on the  $f$ -quantile under the current distribution  $P_{\xi_0}$ :  $w(f(x)) = \tilde{w}(q_{\xi_0}^{-1}(f(x)))$ , where  $q_{\xi_0} : [0, 1] \rightarrow \mathbb{R}$  encodes the quantiles of the distribution of  $f(x)$ ,  $x \sim P_{\xi_0}$ .
- E.g.,  $q_{\xi_0}(1/2)$  is the median of  $f$ -values, and  $\tilde{w}(p) = 1$  for  $p < 1/2$  and  $\tilde{w}(p) = 0$  for  $p \geq 1/2$  encodes truncation (selection): only the better half of the distribution enters the update equation.

## Dynamic Stochastic Relaxation

- Expected fitness  $W_f(\xi) = \mathbb{E}_\xi[f(x)]$  is only one possible stochastic relaxation of  $f$ .
- We have already seen the generalization  $W_f(\xi) = \mathbb{E}_\xi[w(f(x))]$  with a transformation  $w$ .
- In IGO the weight function depends on the  $f$ -quantile under the current distribution  $P_{\xi_0}$ :  $w(f(x)) = \tilde{w}(q_{\xi_0}^{-1}(f(x)))$ , where  $q_{\xi_0} : [0, 1] \rightarrow \mathbb{R}$  encodes the quantiles of the distribution of  $f(x)$ ,  $x \sim P_{\xi_0}$ .
- E.g.,  $q_{\xi_0}(1/2)$  is the median of  $f$ -values, and  $\tilde{w}(p) = 1$  for  $p < 1/2$  and  $\tilde{w}(p) = 0$  for  $p \geq 1/2$  encodes truncation (selection): only the better half of the distribution enters the update equation.
- The dynamic choice of  $w = w(\xi_0)$  rescales  $f$  to a locally relevant range. It emphasizes local improvements relative to the current  $f$  distribution.

## Dynamic Stochastic Relaxation

- Benefit 1:  $W_f$  becomes invariant under rank-preserving transformations of fitness values.

## Dynamic Stochastic Relaxation

- Benefit 1:  $W_f$  becomes invariant under rank-preserving transformations of fitness values.
- Benefit 2: the rank-based utility weights  $u_i = \tilde{w}(q_{\xi_0}^{-1}(f(x_i)))$  of NES are obtained automatically in a principled manner.

## Dynamic Stochastic Relaxation

- Benefit 1:  $W_f$  becomes invariant under rank-preserving transformations of fitness values.
- Benefit 2: the rank-based utility weights  $u_i = \tilde{w}(q_{\xi_0}^{-1}(f(x_i)))$  of NES are obtained automatically in a principled manner.
- Drawback: the objective function  $W_f^{\xi_0}(\xi)$  becomes dependent on the current distribution  $\xi = \xi_0$

## Dynamic Stochastic Relaxation

- Benefit 1:  $W_f$  becomes invariant under rank-preserving transformations of fitness values.
- Benefit 2: the rank-based utility weights  $u_i = \tilde{w}(q_{\xi_0}^{-1}(f(x_i)))$  of NES are obtained automatically in a principled manner.
- Drawback: the objective function  $W_f^{\xi_0}(\xi)$  becomes dependent on the current distribution  $\xi = \xi_0$
- This means that the following situation may exist in principle:

$$W_f^{\xi_1}(\xi_2) < W_f^{\xi_1}(\xi_1)$$

$$W_f^{\xi_2}(\xi_3) < W_f^{\xi_2}(\xi_2)$$

$$W_f^{\xi_3}(\xi_1) < W_f^{\xi_3}(\xi_3)$$

and the “optimization” turns around in circles...

## Dynamic Stochastic Relaxation

- Benefit 1:  $W_f$  becomes invariant under rank-preserving transformations of fitness values.
- Benefit 2: the rank-based utility weights  $u_i = \tilde{w}(q_{\xi_0}^{-1}(f(x_i)))$  of NES are obtained automatically in a principled manner.
- Drawback: the objective function  $W_f^{\xi_0}(\xi)$  becomes dependent on the current distribution  $\xi = \xi_0$
- This means that the following situation may exist in principle:

$$W_f^{\xi_1}(\xi_2) < W_f^{\xi_1}(\xi_1)$$

$$W_f^{\xi_2}(\xi_3) < W_f^{\xi_2}(\xi_2)$$

$$W_f^{\xi_3}(\xi_1) < W_f^{\xi_3}(\xi_3)$$

and the “optimization” turns around in circles...

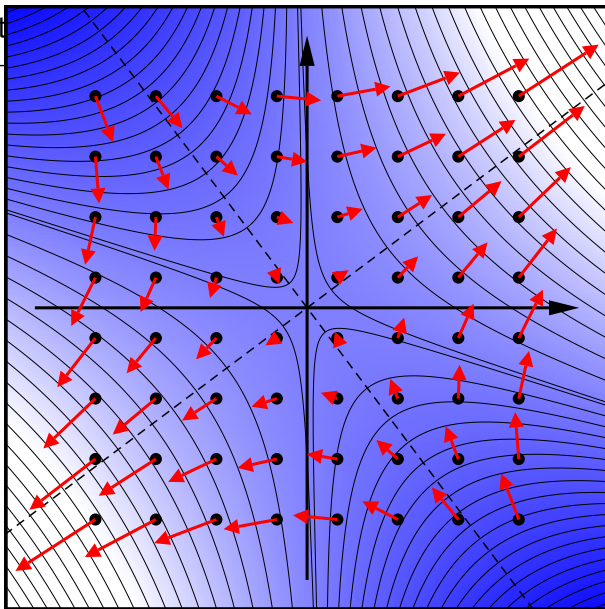
- Provably, in important special cases this does not happen.

## Vector Field, ODE, and Flow

- The natural gradient  $\widetilde{\nabla} W_f(\xi)$  defines the vector field  $V : \mathcal{M} \rightarrow T\mathcal{M}$  via  $V(\xi) = \widetilde{\nabla} W_f(\xi)$ .

# Vector Field, ODE, and Flow

- The nat  
 $V : \mathcal{M} \rightarrow$

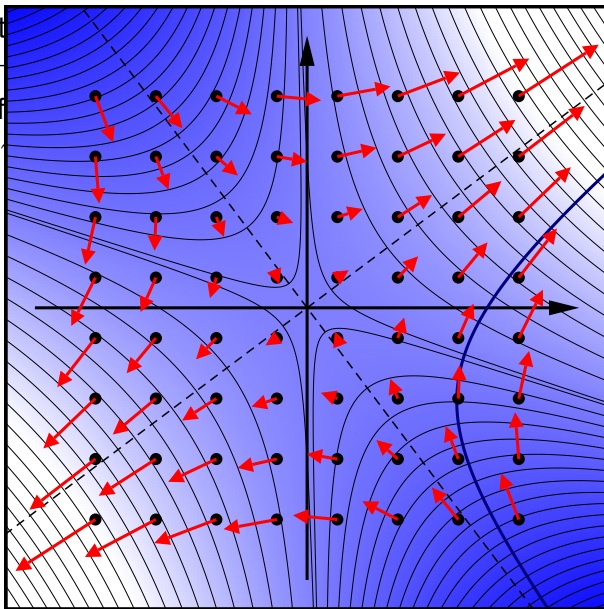


## Vector Field, ODE, and Flow

- The natural gradient  $\tilde{\nabla} W_f(\xi)$  defines the vector field  $V : \mathcal{M} \rightarrow T\mathcal{M}$  via  $V(\xi) = \tilde{\nabla} W_f(\xi)$ .
- Vector field  $\rightarrow$  differential equation  $\dot{\gamma}(t) = V(\gamma(t))$  with solution curves  $\gamma : \mathbb{R} \rightarrow \xi$ . Following these curves is optimization.

# Vector Field, ODE, and Flow

- The natural flow of a vector field  $V : \mathcal{M} \rightarrow T\mathcal{M}$  is a family of curves
- Vector field curves



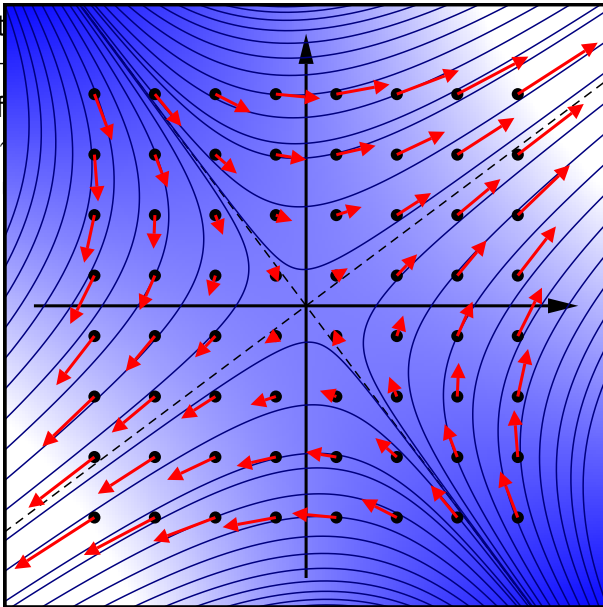
solution  
n.

## Vector Field, ODE, and Flow

- The natural gradient  $\tilde{\nabla} W_f(\xi)$  defines the vector field  $V : \mathcal{M} \rightarrow T\mathcal{M}$  via  $V(\xi) = \tilde{\nabla} W_f(\xi)$ .
- Vector field  $\rightarrow$  differential equation  $\dot{\gamma}(t) = V(\gamma(t))$  with solution curves  $\gamma : \mathbb{R} \rightarrow \xi$ . Following these curves is optimization.
- The solution curves are collected in the flow  $\xi_t = \varphi(\xi, t)$ .

# Vector Field, ODE, and Flow

- The nat
- $V : \mathcal{M} \rightarrow \mathbb{R}^n$
- Vector f
- curves
- The sol



solution

h.

).

## Vector Field, ODE, and Flow

- The natural gradient  $\tilde{\nabla} W_f(\xi)$  defines the vector field  $V : \mathcal{M} \rightarrow T\mathcal{M}$  via  $V(\xi) = \tilde{\nabla} W_f(\xi)$ .
- Vector field  $\rightarrow$  differential equation  $\dot{\gamma}(t) = V(\gamma(t))$  with solution curves  $\gamma : \mathbb{R} \rightarrow \xi$ . Following these curves is optimization.
- The solution curves are collected in the flow  $\xi_t = \varphi(\xi, t)$ .
- Note 1: just like the natural gradient itself this flow is deterministic. This is achieved in the limit of infinite samples in the MC approximation, corresponding to infinite population size.

## Vector Field, ODE, and Flow

- The natural gradient  $\tilde{\nabla} W_f(\xi)$  defines the vector field  $V : \mathcal{M} \rightarrow T\mathcal{M}$  via  $V(\xi) = \tilde{\nabla} W_f(\xi)$ .
- Vector field  $\rightarrow$  differential equation  $\dot{\gamma}(t) = V(\gamma(t))$  with solution curves  $\gamma : \mathbb{R} \rightarrow \xi$ . Following these curves is optimization.
- The solution curves are collected in the flow  $\xi_t = \varphi(\xi, t)$ .
- Note 1: just like the natural gradient itself this flow is deterministic. This is achieved in the limit of infinite samples in the MC approximation, corresponding to infinite population size.
- Note 2: in each point the flow moves tangential to the vector field. This corresponds to re-evaluating the gradient after an infinitesimal step, or to an infinitesimal learning rate in the gradient descent procedure.

- An SNGD algorithm is a two-fold approximation of the flow:
  - it discretizes time and performs Euler steps,
  - it relies on a stochastic gradient based on sampling.

## SNGD Algorithms

- An SNGD algorithm is a two-fold approximation of the flow:
  - it discretizes time and performs Euler steps,
  - it relies on a stochastic gradient based on sampling.
- NES is a rather pure example of an such an algorithm.

## SNGD Algorithms

- An SNGD algorithm is a two-fold approximation of the flow:
  - it discretizes time and performs Euler steps,
  - it relies on a stochastic gradient based on sampling.
- NES is a rather pure example of an such an algorithm.
- Surprisingly many established algorithms are closely connected to SNGD (or IGO) algorithms.

- An SNGD algorithm is a two-fold approximation of the flow:
  - it discretizes time and performs Euler steps,
  - it relies on a stochastic gradient based on sampling.
- NES is a rather pure example of an such an algorithm.
- Surprisingly many established algorithms are closely connected to SNGD (or IGO) algorithms.
- New perspective: EA approximates the flow, hence the flow is an idealized EA.

## Connection to CMA-ES

- CMA-ES mean update:

$$\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \cdot \mathbf{x}_i \ .$$

## Connection to CMA-ES

- CMA-ES mean update:

$$\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \cdot \mathbf{x}_i \ .$$

- CMA-ES rank- $\mu$  covariance matrix update:

$$\mathbf{C} \leftarrow (1 - \gamma_{\mathbf{C}}) \cdot \mathbf{C} + \gamma_{\mathbf{C}} \cdot \sum_{i=1}^{\mu} w_i \cdot (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \ .$$

## Connection to CMA-ES

- Both equations can be written as updates

$$\mathbf{m} \leftarrow \mathbf{m} + \gamma_{\mathbf{m}} \cdot \sum_{i=1}^N w_i \cdot (\mathbf{x}_i - \mathbf{m})$$

$$\mathbf{C} \leftarrow \mathbf{C} + \gamma_{\mathbf{C}} \cdot \sum_{i=1}^N w_i \cdot \left( (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T - \mathbf{C} \right)$$

with fixed learning rate  $\gamma_{\mathbf{m}} = 1$  and  $w_i = 0$  for  $i > \mu$ .

## Connection to CMA-ES

- Both equations can be written as updates

$$\mathbf{m} \leftarrow \mathbf{m} + \gamma_{\mathbf{m}} \cdot \sum_{i=1}^N w_i \cdot (\mathbf{x}_i - \mathbf{m})$$

$$\mathbf{C} \leftarrow \mathbf{C} + \gamma_{\mathbf{C}} \cdot \sum_{i=1}^N w_i \cdot \left( (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T - \mathbf{C} \right)$$

with fixed learning rate  $\gamma_{\mathbf{m}} = 1$  and  $w_i = 0$  for  $i > \mu$ .

- The change of coordinates  $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ ,  $\mathbf{x} = \mathbf{A}\mathbf{z} + \mathbf{m}$  reveals:

$$\mathbf{m} \leftarrow \mathbf{m} + \gamma_{\mathbf{m}} \cdot \mathbf{A} \cdot \sum_{i=1}^N w_i \cdot \mathbf{z}_i$$

$$\mathbf{C} \leftarrow \mathbf{C} + \gamma_{\mathbf{C}} \cdot \mathbf{A} \cdot \left( \sum_{i=1}^N w_i \cdot (\mathbf{z}_i \mathbf{z}_i^T - \mathbf{I}) \right) \cdot \mathbf{A}^T$$

## Connection to CMA-ES

- Both equations can be written as updates

$$\mathbf{m} \leftarrow \mathbf{m} + \gamma_{\mathbf{m}} \cdot \sum_{i=1}^N w_i \cdot (\mathbf{x}_i - \mathbf{m})$$

$$\mathbf{C} \leftarrow \mathbf{C} + \gamma_{\mathbf{C}} \cdot \sum_{i=1}^N w_i \cdot \left( (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T - \mathbf{C} \right)$$

with fixed learning rate  $\gamma_{\mathbf{m}} = 1$  and  $w_i = 0$  for  $i > \mu$ .

- The change of coordinates  $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ ,  $\mathbf{x} = \mathbf{A}\mathbf{z} + \mathbf{m}$  reveals:

$$\mathbf{m} \leftarrow \mathbf{m} + \gamma_{\mathbf{m}} \cdot \mathbf{A} \cdot \sum_{i=1}^N w_i \cdot \mathbf{z}_i$$

$$\mathbf{C} \leftarrow \mathbf{C} + \gamma_{\mathbf{C}} \cdot \mathbf{A} \cdot \left( \sum_{i=1}^N w_i \cdot (\mathbf{z}_i \mathbf{z}_i^T - \mathbf{I}) \right) \cdot \mathbf{A}^T$$

- This is essentially the IGO/NES SGD update (see Akimoto 2010).

## Connection to Maximum Likelihood Estimation

- The CMA-ES update equations can be written as

$$\mathbf{m} \leftarrow (1 - \gamma_{\mathbf{m}}) \cdot \mathbf{m} + \gamma_{\mathbf{m}} \cdot \hat{\mathbf{m}}_{\text{ML}} ,$$

$$\mathbf{C} \leftarrow (1 - \gamma_{\mathbf{C}}) \cdot \mathbf{C} + \gamma_{\mathbf{C}} \cdot \hat{\mathbf{C}}_{\text{ML}} .$$

## Connection to Maximum Likelihood Estimation

- The CMA-ES update equations can be written as

$$\mathbf{m} \leftarrow (1 - \gamma_{\mathbf{m}}) \cdot \mathbf{m} + \gamma_{\mathbf{m}} \cdot \hat{\mathbf{m}}_{\text{ML}} ,$$

$$\mathbf{C} \leftarrow (1 - \gamma_{\mathbf{C}}) \cdot \mathbf{C} + \gamma_{\mathbf{C}} \cdot \hat{\mathbf{C}}_{\text{ML}} .$$

- $\hat{\mathbf{m}}_{\text{ML}} = \sum_{i=1}^N w_i \cdot \mathbf{x}_i$  is the weighted Maximum Likelihood (ML) estimator of  $\mathbf{m}$ .

## Connection to Maximum Likelihood Estimation

- The CMA-ES update equations can be written as

$$\mathbf{m} \leftarrow (1 - \gamma_{\mathbf{m}}) \cdot \mathbf{m} + \gamma_{\mathbf{m}} \cdot \hat{\mathbf{m}}_{\text{ML}} ,$$

$$\mathbf{C} \leftarrow (1 - \gamma_{\mathbf{C}}) \cdot \mathbf{C} + \gamma_{\mathbf{C}} \cdot \hat{\mathbf{C}}_{\text{ML}} .$$

- $\hat{\mathbf{m}}_{\text{ML}} = \sum_{i=1}^N w_i \cdot \mathbf{x}_i$  is the weighted Maximum Likelihood (ML) estimator of  $\mathbf{m}$ .
- The term

$$\hat{\mathbf{C}}_{\text{ML}} = \sum_{i=1}^{\mu} w_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

is the weighted ML estimator of  $\mathbf{C}$ , *provided that  $\mathbf{m}$  remains fixed.*

## Further Connections

- A variety of further algorithms fits into the framework.

## Further Connections

- A variety of further algorithms fits into the framework.
- Among them are methods for discrete and continuous search spaces.

## Further Connections

- A variety of further algorithms fits into the framework.
- Among them are methods for discrete and continuous search spaces.
- Sometimes only the essential principle fits in exactly, i.e., the algorithm needs a little simplification or “clean-up” in order to fit.

## Further Connections

- A variety of further algorithms fits into the framework.
- Among them are methods for discrete and continuous search spaces.
- Sometimes only the essential principle fits in exactly, i.e., the algorithm needs a little simplification or “clean-up” in order to fit.
- We refer to Ollivier 2011, Akimoto 2013, and Malagò et al. 2013 for examples.

## Convergence Results

- Establishing convergence of realistic EAs can be a hard task.

## Convergence Results

- Establishing convergence of realistic EAs can be a hard task.
- Benefit of gradient flow over EA: easier to analyze.

## Convergence Results

- Establishing convergence of realistic EAs can be a hard task.
- Benefit of gradient flow over EA: easier to analyze.
- Question: Do all flow trajectories converge to the optimum?

## Convergence Results

- Establishing convergence of realistic EAs can be a hard task.
- Benefit of gradient flow over EA: easier to analyze.
- Question: Do all flow trajectories converge to the optimum?
- More formally, let  $\delta_{x^*}$  denote the Dirac peak over an (isolated) optimum  $x^* \in \Omega$ . Does it hold  $\lim_{t \rightarrow \infty} P_{\xi_t} = \delta_{x^*}$  for all initial conditions  $\xi$ ?

## Convergence Results

- Establishing convergence of realistic EAs can be a hard task.
- Benefit of gradient flow over EA: easier to analyze.
- Question: Do all flow trajectories converge to the optimum?
- More formally, let  $\delta_{x^*}$  denote the Dirac peak over an (isolated) optimum  $x^* \in \Omega$ . Does it hold  $\lim_{t \rightarrow \infty} P_{\xi_t} = \delta_{x^*}$  for all initial conditions  $\xi$ ?
- Note: convergence of the flow does not directly imply convergence of stochastic approximate algorithms!

## Representation of the Optimum

- Hidden prerequisite: the statistical model must be sufficiently rich to focus the probability mass on optima  $\Omega^* \subset \Omega$ .

## Representation of the Optimum

- Hidden prerequisite: the statistical model must be sufficiently rich to focus the probability mass on optima  $\Omega^* \subset \Omega$ .
- This is not a prerequisite for convergence of the flow to an optimal distribution *within* (the closure of) the statistical model.

## Representation of the Optimum

- Hidden prerequisite: the statistical model must be sufficiently rich to focus the probability mass on optima  $\Omega^* \subset \Omega$ .
- This is not a prerequisite for convergence of the flow to an optimal distribution *within* (the closure of) the statistical model.
- However, it is a prerequisite for convergence to an optimal distribution, possibly outside the closure of the family, and hence to an optimum of the original problem  $\min_{x \in \Omega} f(x)$ .

## Representation of the Optimum

- Hidden prerequisite: the statistical model must be sufficiently rich to focus the probability mass on optima  $\Omega^* \subset \Omega$ .
- This is not a prerequisite for convergence of the flow to an optimal distribution *within* (the closure of) the statistical model.
- However, it is a prerequisite for convergence to an optimal distribution, possibly outside the closure of the family, and hence to an optimum of the original problem  $\min_{x \in \Omega} f(x)$ .
- In the discrete case the optimum of the stochastically relaxed problem describes an optimum of  $f : \Omega \rightarrow \mathbb{R}$  iff a subset of  $S \subset \Omega^*$  corresponds to an *exposed face*  $A$  of the marginal polytope, i.e., if  $S = T^{-1}(A) \subset \Omega^*$ .

## Representation of the Optimum

- Hidden prerequisite: the statistical model must be sufficiently rich to focus the probability mass on optima  $\Omega^* \subset \Omega$ .
- This is not a prerequisite for convergence of the flow to an optimal distribution *within* (the closure of) the statistical model.
- However, it is a prerequisite for convergence to an optimal distribution, possibly outside the closure of the family, and hence to an optimum of the original problem  $\min_{x \in \Omega} f(x)$ .
- In the discrete case the optimum of the stochastically relaxed problem describes an optimum of  $f : \Omega \rightarrow \mathbb{R}$  iff a subset of  $S \subset \Omega^*$  corresponds to an *exposed face*  $A$  of the marginal polytope, i.e., if  $S = T^{-1}(A) \subset \Omega^*$ .
- Gaussians contain Dirac peaks in the limit.

## Convergence Results

### Convergence of IGO flow:

Theorem (Akimoto et al. 2012, Glasmachers 2012)

*Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  be a strictly convex quadratic function with minimum  $x^*$ . Consider the class  $\mathcal{N}(\mathbf{m}, \sigma^2)$  of isotropic Gaussian search distributions. Then all trajectories of the IGO flow converge to the boundary point  $\mathbf{m} = x^*$  and  $\sigma^2 = 0$  (corresponding to  $\delta_{x^*}$ ).*

## Convergence Results

### Convergence of IGO flow:

Theorem (Akimoto et al. 2012, Glasmachers 2012)

*Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  be a strictly convex quadratic function with minimum  $x^*$ . Consider the class  $\mathcal{N}(\mathbf{m}, \sigma^2)$  of isotropic Gaussian search distributions. Then all trajectories of the IGO flow converge to the boundary point  $\mathbf{m} = x^*$  and  $\sigma^2 = 0$  (corresponding to  $\delta_{x^*}$ ).*

Corollary (Akimoto et al. 2012)

*The same holds for monotonically transformed functions.*

## Convergence Results

### Convergence of IGO flow:

Theorem (Akimoto et al. 2012, Glasmachers 2012)

*Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  be a strictly convex quadratic function with minimum  $x^*$ . Consider the class  $\mathcal{N}(\mathbf{m}, \sigma^2)$  of isotropic Gaussian search distributions. Then all trajectories of the IGO flow converge to the boundary point  $\mathbf{m} = x^*$  and  $\sigma^2 = 0$  (corresponding to  $\delta_{x^*}$ ).*

Corollary (Akimoto et al. 2012)

*The same holds for monotonically transformed functions.*

Corollary (Akimoto et al. 2012)

*The same holds in the vicinity of any twice continuously differentiable local optimum.*

## Convergence Results with CMA

Convergence of IGO flow with *fixed* reference distribution:

Theorem (Akimoto 2012)

*Consider  $f(x) = x^T Q x$  with strictly positive definite matrix  $Q$  and multivariate Gaussian search distributions  $\mathcal{N}(\mathbf{m}, \mathbf{C})$ . Then it holds*

$$\mathbf{m} \rightarrow \mathbf{x}^* \quad \mathbf{C} \rightarrow \mathbf{Q}^{-1} .$$

## Convergence Results with CMA

Convergence of gradient flow of  $\mathbb{E}[f]$ :

Lemma (Beyer 2014)

*For multivariate Gaussians  $\mathcal{N}(\mathbf{m}, \mathbf{C})$  on a convex quadratic objective  $f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$  the gradient flow is defined by the differential equation*

$$\begin{aligned}\frac{d\mathbf{m}(t)}{dt} &= -2 \mathbf{C}(t) \mathbf{Q} \mathbf{m}(t) \\ \frac{d\mathbf{C}(t)}{dt} &= -2 \mathbf{C}(t) \mathbf{Q} \mathbf{C}(t)\end{aligned}$$

## Convergence Results with CMA

Convergence of gradient flow of  $\mathbb{E}[f]$ :

Lemma (Beyer 2014)

*For multivariate Gaussians  $\mathcal{N}(\mathbf{m}, \mathbf{C})$  on a convex quadratic objective  $f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$  the gradient flow is defined by the differential equation*

$$\begin{aligned}\frac{d\mathbf{m}(t)}{dt} &= -2 \mathbf{C}(t) \mathbf{Q} \mathbf{m}(t) \\ \frac{d\mathbf{C}(t)}{dt} &= -2 \mathbf{C}(t) \mathbf{Q} \mathbf{C}(t)\end{aligned}$$

A similar result was found by Akimoto for the IGO flow with fixed reference distribution.

## Convergence Results with CMA

Convergence of gradient flow of  $\mathbb{E}[f]$ :

Theorem (Beyer 2014)

*The non-linear ordinary differential equation (ODE) system*

$$\frac{d\mathbf{C}(t)}{dt} = -2 \mathbf{C}(t) \mathbf{Q} \mathbf{C}(t)$$

*with initial condition  $\mathbf{C}(0) = \mathbf{C}_0$  has the solution*

$$\mathbf{C}(t) = (\mathbf{C}_0^{-1} + 2t\mathbf{Q})^{-1} .$$

## Convergence Results with CMA

Convergence of gradient flow of  $\mathbb{E}[f]$ :

Theorem (Beyer 2014)

*The non-linear ordinary differential equation (ODE) system*

$$\frac{d\mathbf{C}(t)}{dt} = -2 \mathbf{C}(t) \mathbf{Q} \mathbf{C}(t)$$

*with initial condition  $\mathbf{C}(0) = \mathbf{C}_0$  has the solution*

$$\mathbf{C}(t) = (\mathbf{C}_0^{-1} + 2t\mathbf{Q})^{-1} .$$

$$\Rightarrow \mathbf{C}(t) \rightarrow \mathbf{Q}^{-1}$$

$$\Rightarrow \|\mathbf{C}(t)\mathbf{Q}^{-1}\| \in \mathcal{O}(1/t)$$

## Convergence Results with CMA

Convergence of gradient flow of  $\mathbb{E}[f]$ :

Theorem (Beyer 2014)

*The non-linear ordinary differential equation (ODE) system*

$$\frac{d\mathbf{C}(t)}{dt} = -2 \mathbf{C}(t) \mathbf{Q} \mathbf{C}(t)$$

*with initial condition  $\mathbf{C}(0) = \mathbf{C}_0$  has the solution*

$$\mathbf{C}(t) = (\mathbf{C}_0^{-1} + 2t\mathbf{Q})^{-1} .$$

$$\Rightarrow \mathbf{C}(t) \rightarrow \mathbf{Q}^{-1}$$

$$\Rightarrow \|\mathbf{C}(t)\mathbf{Q}^{-1}\| \in \mathcal{O}(1/t)$$

Beyer also obtains  $\|\mathbf{m}(t)\| \in \mathcal{O}(1/t)$ .

## Convergence Results with CMA

Convergence of IGO flow with IGO objective:

Theorem (Beyer 2014)

*Under the assumption of (approximate) normality of fitness values the dynamics of IGO (with quantile-based objective) are*

$$\mathbf{m}(t) \approx \alpha \cdot \exp\left(-\sqrt{2/d} \cdot t\right) \cdot \mathbf{Q}^{-1} \mathbf{C}_0^{-1} \mathbf{m}_0 ,$$

$$\mathbf{C}(t) \approx \alpha \cdot \exp\left(-\sqrt{2/d} \cdot t\right) \cdot \mathbf{Q}^{-1} .$$

The flow converges at a linear rate, which is what we'd expect for an evolution strategy.

## Convergence Results – Summary

- Proofs for the discrete case and for isotropic and general multi-variate Gaussians.

## Convergence Results – Summary

- Proofs for the discrete case and for isotropic and general multi-variate Gaussians.
- Continuous case results are restricted to quadratic functions. This models convergence to twice differentiable local optima.

## Convergence Results – Summary

- Proofs for the discrete case and for isotropic and general multi-variate Gaussians.
- Continuous case results are restricted to quadratic functions. This models convergence to twice differentiable local optima.
- No results for more general problem classes like all convex problems.

## Convergence Results – Summary

- Proofs for the discrete case and for isotropic and general multi-variate Gaussians.
- Continuous case results are restricted to quadratic functions. This models convergence to twice differentiable local optima.
- No results for more general problem classes like all convex problems.
- Note once more: convergence results for the gradient flow do not imply convergence of the EA.

## Convergence Results – Summary

- Proofs for the discrete case and for isotropic and general multi-variate Gaussians.
- Continuous case results are restricted to quadratic functions. This models convergence to twice differentiable local optima.
- No results for more general problem classes like all convex problems.
- Note once more: convergence results for the gradient flow do not imply convergence of the EA.
- The deviations of the algorithm from the flow due of stochasticity (finite populations) and finite step sizes (discrete time) are yet to be understood.



**But honestly – does it really work?**

### **But honestly – does it really work?**

- Many EAs apply update equations that can be explained from information geometry.

### **But honestly – does it really work?**

- Many EAs apply update equations that can be explained from information geometry.
- However, statistical models and stochastic relaxations do not and will probably never cover all aspects of EAs.

### **But honestly – does it really work?**

- Many EAs apply update equations that can be explained from information geometry.
- However, statistical models and stochastic relaxations do not and will probably never cover all aspects of EAs.
- Realistic EAs can be built from (at least) two types of components:
  - update equations derived from information geometry,
  - other (classic) tools for handling stochasticity.

### **But honestly – does it really work?**

- Many EAs apply update equations that can be explained from information geometry.
- However, statistical models and stochastic relaxations do not and will probably never cover all aspects of EAs.
- Realistic EAs can be built from (at least) two types of components:
  - update equations derived from information geometry,
  - other (classic) tools for handling stochasticity.
- Each component must do its job.

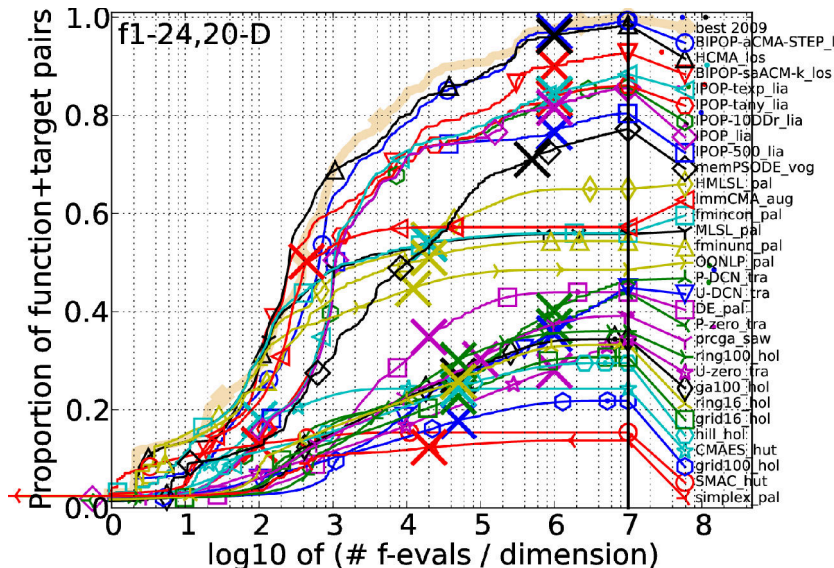
### **But honestly – does it really work?**

- Many EAs apply update equations that can be explained from information geometry.
- However, statistical models and stochastic relaxations do not and will probably never cover all aspects of EAs.
- Realistic EAs can be built from (at least) two types of components:
  - update equations derived from information geometry,
  - other (classic) tools for handling stochasticity.
- Each component must do its job.
- Information geometrical updates generally do a great job at improving the search distribution, provided that stochastic effects are sufficiently well controlled.

### **But honestly – does it really work?**

As far as the above outlined role of information geometry in EAs is concerned the clear answer is:

**Yes, it works great!**



source: N. Hansen, A few overview results from the GECCO BBOB workshops



## Summary

- Information geometry provides update equations for optimization from first principles.

## Summary

- Information geometry provides update equations for optimization from first principles.
- This often amounts to SNGD applied to a stochastically relaxed problem.

## Summary

- Information geometry provides update equations for optimization from first principles.
- This often amounts to SNGD applied to a stochastically relaxed problem.
- This is an approximation to an optimization flow.

## Summary

- Information geometry provides update equations for optimization from first principles.
- This often amounts to SNGD applied to a stochastically relaxed problem.
- This is an approximation to an optimization flow.
- It can help the analysis of existing algorithms like CMA-ES, EDAs, and model-based optimization in general.

## Summary

- Information geometry provides update equations for optimization from first principles.
- This often amounts to SNGD applied to a stochastically relaxed problem.
- This is an approximation to an optimization flow.
- It can help the analysis of existing algorithms like CMA-ES, EDAs, and model-based optimization in general.
- It is a generic design principle for optimization algorithms on any search space and for many families of search distributions.

## Summary

- Information geometry provides update equations for optimization from first principles.
- This often amounts to SNGD applied to a stochastically relaxed problem.
- This is an approximation to an optimization flow.
- It can help the analysis of existing algorithms like CMA-ES, EDAs, and model-based optimization in general.
- It is a generic design principle for optimization algorithms on any search space and for many families of search distributions.
- Dedicated algorithms such as NES were built on this principle.

## Summary

- Information geometry provides update equations for optimization from first principles.
- This often amounts to SNGD applied to a stochastically relaxed problem.
- This is an approximation to an optimization flow.
- It can help the analysis of existing algorithms like CMA-ES, EDAs, and model-based optimization in general.
- It is a generic design principle for optimization algorithms on any search space and for many families of search distributions.
- Dedicated algorithms such as NES were built on this principle.
- Algorithms respecting the information geometry of their search distributions are among the top performers.

## Summary

- However, randomization is outside the framework. It must be controlled by other means.

## Summary

- However, randomization is outside the framework. It must be controlled by other means.
- Information geometric tools must be augmented with “orthogonal” tools for control of stochastic effects—together they provide a modern perspective on EA research.

## Summary

- However, randomization is outside the framework. It must be controlled by other means.
- Information geometric tools must be augmented with “orthogonal” tools for control of stochastic effects—together they provide a modern perspective on EA research.
- The same problem decomposition is a promising route for theoretical analysis: the gradient flow is becoming a well-investigated object, while more traditional tools (Markov chain analysis, etc.) may be necessary to connect it to real EAs.

# References

- 1 Akimoto et al. Bidirectional relation between CMA evolution strategies and natural evolution strategies. *Parallel Problem Solving from Nature (PPSN) XI* 2010.
- 2 Akimoto et al. Convergence of the Continuous Time Trajectory of Isotropic Evolution Strategies on Monotonic  $C^2$ -composite Functions. *Parallel Problem Solving from Nature (PPSN) XII*, 2012.
- 3 Beyer. Convergence Analysis of Evolutionary Algorithms which are Based on the Paradigm of Information Geometry. *Evolutionary Computation*, 2014.
- 4 Glasmachers et al. Exponential Natural Evolution Strategies. *Genetic and Evolutionary Computation Conference (GECCO)*, 2010.
- 5 Glasmachers. Convergence of the IGO-Flow of Isotropic Gaussian Distributions on Convex Quadratic Problems. *Parallel Problem Solving from Nature (PPSN) XII*, 2012.
- 6 Malagò et al. Natural Gradient, Fitness Modelling and Model Selection: A Unifying Perspective. *Congress on Evolutionary Computation (CEC)*, 2013.
- 7 Malagò et al. Towards the geometry of estimation of distribution algorithms based on the exponential family. *FOGA*, 2011.
- 8 Ollivier et al. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *arXiv:1106.3708*, 2011.
- 9 Wierstra et al. Natural Evolution Strategies. *Congress on Evolutionary Computation (CEC)*, 2008.



Thank you!