Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles

Ludovic Arnold, Anne Auger, Nikolaus Hansen, Yann Ollivier

Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles

Ludovic Arnold, Anne Auger, Nikolaus Hansen, Yann Ollivier

Abstract

We present a canonical way to turn any smooth parametric family of probability distributions on an arbitrary search space X into a continuous-time black-box optimization method on X, the *information-geometric optimization* (IGO) method. Invariance as a major design principle keeps the number of arbitrary choices to a minimum. The resulting method conducts a natural gradient ascent using an adaptive, time-dependent transformation of the objective function, and makes no particular assumptions on the objective function to be optimized.

The IGO method produces explicit IGO algorithms through time discretization. The cross-entropy method is recovered in a particular case with a large time step, and can be extended into a smoothed, parametrization-independent maximum likelihood update.

When applied to specific families of distributions on discrete or continuous spaces, the IGO framework allows to naturally recover versions of known algorithms. From the family of Gaussian distributions on $\mathbb{R}^d$, we arrive at a version of the well-known CMA-ES algorithm. From the family of Bernoulli distributions on $\{0,1\}^d$, we recover the seminal PBIL algorithm. From the distributions of restricted Boltzmann machines, we naturally obtain a novel algorithm for discrete optimization on $\{0,1\}^d$.

The IGO method achieves, thanks to its intrinsic formulation, maximal invariance properties: invariance under reparametrization of the search space X, under a change of parameters of the probability distribution, and under increasing transformation of the function to be optimized. The latter is achieved thanks to an adaptive formulation of the objective.

Theoretical considerations strongly suggest that IGO algorithms are characterized by a minimal change of the distribution. Therefore they have minimal loss in diversity through the course of optimization, provided the initial diversity is high. First experiments using restricted Boltzmann machines confirm this insight. As a simple consequence, IGO seems to provide, from information theory, an elegant way to spontaneously explore several valleys of a fitness landscape in a single run.

Contents

Introduction

In this article, we consider an objective function $f : X \to \mathbb{R}$ to be minimized over a search space X. No particular assumption on X is needed: it may be discrete or continuous, finite or infinite. We adopt a standard scenario where we consider f as a *black box* that delivers values $f(x)$ for any desired input $x \in X$. The objective of black-box optimization is to find solutions $x \in X$ with small value $f(x)$, using the least number of calls to the black box. In this context, we design a stochastic optimization method from sound theoretical principles.

We assume that we are given a family of probability distributions P_θ on X depending on a continuous multicomponent parameter $\theta \in \Theta$. A basic example is to take $X = \mathbb{R}^d$ and to consider the family of all Gaussian distributions P_θ on $\mathbb{R}^d$, with $\theta = (m, C)$ the mean and covariance matrix. Another simple example is $X = \{0, 1\}^d$ equipped with the family of Bernoulli measures, i.e. $\theta = (\theta_i)_{1 \leqslant i \leqslant d}$ and $P_\theta(x) = \prod \theta_i^{x_i}(1 - \theta_i)^{1-x_i}$ for $x = (x_i) \in X$. The parameters θ of the family P_θ belong to a space, Θ, assumed to be a smooth manifold.

From this setting, we build in a natural way an optimization method, the *information-geometric optimization* (IGO). At each time t, we maintain a value θ^t such that P_{θ^t} represents, loosely speaking, the current belief about where the smallest values of the function f may lie. Over time, P_{θ^t} evolves and is expected to concentrate around the minima of f. This general approach resembles the wide family of *estimation of distribution algorithms* (EDA) [LL02, BC95, PGL02]. However, we deviate somewhat from the common EDA reasoning, as explained in the following.

The IGO method takes the form of a gradient ascent on θ^t in the parameter space Θ. We follow the gradient of a suitable transformation of f, based on the P_{θ^t}*-quantiles* of f. The gradient used for θ is the *natural gradient* defined from the Fisher information metric [Rao45, Jef46, AN00], as is the case in various other optimization strategies, for instance so-called *natural evolution strategies* [WSPS08, SWSS09, GSS+10]. Thus, we extend the scope of optimization strategies based on this gradient to arbitrary search spaces.

The IGO method also has an equivalent description as an infinitesimal maximum likelihood update; this reveals a new property of the natural gradient. This also relates IGO to the *cross-entropy method* [dBKMR05] in some situations.

When we instantiate IGO using the family of Gaussian distributions on $\mathbb{R}^d$, we naturally obtain variants of the well-known *covariance matrix adaptation evolution strategy* (CMA-ES) [HO01, HK04, JA06] and of *natural evolution strategies*. With Bernoulli measures on the discrete cube $\{0,1\}^d$, we recover the well-known *population based incremental learning* (PBIL) [BC95, Bal94]; this derivation of PBIL as a natural gradient ascent appears to be new, and sheds some light on the common ground between continuous and discrete optimization.

From the IGO framework, it is immediate to build new optimization algorithms using more complex families of distributions than Gaussian or Bernoulli. As an illustration, distributions associated with restricted Boltzmann machines (RBMs) provide a new but natural algorithm for discrete optimization on $\{0,1\}^d$, able to handle dependencies between the bits (see also [Ber02]). The probability distributions associated with RBMs are multimodal; combined with specific information-theoretic properties of IGO that guarantee minimal loss of diversity over time, this allows IGO to reach multiple optima at once very naturally, at least in a simple experimental setup.

Our method is built to achieve maximal *invariance properties*. First, it will be invariant under reparametrization of the family of distributions P_θ, that is, at least for infinitesimally small steps, it only depends on P_θ and not on the way we write the parameter θ. (For instance, for Gaussian measures it should not matter whether we use the covariance matrix or its inverse or a Cholesky factor as the parameter.) This limits the influence of encoding choices on the behavior of the algorithm. Second, it will be invariant under a change of coordinates in the search space X, provided that this change of coordinates globally preserves our family of distributions P_θ. (For Gaussian distributions on $\mathbb{R}^d$, this includes all affine changes of coordinates.) This means that the algorithm, apart from initialization, does not depend on the precise way the data is presented. Last, the algorithm will be invariant under applying an increasing function to f, so that it is indifferent whether we minimize e.g. f, f^3 or $f \times |f|^{-2/3}$. This way some non-convex or non-smooth functions can be as "easily" optimised as convex ones. Contrary to previous formulations using natural gradients [WSPS08, GSS+10, ANOK10],

this invariance under increasing transformation of the objective function is achieved from the start.

Invariance under X-reparametrization has been—we believe—one of the keys to the success of the CMA-ES algorithm, which derives from a particular case of ours. Invariance under θ-reparametrization is the main idea behind *information geometry* [AN00]. Invariance under f-transformation is not uncommon, e.g., for evolution strategies [Sch95] or pattern search methods [HJ61, Tor97, NM65]; however it is not always recognized as an attractive feature. Such invariance properties mean that we deal with *intrinsic* properties of the objects themselves, and not with the way we encode them as collections of numbers in $\mathbb{R}^d$. It also means, most importantly, that we make a minimal number of arbitrary choices.

In Section 1, we define the IGO flow and the IGO algorithm. We begin with standard facts about the definition and basic properties of the natural gradient, and its connection with Kullback–Leibler divergence and diversity. We then proceed to the detailed description of our algorithm.

In Section 2, we state some first mathematical properties of IGO. These include monotone improvement of the objective function, invariance properties, the form of IGO for exponential families of probability distributions, and the case of noisy objective functions.

In Section 3 we explain the theoretical relationships between IGO, maximum likelihood estimates and the cross-entropy method. In particular, IGO is uniquely characterized by a weighted log-likelihood maximization property.

In Section 4, we derive several well-known optimization algorithms from the IGO framework. These include PBIL, versions of CMA-ES and other Gaussian evolutionary algorithms such as EMNA. This also illustrates how a large step size results in more and more differing algorithms w.r.t. the continuous-time IGO flow.

In Section 5, we illustrate how IGO can be used to design new optimization algorithms. As a proof of concept, we derive the IGO algorithm associated with restricted Boltzmann machines for discrete optimization, allowing for multimodal optimization. We perform a preliminary experimental study of the specific influence of the Fisher information matrix on the performance of the algorithm and on diversity of the optima obtained.

In Section 6, we discuss related work, and in particular, IGO's relationship with and differences from various other optimization algorithms such as natural evolution strategies or the cross-entropy method. We also sum up the main contributions of the paper and the design philosophy of IGO.

1 Algorithm description

We now present the outline of our algorithms. Each step is described in more detail in the sections below.

Our method can be seen as an *estimation of distribution algorithm*: at each time t, we maintain a probability distribution P_{θ^t} on the search space X, where $\theta^t \in \Theta$. The value of θ^t will evolve so that, over time, P_{θ^t} gives more weight to points x with better values of the function $f(x)$ to optimize.

A straightforward way to proceed is to transfer f from x-space to θ-space: define a function $F(\theta)$ as the P_θ-average of f and then to do a gradient descent for $F(\theta)$ in space Θ [Ber00]. This way, θ will converge to

a point such that P_θ yields a good average value of f. We depart from this approach in two ways:

- At each time, we replace f with an adaptive transformation of f representing how good or bad observed values of f are relative to other observations. This provides invariance under all monotone transformations of f.
- Instead of the vanilla gradient for θ, we use the so-called *natural gradient* given by the Fisher information matrix. This reflects the intrinsic geometry of the space of probability distributions, as introduced by Rao and Jeffreys [Rao45, Jef46] and later elaborated upon by Amari and others [AN00]. This provides invariance under reparametrization of θ and, importantly, minimizes the change of diversity of P_θ.

The algorithm is constructed in two steps: we first give an "ideal" version, namely, a version in which time t is continuous so that the evolution of θ^t is given by an ordinary differential equation in Θ. Second, the actual algorithm is a time discretization using a finite time step and Monte Carlo sampling instead of exact P_θ-averages.

1.1 The natural gradient on parameter space

About gradients and the shortest path uphill. Let g be a smooth function from $\mathbb{R}^d$ to $\mathbb{R}$, to be maximized. We first present the interpretation of gradient ascent as "the shortest path uphill".

Let $y \in \mathbb{R}^d$. Define the vector z by

$$z = \lim_{\varepsilon \to 0} \operatorname*{arg\,max}_{z,\, \|z\| \leqslant 1} g(y + \varepsilon z). \tag{1}$$

Then one can check that z is the normalized gradient of g at y: $z_i = \frac{\partial g/\partial y_i}{\|\partial g/\partial y_k\|}$. (This holds only at points y where the gradient of g does not vanish.)

This shows that, for small δt, the well-known gradient ascent of g given by

$$y_i^{t+\delta t} = y_i^t + \delta t \, \frac{\partial g}{\partial y_i}$$

realizes the largest increase in the value of g, for a given step size $\|y^{t+\delta t} - y^t\|$.

The relation (1) depends on the choice of a norm $\|\cdot\|$ (the gradient of g is given by $\partial g/\partial y_i$ only in an orthonormal basis). If we use, instead of the standard metric $\|y - y'\| = \sqrt{\sum (y_i - y_i')^2}$ on $\mathbb{R}^d$, a metric $\|y - y'\|_A = \sqrt{\sum A_{ij}(y_i - y_i')(y_j - y_j')}$ defined by a positive definite matrix A_{ij}, then the gradient of g with respect to this metric is given by $\sum_j A_{ij}^{-1} \frac{\partial g}{\partial y_i}$. (This follows from the textbook definition of gradients by $g(y + \varepsilon z) = g(y) + \varepsilon \langle \nabla g, z \rangle_A + O(\varepsilon^2)$ with $\langle \cdot, \cdot \rangle_A$ the scalar product associated with the matrix A_{ij} [Sch92].)

We can write the analogue of (1) using the A-norm. We get that the gradient ascent associated with metric A, given by

$$y^{t+\delta t} = y^t + \delta t \, A^{-1} \, \frac{\partial g}{\partial y_i},$$

for small δt, maximizes the increment of g for a given A-distance $\|y^{t+\delta t} - y^t\|_A$—it realizes the steepest A-ascent. Maybe this viewpoint clarifies the relationship between gradient and metric: this steepest ascent property can actually be used as a definition of gradients.

In our setting we want to use a gradient ascent in the parameter space Θ of our distributions P_θ. The metric $\|\theta - \theta'\| = \sqrt{\sum (\theta_i - \theta_i')^2}$ clearly depends on the choice of parametrization θ, and thus is not intrinsic. Therefore, we use a metric depending on θ only through the distributions P_θ, as follows.

Fisher information and the natural gradient on parameter space. Let $\theta, \theta' \in \Theta$ be two values of the distribution parameter. The Kullback–Leibler divergence between P_θ and $P_{\theta'}$ is defined [Kul97] as

$$\mathrm{KL}(P_{\theta'} \,\|\, P_\theta) = \int_x \ln \frac{P_{\theta'}(x)}{P_\theta(x)} \, P_{\theta'}(\mathrm{d}x).$$

When $\theta' = \theta + \delta\theta$ is close to θ, under mild smoothness assumptions we can expand the Kullback–Leibler divergence at second order in $\delta\theta$. This expansion defines the Fisher information matrix I at θ [Kul97]:

$$\mathrm{KL}(P_{\theta+\delta\theta} \,\|\, P_\theta) = \frac{1}{2} \sum I_{ij}(\theta) \, \delta\theta_i \delta\theta_j + O(\delta\theta^3).$$

An equivalent definition of the Fisher information matrix is by the usual formulas [CT06]

$$I_{ij}(\theta) = \int_x \frac{\partial \ln P_\theta(x)}{\partial \theta_i} \frac{\partial \ln P_\theta(x)}{\partial \theta_j} \, \mathrm{d}P_\theta(x) = -\int_x \frac{\partial^2 \ln P_\theta(x)}{\partial \theta_i \, \partial \theta_j} \, \mathrm{d}P_\theta(x).$$

The Fisher information matrix defines a (Riemannian) metric on Θ: the distance, in this metric, between two very close values of θ is given by the square root of twice the Kullback–Leibler divergence. Since the Kullback–Leibler divergence depends only on P_θ and not on the parametrization of θ, this metric is intrinsic.

If $g : \Theta \to \mathbb{R}$ is a smooth function on the parameter space, its *natural gradient* at θ is defined in accordance with the Fisher metric as

$$(\widetilde{\nabla}_\theta \, g)_i = \sum_j I^{-1}_{ij}(\theta) \, \frac{\partial g(\theta)}{\partial \theta_j}$$

or more synthetically

$$\widetilde{\nabla}_\theta \, g = I^{-1} \, \frac{\partial g}{\partial \theta}.$$

From now on, we will use $\widetilde{\nabla}_\theta$ to denote the natural gradient and $\frac{\partial}{\partial \theta}$ to denote the vanilla gradient.

By construction, the natural gradient descent is intrinsic: it does not depend on the chosen parametrization θ of P_θ, so that it makes sense to speak of the natural gradient ascent of a function $g(P_\theta)$.

Given that the Fisher metric comes from the Kullback–Leibler divergence, the "shortest path uphill" property of gradients mentioned above translates as follows (see also [Ama98, Theorem 1]):

Proposition 1. *The natural gradient ascent points in the direction $\delta\theta$ achieving the largest change of the objective function, for a given distance between P_θ and $P_{\theta+\delta\theta}$ in Kullback–Leibler divergence. More precisely, let g be a smooth function on the parameter space Θ. Let $\theta \in \Theta$ be a point where $\widetilde{\nabla} g(\theta)$ does not vanish. Then*

$$\frac{\widetilde{\nabla} g(\theta)}{\|\widetilde{\nabla} g(\theta)\|} = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \operatorname*{arg\,max}_{\substack{\delta\theta \text{ such that} \\ \mathrm{KL}(P_{\theta+\delta\theta} \,\|\, P_\theta) \leqslant \varepsilon^2/2}} g(\theta + \delta\theta).$$

Here we have implicitly assumed that the parameter space Θ is non-degenerate and proper (that is, no two points $\theta \in \Theta$ define the same probability distribution, and the mapping $P_\theta \mapsto \theta$ is continuous).

Why use the Fisher metric gradient for optimization? Relationship to diversity. The first reason for using the natural gradient is its reparametrization invariance, which makes it the only gradient available in a general abstract setting [AN00]. Practically, this invariance also limits the influence of encoding choices on the behavior of the algorithm. More prosaically, the Fisher matrix can be also seen as an *adaptive learning rate* for different components of the parameter vector θ_i: components i with a high impact on P_θ will be updated more cautiously.

Another advantage comes from the relationship with Kullback–Leibler distance in view of the "shortest path uphill" (see also [Ama98]). To minimize the value of some function $g(\theta)$ defined on the parameter space Θ, the naive approach follows a gradient descent for g using the "vanilla" gradient

$$\theta_i^{t+\delta t} = \theta_i^t + \delta t \frac{\partial g}{\partial \theta_i}$$

and, as explained above, this maximizes the increment of g for a given increment $\|\theta^{t+\delta t} - \theta^t\|$. On the other hand, the Fisher gradient

$$\theta_i^{t+\delta t} = \theta_i^t + \delta t I^{-1} \frac{\partial g}{\partial \theta_i}$$

maximizes the increment of g for a given Kullback–Leibler distance $\mathrm{KL}(P_{\theta^{t+\delta t}} \,\|\, P_{\theta^t})$.

In particular, if we choose an initial value θ^0 such that P_{θ^0} covers a wide portion of the space X uniformly, the Kullback–Leibler divergence between P_{θ^t} and P_{θ^0} measures the loss of diversity of P_{θ^t}. So the natural gradient descent is a way to optimize the function g with *minimal loss of diversity, provided the initial diversity is large*. On the other hand the vanilla gradient descent optimizes g with minimal change in the numerical values of the parameter θ, which is of little interest.

So arguably this method realizes the best trade-off between optimization and loss of diversity. (Though, as can be seen from the detailed algorithm description below, maximization of diversity occurs only greedily at each step, and so there is no guarantee that after a given time, IGO will provide the highest possible diversity for a given objective function value.)

An experimental confirmation of the positive influence of the Fisher matrix on diversity is given in Section 5 below. This may also provide a theoretical explanation to the good performance of CMA-ES.

1.2 IGO: Information-geometric optimization

Quantile rewriting of f. Our original problem is to minimize a function $f : X \to \mathbb{R}$. A simple way to turn f into a function on Θ is to use the expected value $-\mathbb{E}_{P_\theta} f$ [Ber00, WSPS08], but expected values can be unduly influenced by extreme values and using them is rather unstable [Whi89]; moreover $-\mathbb{E}_{P_\theta} f$ is not invariant under increasing transformation of f (this invariance implies we can only compare f-values, not add them).

Instead, we take an adaptive, quantile-based approach by first replacing the function f with a monotone rewriting W_θ^f and then following the gradient of $\mathbb{E}_{P_\theta} W_\theta^f$. A due choice of W_θ^f allows to control the range of the resulting values and achieves the desired invariance. Because the rewriting W_θ^f depends on θ, it might be viewed as an *adaptive* f-transformation.

The goal is that if $f(x)$ is "small" then $W_\theta^f(x) \in \mathbb{R}$ is "large" and vice versa, and that W_θ^f remains invariant under monotone transformations of f. The meaning of "small" or "large" depends on $\theta \in \Theta$ and is taken with respect to typical values of f under the current distribution P_θ. This is

measured by the P_θ-quantile in which the value of $f(x)$ lies. We write the lower and upper P_θ-f-quantiles of $x \in X$ as

$$\begin{aligned} q_\theta^-(x) &= \Pr_{x' \sim P_\theta}(f(x') < f(x)) \\ q_\theta^+(x) &= \Pr_{x' \sim P_\theta}(f(x') \leqslant f(x)) \ . \end{aligned} \tag{2}$$

These quantile functions reflect the probability to sample a better value than $f(x)$. They are monotone in f (if $f(x_1) \leqslant f(x_2)$ then $q_\theta^\pm(x_1) \leqslant q_\theta^\pm(x_2)$) and invariant under increasing transformations of f.

Given $q \in [0;1]$, we now choose a non-increasing function $w : [0;1] \to \mathbb{R}$ (fixed once and for all). A typical choice for w is $w(q) = \mathbb{1}_{q \leqslant q_0}$ for some fixed value q_0, the *selection quantile*. The transform $W_\theta^f(x)$ is defined as a function of the P_θ-f-quantile of x as

$$W_\theta^f(x) = \begin{cases} w(q_\theta^+(x)) & \text{if } q_\theta^+(x) = q_\theta^-(x), \\ \frac{1}{q_\theta^+(x) - q_\theta^-(x)} \int_{q=q_\theta^-(x)}^{q=q_\theta^+(x)} w(q) \, \mathrm{d}q & \text{otherwise.} \end{cases} \tag{3}$$

As desired, the definition of W_θ^f is invariant under an increasing transformation of f. For instance, the P_θ-median of f gets remapped to $w(\frac{1}{2})$.

Note that $\mathbb{E}_{P_\theta} W_\theta^f = \int_0^1 w$ is independent of f and θ: indeed, by definition, the quantile of a random point under P_θ is uniformly distributed in $[0;1]$. In the following, our objective will be to maximize the expected value of $W_{\theta^t}^f$ in θ, that is, to maximize

$$\mathbb{E}_{P_\theta} W_{\theta^t}^f = \int W_{\theta^t}^f(x) \, P_\theta(\mathrm{d}x) \tag{4}$$

over θ, where θ^t is fixed at a given step but will adapt over time.

Importantly, $W_\theta^f(x)$ can be estimated in practice: indeed, the quantiles $\Pr_{x' \sim P_\theta}(f(x') < f(x))$ can be estimated by taking samples of P_θ and ordering the samples according to the value of f (see below). The estimate remains invariant under increasing f-transformations.

The IGO gradient flow. At the most abstract level, IGO is a continuous-time gradient flow in the parameter space Θ, which we define now. In practice, discrete time steps (a.k.a. iterations) are used, and P_θ-integrals are approximated through sampling, as described in the next section.

Let θ^t be the current value of the parameter at time t, and let $\delta t \ll 1$. We define $\theta^{t+\delta t}$ in such a way as to increase the P_θ-weight of points where f is small, while not going too far from P_{θ^t} in Kullback–Leibler divergence. We use the adaptive weights $W_{\theta^t}^f$ as a way to measure which points have large or small values. In accordance with (4), this suggests taking the gradient ascent

$$\theta^{t+\delta t} = \theta^t + \delta t \, \widetilde{\nabla}_\theta \int W_{\theta^t}^f(x) \, P_\theta(\mathrm{d}x) \tag{5}$$

where the natural gradient is suggested by Proposition 1.

Note again that we use $W_{\theta^t}^f$ and not W_θ^f in the integral. So the gradient $\widetilde{\nabla}_\theta$ does not act on the adaptive objective $W_{\theta^t}^f$. If we used W_θ^f instead, we would face a paradox: right after a move, previously good points do not seem so good any more since the distribution has improved. More precisely, $\int W_\theta^f(x) \, P_\theta(\mathrm{d}x)$ is constant and always equal to the average weight $\int_0^1 w$, and so the gradient would always vanish.

Using the log-likelihood trick $\widetilde{\nabla} P_\theta = P_\theta \, \widetilde{\nabla} \ln P_\theta$ (assuming P_θ is smooth), we get an equivalent expression of the update above as an integral under the current distribution P_{θ^t}; this is important for practical implementation. This leads to the following definition.

Definition 2 (IGO flow)**.** *The IGO flow is the set of continuous-time trajectories in space* Θ*, defined by the differential equation*

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = \widetilde{\nabla}_\theta \int W^f_{\theta^t}(x)\, P_\theta(\mathrm{d}x) \tag{6}$$

$$= \int W^f_{\theta^t}(x)\, \widetilde{\nabla}_\theta \ln P_\theta(x)\, P_{\theta^t}(\mathrm{d}x) \tag{7}$$

$$= I^{-1}(\theta^t) \int W^f_{\theta^t}(x)\, \frac{\partial \ln P_\theta(x)}{\partial \theta}\, P_{\theta^t}(\mathrm{d}x). \tag{8}$$

where the gradients are taken at point $\theta = \theta^t$*, and* I *is the Fisher information matrix.*

Natural evolution strategies (NES, [WSPS08, GSS+10, SWSS09]) feature a related gradient *descent* with $f(x)$ instead of $W^f_{\theta^t}(x)$. The associated flow would read

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = -\widetilde{\nabla}_\theta \int f(x)\, P_\theta(\mathrm{d}x) \ , \tag{9}$$

where the gradient is taken at θ^t (in the sequel when not explicitly stated, gradients in θ are taken at $\theta = \theta^t$). However, in the end NESs always implement algorithms using sample quantiles, as if derived from the gradient ascent of $W^f_{\theta^t}(x)$.

The update (7) is a weighted average of "intrinsic moves" increasing the log-likelihood of some points. We can slightly rearrange the update as

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = \int \overbrace{W^f_{\theta^t}(x)}^{\text{preference weight}} \underbrace{\widetilde{\nabla}_\theta \ln P_\theta(x)}_{\text{intrinsic move to reinforce } x} \overbrace{P_{\theta^t}(\mathrm{d}x)}^{\text{current sample distribution}} \tag{10}$$

$$= \widetilde{\nabla}_\theta \int \underbrace{W^f_{\theta^t}(x) \ln P_\theta(x)}_{\text{weighted log-likelihood}}\, P_{\theta^t}(\mathrm{d}x). \tag{11}$$

which provides an interpretation for the IGO gradient flow as a gradient ascent optimization of the weighted log-likelihood of the "good points" of the current distribution. In a precise sense, IGO is in fact the "best" way to increase this log-likelihood (Theorem 13).

For exponential families of probability distributions, we will see later that the IGO flow rewrites as a nice derivative-free expression (18).

The IGO algorithm. Time discretization and sampling. The above is a mathematically well-defined continuous-time flow in the parameter space. Its practical implementation involves three approximations depending on two parameters N and δt:

- the integral under P_{θ^t} is approximated using N samples taken from P_{θ^t};
- the value $W^f_{\theta^t}$ is approximated for each sample taken from P_{θ^t};
- the time derivative $\frac{\mathrm{d}\theta^t}{\mathrm{d}t}$ is approximated by a δt time increment.

We also assume that the Fisher information matrix $I(\theta)$ and $\frac{\partial \ln P_\theta(x)}{\partial \theta}$ can be computed (see discussion below if $I(\theta)$ is unkown).

At each step, we pick N samples $x_1, \dots, x_N$ under P_{θ^t}. To approximate the quantiles, we rank the samples according to the value of f. Define $\mathrm{rk}(x_i) = \#\{j,\ f(x_j) < f(x_i)\}$ and let the estimated weight of sample x_i be

$$\widehat{w}_i = \frac{1}{N}\, w\left(\frac{\mathrm{rk}(x_i) + 1/2}{N}\right), \tag{12}$$

using the weighting scheme function w introduced above. (This is assuming there are no ties in our sample; in case several sample points have the same value of f, we define $\widehat{w}_i$ by averaging the above over all possible rankings of the ties[1].)

Then we can approximate the IGO flow as follows.

Definition 3 (IGO algorithm)**.** *The* IGO algorithm *with sample size N and step size δt is the following update rule for the parameter θ^t. At each step, N sample points $x_1, \dots, x_N$ are picked according to the distribution P_{θ^t}. The parameter is updated according to*

$$\theta^{t+\delta t} = \theta^t + \delta t \sum_{i=1}^{N} \widehat{w}_i \ \widetilde{\nabla}_\theta \ln P_\theta(x_i)\Big|_{\theta=\theta^t} \tag{13}$$

$$= \theta^t + \delta t \, I^{-1}(\theta^t) \sum_{i=1}^{N} \widehat{w}_i \ \frac{\partial \ln P_\theta(x_i)}{\partial \theta}\Big|_{\theta=\theta^t} \tag{14}$$

where $\widehat{w}_i$ is the weight (12) obtained from the ranked values of the objective function f.

Equivalently one can fix the weights $w_i = \frac{1}{N} w\left(\frac{i-1/2}{N}\right)$ once and for all and rewrite the update as

$$\theta^{t+\delta t} = \theta^t + \delta t \, I^{-1}(\theta^t) \sum_{i=1}^{N} w_i \ \frac{\partial \ln P_\theta(x_{i:N})}{\partial \theta}\Big|_{\theta=\theta^t} \tag{15}$$

where $x_{i:N}$ denotes the i^{th} sampled point ranked according to f, i.e. $f(x_{1:N}) < \dots < f(x_{N:N})$ (assuming again there are no ties). Note that $\{x_{i:N}\} = \{x_i\}$ and $\{w_i\} = \{\widehat{w}_i\}$.

As will be discussed in Section 4, this update applied to multivariate normal distributions or Bernoulli measures allows to neatly recover versions of some well-established algorithms, in particular CMA-ES and PBIL. Actually, in the Gaussian context updates of the form (14) have already been introduced [GSS+10, ANOK10], though not formally derived from a continuous-time flow with quantiles.

When $N \to \infty$, the IGO algorithm using samples approximates the continuous-time IGO gradient flow, see Theorem 4 below. Indeed, the IGO algorithm, with $N = \infty$, is simply the Euler approximation scheme for the ordinary differential equation defining the IGO flow (6). The latter result thus provides a sound mathematical basis for currently used rank-based updates.

IGO flow versus IGO algorithms. The IGO *flow* (6) is a well-defined continuous-time set of trajectories in the space of probability distributions P_θ, depending only on the objective function f and the chosen family of distributions. It does not depend on the chosen parametrization for θ (Proposition 8).

On the other hand, there are several IGO *algorithms* associated with this flow. Each IGO algorithm approximates the IGO flow in a slightly different

[1] A mathematically neater but less intuitive version would be

$$\widehat{w}_i = \frac{1}{\mathrm{rk}^+(x_i) - \mathrm{rk}^-(x_i)} \int_{u=\mathrm{rk}^-(x_i)/N}^{u=\mathrm{rk}^+(x_i)/N} w(u)\mathrm{d}u$$

with $\mathrm{rk}^-(x_i) = \#\{j,\ f(x_j) < f(x_i)\}$ and $\mathrm{rk}^+(x_i) = \#\{j,\ f(x_j) \leqslant f(x_i)\}$.

way. An IGO algorithm depends on three further choices: a sample size N, a time discretization step size δt, and a choice of parametrization for θ in which to implement (14).

If δt is small enough, and N large enough, the influence of the parametrization θ disappears and all IGO algorithms are approximations of the "ideal" IGO flow trajectory. However, the larger δt, the poorer the approximation gets.

So for large δt, different IGO algorithms for the same IGO flow may exhibit different behaviors. We will see an instance of this phenomenon for Gaussian distributions: both CMA-ES and the maximum likelihood update (EMNA) can be seen as IGO algorithms, but the latter with $\delta t = 1$ is known to exhibit premature loss of diversity (Section 4.2).

Still, two IGO algorithms for the same IGO flow will differ less from each other than from a non-IGO algorithm: at each step the difference is only $O(\delta t^2)$ (Section 2.4). On the other hand, for instance, the difference between an IGO algorithm and the vanilla gradient ascent is, generally, not smaller than $O(\delta t)$ at each step, i.e. roughly as big as the steps themselves.

Unknown Fisher matrix. The algorithm presented so far assumes that the Fisher matrix $I(\theta)$ is known as a function of θ. This is the case for Gaussian distributions in CMA-ES and for Bernoulli distributions. However, for restricted Boltzmann machines as considered below, no analytical form is known. Yet, provided the quantity $\frac{\partial}{\partial\theta} \ln P_\theta(x)$ can be computed or approximated, it is possible to approximate the integral

$$I_{ij}(\theta) = \int_x \frac{\partial \ln P_\theta(x)}{\partial \theta_i} \frac{\partial \ln P_\theta(x)}{\partial \theta_j} P_\theta(\mathrm{d}x)$$

using P_θ-Monte Carlo samples for x. These samples may or may not be the same as those used in the IGO update (14): in particular, it is possible to use as many Monte Carlo samples as necessary to approximate I_{ij}, at no additional cost in terms of the number of calls to the black-box function f to optimize.

Note that each Monte Carlo sample x will contribute $\frac{\partial \ln P_\theta(x)}{\partial \theta_i} \frac{\partial \ln P_\theta(x)}{\partial \theta_j}$ to the Fisher matrix approximation. This is a rank-1 matrix. So, for the approximated Fisher matrix to be invertible, the number of (distinct) samples x needs to be at least equal to the number of components of the parameter θ i.e. $N_{\text{Fisher}} \geqslant \dim \Theta$.

For exponential families of distributions, the IGO update has a particular form (18) which simplifies this matter somewhat. More details are given below (see Section 5) for the concrete situation of restricted Boltzmann machines.

2 First properties of IGO

2.1 Consistency of sampling

The first property to check is that when $N \to \infty$, the update rule using N samples converges to the IGO update rule. This is *not* a straightforward application of the law of large numbers, because the estimated weights $\widehat{w}_i$ depend (non-continuously) on the whole sample $x_1, \ldots, x_N$, and not only on x_i.

Theorem 4 (Consistency). *When $N \to \infty$, the N-sample IGO update rule (14):*

$$\theta^{t+\delta t} = \theta^t + \delta t\, I^{-1}(\theta^t) \sum_{i=1}^{N} \widehat{w}_i \left.\frac{\partial \ln P_\theta(x_i)}{\partial \theta}\right|_{\theta=\theta^t}$$

converges with probability 1 *to the update rule (5):*

$$\theta^{t+\delta t} = \theta^t + \delta t\, \widetilde{\nabla}_\theta \int W^f_{\theta^t}(x)\, P_\theta(\mathrm{d}x).$$

The proof is given in the Appendix, under mild regularity assumptions. In particular we do not require that w be continuous.

This theorem may clarify previous claims [WSPS08, ANOK10] where rank-based updates similar to (5), such as in NES or CMA-ES, were derived from optimizing the expected value $-\mathbb{E}_{P_\theta} f$. The rank-based weights $\widehat{w}_i$ were then introduced somewhat arbitrarily. Theorem 4 shows that, for large N, CMA-ES and NES actually follow the gradient flow of the quantity $\mathbb{E}_{P_\theta} W^f_{\theta^t}$: the update can be rigorously derived from optimizing the expected value of the quantile-rewriting $W^f_{\theta^t}$.

2.2 Monotonicity: quantile improvement

Gradient descents come with a guarantee that the fitness value decreases over time. Here, since we work with probability distributions on X, we need to define the fitness of the distribution P_{θ^t}. An obvious choice is the expectation $\mathbb{E}_{P_{\theta^t}} f$, but it is not invariant under f-transformation and moreover may be sensitive to extreme values.

It turns out that the monotonicity properties of the IGO gradient flow depend on the choice of the weighting scheme w. For instance, if $w(u) = \mathbb{1}_{u \leqslant 1/2}$, then the median of f improves over time.

Proposition 5 (Quantile improvement). *Consider the IGO flow given by (6), with the weight $w(u) = \mathbb{1}_{u \leqslant q}$ where $0 < q < 1$ is fixed. Then the value of the q-quantile of f improves over time: if $t_1 \leqslant t_2$ then either $\theta^{t_1} = \theta^{t_2}$ or $Q^q_{P_{\theta^{t_2}}}(f) < Q^q_{P_{\theta^{t_1}}}(f)$.*

Here the q-quantile $Q^q_P(f)$ of f under a probability distribution P is defined as any number m such that $\Pr_{x\sim P}(f(x) \leqslant m) \geqslant q$ and $\Pr_{x\sim P}(f(x) \geqslant m) \geqslant 1-q$.

The proof is given in the Appendix, together with the necessary regularity assumptions.

Of course this holds only for the IGO gradient flow (6) with $N = \infty$ and $\delta t \to 0$. For an IGO algorithm with finite N, the dynamics is random and one cannot expect monotonicity. Still, Theorem 4 ensures that, with high probability, trajectories of a large enough finite population dynamics stay close to the infinite-population limit trajectory.

2.3 The IGO flow for exponential families

The expressions for the IGO update simplify somewhat if the family P_θ happens to be an exponential family of probability distributions (see also [MMS08]). Suppose that P_θ can be written as

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp\Big(\sum \theta_i T_i(x)\Big)\, H(\mathrm{d}x)$$

where $T_1, \dots, T_k$ is a finite family of functions on X, $H(\mathrm{d}x)$ is an arbitrary reference measure on X, and $Z(\theta)$ is the normalization constant. It is well-known [AN00, (2.33)] that

$$\frac{\partial \ln P_\theta(x)}{\partial \theta_i} = T_i(x) - \mathbb{E}_{P_\theta} T_i \tag{16}$$

so that [AN00, (3.59)]

$$I_{ij}(\theta) = \mathrm{Cov}_{P_\theta}(T_i, T_j). \tag{17}$$

In the end we find:

Proposition 6. *Let P_θ be an exponential family parametrized by the natural parameters θ as above. Then the IGO flow is given by*

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \mathrm{Cov}_{P_\theta}(T, T)^{-1} \, \mathrm{Cov}_{P_\theta}(T, W_\theta^f) \tag{18}$$

where $\mathrm{Cov}_{P_\theta}(T, W_\theta^f)$ denotes the vector $(\mathrm{Cov}_{P_\theta}(T_i, W_\theta^f))_i$, and $\mathrm{Cov}_{P_\theta}(T, T)$ the matrix $(\mathrm{Cov}_{P_\theta}(T_i, T_j))_{ij}$.

Note that the right-hand side does not involve derivatives w.r.t. θ any more. This result makes it easy to simulate the IGO flow using e.g. a Gibbs sampler for P_θ: both covariances in (18) may be approximated by sampling, so that neither the Fisher matrix nor the gradient term need to be known in advance, and no derivatives are involved.

The CMA-ES uses the family of all Gaussian distributions on $\mathbb{R}^d$. Then, the family T_i is the family of all linear and quadratic functions of the coordinates on $\mathbb{R}^d$. The expression above is then a particularly concise rewriting of a CMA-ES update, see also Section 4.2.

Moreover, the expected values $\bar{T}_i = \mathbb{E}T_i$ of T_i satisfy the simple evolution equation under the IGO flow

$$\frac{\mathrm{d}\bar{T}_i}{\mathrm{d}t} = \mathrm{Cov}(T_i, W_\theta^f) = \mathbb{E}(T_i \, W_\theta^f) - \bar{T}_i \, \mathbb{E}W_\theta^f. \tag{19}$$

The proof is given in the Appendix, in the proof of Theorem 15.

The variables $\bar{T}_i$ can sometimes be used as an alternative parametrization for an exponential family (e.g. for a one-dimensional Gaussian, these are the mean μ and the second moment $\mu^2 + \sigma^2$). Then the IGO flow (7) may be rewritten using the relation $\widetilde{\nabla}_{\theta_i} = \frac{\partial}{\partial \bar{T}_i}$ for the natural gradient of exponential families (Appendix, Proposition 22), which sometimes results in simpler expressions. We shall further exploit this fact in Section 3.

Exponential families with latent variables. Similar formulas hold when the distribution $P_\theta(x)$ is the marginal of an exponential distribution $P_\theta(x, h)$ over a "hidden" or "latent" variable h, such as the restricted Boltzmann machines of Section 5.

Namely, with $P_\theta(x) = \frac{1}{Z(\theta)} \sum_h \exp(\sum_i \theta_i T_i(x, h)) \, H(\mathrm{d}x, \mathrm{d}h)$ the Fisher matrix is

$$I_{ij}(\theta) = \mathrm{Cov}_{P_\theta}(U_i, U_j) \tag{20}$$

where $U_i(x) = \mathbb{E}_{P_\theta}(T_i(x, h)|x)$. Consequently, the IGO flow takes the form

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \mathrm{Cov}_{P_\theta}(U, U)^{-1} \, \mathrm{Cov}_{P_\theta}(U, W_\theta^f). \tag{21}$$

2.4 Invariance properties

Here we formally state the invariance properties of the IGO flow under various reparametrizations. Since these results follow from the very construction of the algorithm, the proofs are omitted.

Proposition 7 (f-invariance)**.** *Let $\varphi : \mathbb{R} \to \mathbb{R}$ be an increasing function. Then the trajectories of the IGO flow when optimizing the functions f and $\varphi(f)$ are the same.*

The same is true for the discretized algorithm with population size N and step size $\delta t > 0$.

Proposition 8 (θ-invariance)**.** *Let $\theta' = \varphi(\theta)$ be a one-to-one function of θ and let $P'_{\theta'} = P_{\varphi^{-1}(\theta)}$. Let θ^t be the trajectory of the IGO flow when optimizing a function f using the distributions P_θ, initialized at θ^0. Then the IGO flow trajectory $(\theta')^t$ obtained from the optimization of the function f using the distributions $P'_{\theta'}$, initialized at $(\theta')^0 = \varphi(\theta^0)$, is the same, namely $(\theta')^t = \varphi(\theta^t)$.*

For the algorithm with finite N and $\delta t > 0$, invariance under θ-reparametrization is only true approximately, in the limit when $\delta t \to 0$. As mentioned above, the IGO update (14), with $N = \infty$, is simply the Euler approximation scheme for the ordinary differential equation (6) defining the IGO flow. At each step, the Euler scheme is known to make an error $O(\delta t^2)$ with respect to the true flow. This error actually depends on the parametrization of θ.

So the IGO updates for different parametrizations coincide at first order in δt, and may, in general, differ by $O(\delta t^2)$. For instance the difference between the CMA-ES and xNES updates is indeed $O(\delta t^2)$, see Section 4.2.

For comparison, using the vanilla gradient results in a divergence of $O(\delta t)$ at each step between different parametrizations. So the divergence could be of the same magnitude as the steps themselves.

In that sense, one can say that IGO algorithms are "more parametrization-invariant" than other algorithms. This stems from their origin as a discretization of the IGO flow.

The next proposition states that, for example, if one uses a family of distributions on $\mathbb{R}^d$ which is invariant under affine transformations, then our algorithm optimizes equally well a function and its image under any affine transformation (up to an obvious change in the initialization). This proposition generalizes the well-known corresponding property of CMA-ES [HO01].

Here, as usual, the image of a probability distribution P by a transformation φ is defined as the probability distribution P' such that $P'(Y) = P(\varphi^{-1}(Y))$ for any subset $Y \subset X$. In the continuous domain, the density of the new distribution P' is obtained by the usual change of variable formula involving the Jacobian of φ.

Proposition 9 (X-invariance)**.** *Let $\varphi : X \to X$ be a one-to-one transformation of the search space, and assume that φ globally preserves the family of measures P_θ. Let θ^t be the IGO flow trajectory for the optimization of function f, initialized at P_{θ^0}. Let $(\theta')^t$ be the IGO flow trajectory for optimization of $f \circ \varphi^{-1}$, initialized at the image of P_{θ^0} by φ. Then $P_{(\theta')^t}$ is the image of P_{θ^t} by φ.*

For the discretized algorithm with population size N and step size $\delta t > 0$, the same is true up to an error of $O(\delta t^2)$ per iteration. This error disappears if the map φ acts on Θ in an affine way.

The latter case of affine transforms is well exemplified by CMA-ES: here, using the variance and mean as the parametrization of Gaussians, the new mean and variance after an affine transform of the search space are an affine function of the old mean and variance; specifically, for the affine transformation $A : x \mapsto Ax + b$ we have $(m, C) \mapsto (Am + b, ACA^{\mathrm{T}})$.

2.5 Speed of the IGO flow

Proposition 10. *The speed of the IGO flow, i.e. the norm of* $\frac{\mathrm{d}\theta^t}{\mathrm{d}t}$ *in the Fisher metric, is at most* $\sqrt{\int_0^1 w^2 - (\int_0^1 w)^2}$ *where* w *is the weighting scheme.*

This speed can be tested in practice in at least two ways. The first is just to compute the Fisher norm of the increment $\theta^{t+\delta t} - \theta^t$ using the Fisher matrix; for small δt this is close to $\delta t \|\frac{\mathrm{d}\theta}{\mathrm{d}t}\|$ with $\|\cdot\|$ the Fisher metric. The second is as follows: since the Fisher metric coincides with the Kullback–Leibler divergence up to a factor $1/2$, we have $\mathrm{KL}(P_{\theta^{t+\delta t}} \,\|\, P_{\theta^t}) \approx \frac{1}{2}\delta t^2 \|\frac{\mathrm{d}\theta}{\mathrm{d}t}\|^2$ at least for small δt. Since it is relatively easy to estimate $\mathrm{KL}(P_{\theta^{t+\delta t}} \,\|\, P_{\theta^t})$ by comparing the new and old log-likelihoods of points in a Monte Carlo sample, one can obtain an estimate of $\|\frac{\mathrm{d}\theta}{\mathrm{d}t}\|$.

Corollary 11. *Consider an IGO algorithm with weighting scheme* w*, step size* δt *and sample size* N*. Then, for small* δt *and large* N *we have*

$$\mathrm{KL}(P_{\theta^{t+\delta t}} \,\|\, P_{\theta^t}) \leqslant \frac{1}{2}\delta t^2 \operatorname{Var}_{[0,1]} w + O(\delta t^3) + O(1/\sqrt{N}).$$

For instance, with $w(q) = \mathbb{1}_{q \leqslant q_0}$ and neglecting the error terms, an IGO algorithm introduces at most $\frac{1}{2}\delta t^2 \, q_0(1 - q_0)$ bits of information (in base e) per iteration into the probability distribution P_θ.

Thus, the time discretization parameter δt is not just an arbitrary variable: it has an intrinsic interpretation related to a number of bits introduced at each step of the algorithm. This kind of relationship suggests, more generally, to use the Kullback–Leibler divergence as an external and objective way to measure learning rates in those optimization algorithms which use probability distributions.

The result above is only an upper bound. Maximal speed can be achieved only if all "good" points point in the same direction. If the various good points in the sample suggest moves in inconsistent directions, then the IGO update will be much smaller. The latter may be a sign that the signal is noisy, or that the family of distributions P_θ is not well suited to the problem at hand and should be enriched.

As an example, using a family of Gaussian distributions with unkown mean and fixed identity variance on $\mathbb{R}^d$, one checks that for the optimization of a linear function on $\mathbb{R}^d$, with the weight $w(u) = -\mathbb{1}_{u>1/2} + \mathbb{1}_{u<1/2}$, the IGO flow moves at constant speed $1/\sqrt{2\pi} \approx 0.4$, whatever the dimension d. On a rapidly varying sinusoidal function, the moving speed will be much slower because there are "good" and "bad" points in all directions.

This may suggest ways to design the weighting scheme w to achieve maximal speed in some instances. Indeed, looking at the proof of the proposition, which involves a Cauchy–Schwarz inequality, one can see that the maximal speed is achieved only if there is a linear relationship between the weights $W_\theta^f(x)$ and the gradient $\nabla_\theta \ln P_\theta(x)$. For instance, for the optimization of a linear function on $\mathbb{R}^d$ using Gaussian measures of known variance, the maximal speed will be achieved when the weighting scheme $w(u)$ is the inverse of the Gaussian cumulative distribution function. (In particular, $w(u)$ tends

to $+\infty$ when $u \to 0$ and to $-\infty$ when $u \to 1$.) This is in accordance with previously known results: the expected value of the i-th order statistic of N standard Gaussian variates is the optimal $\widehat{w}_i$ value in evolution strategies [Bey01, Arn06]. For $N \to \infty$, this order statistic converges to the inverse Gaussian cumulative distribution function.

2.6 Noisy objective function

Suppose that the objective function f is non-deterministic: each time we ask for the value of f at a point $x \in X$, we get a random result. In this setting we may write the random value $f(x)$ as $f(x) = \tilde{f}(x, \omega)$ where ω is an unseen random parameter, and $\tilde{f}$ is a deterministic function of x and ω. Without loss of generality, up to a change of variables we can assume that ω is uniformly distributed in $[0, 1]$.

We can still use the IGO algorithm without modification in this context. One might wonder which properties (consistency of sampling, etc.) still apply when f is not deterministic. Actually, IGO algorithms for noisy functions fit very nicely into the IGO framework: the following proposition allows to transfer any property of IGO to the case of noisy functions.

Proposition 12 (Noisy IGO)**.** *Let f be a random function of $x \in X$, namely, $f(x) = \tilde{f}(x, \omega)$ where ω is a random variable uniformly distributed in $[0, 1]$, and $\tilde{f}$ is a deterministic function of x and ω. Then the two following algorithms coincide:*

- *The IGO algorithm (13), using a family of distributions P_θ on space X, applied to the noisy function f, and where the samples are ranked according to the random observed value of f (here we assume that, for each sample, the noise ω is independent from everything else);*
- *The IGO algorithm on space $X \times [0, 1]$, using the family of distributions $\tilde{P}_\theta = P_\theta \otimes U_{[0,1]}$, applied to the deterministic function $\tilde{f}$. Here $U_{[0,1]}$ denotes the uniform law on $[0, 1]$.*

The (easy) proof is given in the Appendix.

This proposition states that noisy optimization is the same as ordinary optimization using a family of distributions which cannot operate any selection or convergence over the parameter ω. More generally, any component of the search space in which a distribution-based evolutionary strategy cannot perform selection or specialization will effectively act as a random noise on the objective function.

As a consequence of this result, all properties of IGO can be transferred to the noisy case. Consider, for instance, consistency of sampling (Theorem 4). The N-sample IGO update rule for the noisy case is identical to the non-noisy case (14):

$$\theta^{t+\delta t} = \theta^t + \delta t \, I^{-1}(\theta^t) \sum_{i=1}^{N} \widehat{w}_i \left. \frac{\partial \ln P_\theta(x_i)}{\partial \theta} \right|_{\theta=\theta^t}$$

where each weight $\widehat{w}_i$ computed from (12) now incorporates noise from the objective function because the rank of x_i is computed on the random function, or equivalently on the deterministic function $\tilde{f}$: $\mathrm{rk}(x_i) = \#\{j, \tilde{f}(x_j, \omega_j) < \tilde{f}(x_i, \omega_i)\}$.

Consistency of sampling (Theorem 4) thus takes the following form: When $N \to \infty$, the N-sample IGO update rule on the noisy function f

converges with probability 1 to the update rule

$$\begin{aligned}\theta^{t+\delta t} &= \theta^t + \delta t\, \widetilde{\nabla}_\theta \int_0^1 \int W^{\tilde{f}}_{\theta^t}(x,\omega)\, P_\theta(\mathrm{d}x)\, \mathrm{d}\omega. \\ &= \theta^t + \delta t\, \widetilde{\nabla}_\theta \int \bar{W}^f_{\theta^t}(x)\, P_\theta(\mathrm{d}x) \end{aligned} \tag{22}$$

where $\bar{W}^f_\theta(x) = \mathbb{E}_\omega W^{\tilde{f}}_\theta(x,\omega)$. This entails, in particular, that when $N \to \infty$, the noise disappears asymptotically, as could be expected.

Consequently, the IGO flow in the noisy case should be defined by the $\delta t \to 0$ limit of the update (22) using $\bar{W}$. Note that the quantiles $q^\pm(x)$ defined by (2) still make sense in the noisy case, and are deterministic functions of x; thus $W^f_\theta(x)$ can also be defined by (3) and is deterministic. However, unless the weighting scheme $w(q)$ is affine, $\bar{W}^f_\theta(x)$ is different from $W^f_\theta(x)$ in general. Thus, unless w is affine the flows defined by W and $\bar{W}$ do not coincide in the noisy case. The flow using W would be the $N \to \infty$ limit of a slightly more complex algorithm using several evaluations of f for each sample x_i in order to compute noise-free ranks.

2.7 Implementation remarks

Influence of the weighting scheme w. The weighting scheme w directly affects the update rule (15).

A natural choice is $w(u) = \mathbb{1}_{u \leqslant q}$. This, as we have proved, results in an improvement of the q-quantile over the course of optimization. Taking $q = 1/2$ springs to mind; however, this is not selective enough, and both theory and experiments confirm that for the Gaussian case (CMA-ES), most efficient optimization requires $q < 1/2$ (see Section 4.2). The optimal q is about 0.27 if N is not larger than the search space dimension d [Bey01] and even smaller otherwise.

Second, replacing w with $w+c$ for some constant c clearly has no influence on the IGO continuous-time flow (5), since the gradient will cancel out the constant. However, this is not the case for the update rule (15) with a finite sample of size N.

Indeed, adding a constant c to w adds a quantity $c\frac{1}{N}\sum \widetilde{\nabla}_\theta \ln P_\theta(x_i)$ to the update. Since we know that the P_θ-expected value of $\widetilde{\nabla}_\theta \ln P_\theta$ is 0 (because $\int (\widetilde{\nabla}_\theta \ln P_\theta)\, P_\theta = \int \widetilde{\nabla} P_\theta = \widetilde{\nabla} 1 = 0$), we have $\mathbb{E}\frac{1}{N}\widetilde{\nabla}_\theta \ln P_\theta(x_i) = 0$. So adding a constant to w does not change the expected value of the update, but it may change e.g. its variance. The empirical average of $\widetilde{\nabla}_\theta \ln P_\theta(x_i)$ in the sample will be $O(1/\sqrt{N})$. So translating the weights results in a $O(1/\sqrt{N})$ change in the update. See also Section 4 in [SWSS09].

Determining an optimal value for c to reduce the variance of the update is difficult, though: the optimal value actually depends on possible correlations between $\widetilde{\nabla}_\theta \ln P_\theta$ and the function f. The only general result is that one should shift w so that 0 lies within its range. Assuming independence, or dependence with enough symmetry, the optimal shift is when the weights average to 0.

Adaptive learning rate. Comparing consecutive updates to evaluate a learning rate or step size is an effective measure. For example, in back-propagation, the update sign has been used to adapt the learning rate of each single weight in an artificial neural network [SA90]. In CMA-ES, a step size is adapted depending on whether recent steps tended to move in a consistent direction or to backtrack. This is measured by considering

the changes of the mean m of the Gaussian distribution. For a probability distribution P_θ on an arbitrary search space, in general no notion of mean may be defined. However, it is still possible to define "backtracking" in the evolution of θ as follows.

Consider two successive updates $\delta\theta^t = \theta^t - \theta^{t-\delta t}$ and $\delta\theta^{t+\delta t} = \theta^{t+\delta t} - \theta^t$. Their scalar product in the Fisher metric $I(\theta^t)$ is

$$\langle \delta\theta^t, \delta\theta^{t+\delta t} \rangle = \sum_{ij} I_{ij}(\theta^t)\, \delta\theta^t_i\, \delta\theta^{t+\delta t}_j .$$

Dividing by the associated norms will yield the cosine $\cos\alpha$ of the angle between $\delta\theta^t$ and $\delta\theta^{t+\delta t}$.

If this cosine is positive, the learning rate δt may be increased. If the cosine is negative, the learning rate probably needs to be decreased. Various schemes for the change of δt can be devised; for instance, inspired by CMA-ES, one can multiply δt by $\exp(\beta(\cos\alpha)/2)$ or $\exp(\beta(\mathbb{1}_{\cos\alpha>0} - \mathbb{1}_{\cos\alpha<0})/2)$, where $\beta \approx \min(N/\dim\Theta, 1/2)$.

As before, this scheme is constructed to be robust w.r.t. reparametrization of θ, thanks to the use of the Fisher metric. However, for large learning rates δt, in practice the parametrization might well become relevant.

A consistent direction of the updates does not necessarily mean that the algorithm is performing well: for instance, when CEM/EMNA exhibits premature convergence (see below), the parameters consistently move towards a zero covariance matrix and the cosines above are positive.

Complexity. The complexity of the IGO algorithm depends much on the computational cost model. In optimization, it is fairly common to assume that the objective function f is very costly compared to any other calculations performed by the algorithm. Then the cost of IGO in terms of number of f-calls is N per iteration, and the cost of using quantiles and computing the natural gradient is negligible.

Setting the cost of f aside, the complexity of the IGO algorithm depends mainly on the computation of the (inverse) Fisher matrix. Assume an analytical expression for this matrix is known. Then, with $p = \dim\Theta$ the number of parameters, the cost of storage of the Fisher matrix is $O(p^2)$ per iteration, and its inversion typically costs $O(p^3)$ per iteration. However, depending on the situation and on possible algebraic simplifications, strategies exist to reduce this cost (e.g. [LRMB07] in a learning context). For instance, for CMA-ES the cost is $O(Np)$ [SHI09]. More generally, parametrization by expectation parameters (see above), when available, may reduce the cost to $O(p)$ as well.

If no analytical form of the Fisher matrix is known and Monte Carlo estimation is required, then complexity depends on the particular situation at hand and is related to the best sampling strategies available for a particular family of distributions. For Boltzmann machines, for instance, a host of such strategies are available. Still, in such a situation, IGO can be competitive if the objective function f is costly.

Recycling samples. We might use samples not only from the last iteration to compute the ranks in (12), see e.g. [SWSS09]. For $N = 1$ this is indispensable. In order to preserve sampling consistency (Theorem 4) the old samples need to be reweighted (using the ratio of their new vs old likelihood, as in importance sampling).

Initialization. As with other optimization algorithms, it is probably a good idea to initialize in such a way as to cover a wide portion of the search space, i.e. θ^0 should be chosen so that P_{θ^0} has maximal diversity. For IGO algorithms this is particularly relevant, since, as explained above, the natural gradient provides minimal change of diversity (greedily at each step) for a given change in the objective function.

3 IGO, maximum likelihood and the cross-entropy method

IGO as a smooth-time maximum likelihood estimate. The IGO flow turns out to be the only way to maximize a *weighted* log-likelihood, where points of the current distribution are slightly reweighted according to f-preferences.

This relies on the following interpretation of the natural gradient as a weighted maximum likelihood update with infinitesimal learning rate. This result singles out, in yet another way, the *natural* gradient among all possible gradients. The proof is given in the Appendix.

Theorem 13 (Natural gradient as ML with infinitesimal weights)**.** *Let $\varepsilon > 0$ and $\theta_0 \in \Theta$. Let $W(x)$ be a function of x and let θ be the solution of*

$$\theta = \arg\max_\theta \left\{ (1-\varepsilon) \underbrace{\int \log P_\theta(x)\, P_{\theta_0}(\mathrm{d}x)}_{\text{maximal for } \theta=\theta_0} + \varepsilon \int \log P_\theta(x)\, W(x)\, P_{\theta_0}(\mathrm{d}x) \right\}.$$

Then, when $\varepsilon \to 0$, up to $O(\varepsilon^2)$ we have

$$\theta = \theta_0 + \varepsilon \int \widetilde{\nabla}_\theta \ln P_\theta(x)\ W(x)\, P_{\theta_0}(\mathrm{d}x).$$

Likewise for discrete samples: with $x_1, \dots, x_N \in X$, let θ be the solution of

$$\theta = \arg\max \left\{ (1-\varepsilon) \int \log P_\theta(x)\, P_{\theta_0}(\mathrm{d}x) + \varepsilon \sum_i W(x_i)\ \log P_\theta(x_i) \right\}.$$

Then when $\varepsilon \to 0$, up to $O(\varepsilon^2)$ we have

$$\theta = \theta_0 + \varepsilon \sum_i W(x_i)\ \widetilde{\nabla}_\theta \ln P_\theta(x_i).$$

So if $W(x) = W^f_{\theta_0}(x)$ is the weight of the points according to quantilized f-preferences, the weighted maximum log-likelihood necessarily is the IGO flow (7) using the natural gradient—or the IGO update (14) when using samples.

Thus the IGO flow is the unique flow that, continuously in time, slightly changes the distribution to maximize the log-likelihood of points with good values of f. Moreover IGO continuously updates the weight $W^f_{\theta_0}(x)$ depending on f and on the current distribution, so that we keep optimizing.

This theorem suggests a way to approximate the IGO flow by enforcing this interpretation for a given non-infinitesimal step size δt, as follows.

Definition 14 (IGO-ML algorithm)**.** *The* IGO-ML algorithm *with step size δt updates the value of the parameter θ^t according to*

$$\theta^{t+\delta t} = \arg\max_\theta \left\{ (1 - \delta t \sum_i \widehat{w}_i) \int \log P_\theta(x)\, P_{\theta^t}(\mathrm{d}x) \ +\ \delta t \sum_i \widehat{w}_i \log P_\theta(x_i) \right\} \tag{23}$$

where $x_1, \dots, x_N$ are sample points picked according to the distribution P_{θ^t}, and $\widehat{w}_i$ is the weight (12) obtained from the ranked values of the objective function f.

As for the cross-entropy method below, this only makes algorithmic sense if the argmax is tractable.

It turns out that IGO-ML is just the IGO algorithm in a particular parametrization (see Theorem 15).

The cross-entropy method. Taking $\delta t = 1$ in (23) above corresponds to a full maximum likelihood update, which is also related to the *cross-entropy method* (CEM). The cross-entropy method can be defined as follows [dBKMR05] in an optimization setting. Like IGO, it depends on a family of probability distributions P_θ parametrized by $\theta \in \Theta$, and a number of samples N at each iteration. Let also $N_e = \lceil qN \rceil$ $(0 < q < 1)$ be a number of *elite* samples.

At each step, the cross-entropy method for optimization samples N points $x_1, \dots, x_N$ from the current distribution P_{θ^t}. Let $\widehat{w}_i$ be $1/N_e$ if x_i belongs to the N_e samples with the best value of the objective function f, and $\widehat{w}_i = 0$ otherwise. Then the *cross-entropy method* or *maximum likelihoood* update (CEM/ML) for optimization is

$$\theta^{t+1} = \arg\max_\theta \sum \widehat{w}_i \log P_\theta(x_i) \tag{24}$$

(assuming the argmax is tractable).

A version with a smoother update depends on a step size parameter $0 < \alpha \leqslant 1$ and is given [dBKMR05] by

$$\theta^{t+1} = (1-\alpha)\theta^t + \alpha \arg\max_\theta \sum \widehat{w}_i \log P_\theta(x_i). \tag{25}$$

The standard CEM/ML update corresponds to $\alpha = 1$.

For $\alpha = 1$ the standard cross-entropy method is independent of the parametrization θ, whereas for $\alpha < 1$ this is not the case.

Note the difference between the IGO-ML algorithm (23) and the smoothed CEM update (25) with step size $\alpha = \delta t$: the smoothed CEM update performs a weighted average of the parameter value *after* taking the maximum likelihood estimate, whereas IGO-ML uses a weighted average of current and previous likelihoods, *then* takes a maximum likelihood estimate. In general, these two rules can greatly differ, as they do for Gaussian distributions (Section 4.2).

This interversion of averaging makes IGO-ML parametrization-independent whereas the smoothed CEM update is not.

Yet, for exponential families of probability distributions, there exists one particular parametrization θ in which the IGO algorithm and the smoothed CEM update coincide. We now proceed to this construction.

IGO for expectation parameters and maximum likelihood. The particular form of IGO for exponential families has an interesting consequence if the parametrization chosen for the exponential family is the set of *expectation parameters*. Let $P_\theta(x) = \frac{1}{Z(\theta)} \exp\left(\sum \theta_j T_j(x)\right) H(\mathrm{d}x)$ be an exponential family as above. The *expectation parameters* are $\bar{T}_j = \bar{T}_j(\theta) = \mathbb{E}_{P_\theta} T_j$, (denoted η_j in [AN00, (3.56)]). The notation $\bar{T}$ will denote the collection $(\bar{T}_j)$.

It is well-known that, in this parametrization, the maximum likelihood estimate for a sample of points $x_1, \dots, x_k$ is just the empirical average of the expectation parameters over that sample:

$$\arg\max_{\bar{T}} \frac{1}{k} \sum_{i=1}^{k} \log P_{\bar{T}}(x_i) = \frac{1}{k} \sum_{i=1}^{k} T(x_i). \tag{26}$$

In the discussion above, one main difference between IGO and smoothed CEM was whether we took averages before or after taking the maximum log-likelihood estimate. For the expectation parameters $\bar{T}_i$, we see that these operations commute. (One can say that these expectation parameters "linearize maximum likelihood estimates".) A little work brings us to the

Theorem 15 (IGO, CEM and maximum likelihood)**.** *Let*

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp\left(\sum \theta_j T_j(x)\right) \, H(\mathrm{d}x)$$

be an exponential family of probability distributions, where the T_j are functions of x and H is some reference measure. Let us parametrize this family by the expected values $\bar{T}_j = \mathbb{E} T_j$.

Let us assume the chosen weights $\widehat{w}_i$ sum to 1. For a sample $x_1, \dots, x_N$, let

$$T_j^* = \sum_i \widehat{w}_i \, T_j(x_i).$$

Then the IGO update (14) in this parametrization reads

$$\bar{T}_j^{t+\delta t} = (1 - \delta t) \, \bar{T}_j^t + \delta t \, T_j^*. \tag{27}$$

Moreover these three algorithms coincide:

- *The IGO-ML algorithm (23).*
- *The IGO algorithm written in the parametrization $\bar{T}_j$.*
- *The smoothed CEM algorithm (25) written in the parametrization $\bar{T}_j$, with $\alpha = \delta t$.*

Corollary 16. *The standard CEM/ML update (24) is the IGO algorithm in parametrization $\bar{T}_j$ with $\delta t = 1$.*

Beware that the expectation parameters $\bar{T}_j$ are not always the most obvious parameters [AN00, Section 3.5]. For example, for 1-dimensional Gaussian distributions, these expectation parameters are the mean μ and the second moment $\mu^2 + \sigma^2$. When expressed back in terms of mean and variance, with the update (27) the new mean is $(1 - \delta t)\mu + \delta t \mu^*$, but the new variance is $(1 - \delta t)\sigma^2 + \delta t (\sigma^*)^2 + \delta t (1 - \delta t)(\mu^* - \mu)^2$.

On the other hand, when using smoothed CEM with mean and variance as parameters, the new variance is $(1 - \delta t)\sigma^2 + \delta t (\sigma^*)^2$, which can be significantly smaller for $\delta t \in (0, 1)$. This proves, in passing, that the smoothed CEM update in other parametrizations is generally *not* an IGO algorithm (because it can differ at first order in δt).

The case of Gaussian distributions is further exemplified in Section 4.2 below: in particular, smoothed CEM in the (μ, σ) parametrization exhibits premature reduction of variance, preventing good convergence.

For these reasons we think that the IGO-ML algorithm is the sensible way to perform an interpolated ML estimate for $\delta t < 1$, in a parametrization-independent way. In Section 6 we further discuss IGO and CEM and sum up the differences and relative advantages.

Taking $\delta t = 1$ is a bold approximation choice: the "ideal" continuous-time IGO flow itself, after time 1, does not coincide with the maximum likelihood update of the best points in the sample. Since the maximum likelihood algorithm is known to converge prematurely in some instances (Section 4.2), using the parametrization by expectation parameters with large δt may not be desirable.

The considerable simplification of the IGO update in these coordinates reflects the duality of coordinates $\bar{T}_i$ and θ_i. More precisely, the natural gradient ascent w.r.t. the parameters $\bar{T}_i$ is given by the vanilla gradient w.r.t. the parameters θ_i:

$$\widetilde{\nabla}_{\bar{T}_i} = \frac{\partial}{\partial \theta_i}$$

(Proposition 22 in the Appendix).

4 CMA-ES, NES, EDAs and PBIL from the IGO framework

In this section we investigate the IGO algorithm for Bernoulli measures and for multivariate normal distributions and show the correspondence to well-known algorithms. In addition, we discuss the influence of the parametrization of the distributions.

4.1 IGO algorithm for Bernoulli measures and PBIL

We consider on $X = \{0,1\}^d$ a family of Bernoulli measures $P_\theta(x) = p_{\theta_1}(x_1) \times \ldots \times p_{\theta_d}(x_d)$ with $p_{\theta_i}(x_i) = \theta_i^{x_i}(1-\theta_i)^{1-x_i}$. As this family is a product of probability measures $p_{\theta_i}(x_i)$, the different components of a random vector y following P_θ are independent and all off-diagonal terms of the Fisher information matrix (FIM) are zero. Diagonal terms are given by $\frac{1}{\theta_i(1-\theta_i)}$. Therefore the inverse of the FIM is a diagonal matrix with diagonal entries equal to $\theta_i(1-\theta_i)$. In addition, the partial derivative of $\ln P_\theta(x)$ w.r.t. θ_i can be computed in a straightforward manner resulting in

$$\frac{\partial \ln P_\theta(x)}{\partial \theta_i} = \frac{x_i}{\theta_i} - \frac{1-x_i}{1-\theta_i} \ .$$

Let $x_1, \ldots, x_N$ be N samples at step t with distribution P_{θ^t} and let $x_{1:N}, \ldots, x_{N:N}$ be the ranked samples according to f. The natural gradient update (15) with Bernoulli measures is then given by

$$\theta_i^{t+\delta t} = \theta_i^t + \delta t \, \theta_i^t(1-\theta_i^t) \sum_{j=1}^{N} w_j \left(\frac{[x_{j:N}]_i}{\theta_i^t} - \frac{1-[x_{j:N}]_i}{1-\theta_i^t} \right) \tag{28}$$

where $w_j = w((j-1/2)/N)/N$ and $[y]_i$ denotes the i^{th} coordinate of $y \in X$. The previous equation simplifies to

$$\theta_i^{t+\delta t} = \theta_i^t + \delta t \sum_{j=1}^{N} w_j \left([x_{j:N}]_i - \theta_i^t \right) \ ,$$

or, denoting $\bar{w}$ the sum of the weights $\sum_{j=1}^N w_j$,

$$\theta_i^{t+\delta t} = (1 - \bar{w}\delta t)\, \theta_i^t + \delta t \sum_{j=1}^{N} w_j \, [x_{j:N}]_i \ .$$

If we set the IGO weights as $w_1 = 1$, $w_j = 0$ for $j = 2, \dots, N$, we recover the PBIL/EGA algorithm with update rule towards the best solution only (disregarding the mutation of the probability vector), with $\delta t = \mathrm{LR}$ where LR is the so-called learning rate of the algorithm [Bal94, Figure 4]. The PBIL update rule towards the μ best solutions, proposed in [BC95, Figure 4][2], can be recovered as well using

$$\begin{aligned} \delta t &= \mathrm{LR} \\ w_j &= (1-\mathrm{LR})^{j-1}, \text{ for } j = 1, \dots, \mu \\ w_j &= 0, \text{ for } j = \mu+1, \dots, N \ . \end{aligned}$$

Interestingly, the parameters θ_i are the expectation parameters described in Section 3: indeed, the expectation of x_i is θ_i. So the formulas above are particular cases of (27). Thus, by Theorem 15, PBIL is both a smoothed CEM in these parameters and an IGO-ML algorithm.

Let us now consider another, so-called "logit" representation, given by the logistic function $P(x_i = 1) = \frac{1}{1+\exp(-\tilde{\theta}_i)}$. This $\tilde{\theta}$ is the exponential parametrization of Section 2.3. We find that

$$\frac{\partial \ln P_{\tilde{\theta}}(x)}{\partial \tilde{\theta}_i} = (x_i - 1) + \frac{\exp(-\tilde{\theta}_i)}{1+\exp(-\tilde{\theta}_i)} = x_i - \mathbb{E} x_i$$

(cf. (16)) and that the diagonal elements of the Fisher information matrix are given by $\exp(-\tilde{\theta}_i)/(1+\exp(-\tilde{\theta}_i))^2 = \mathrm{Var}\, x_i$ (compare with (17)). So the natural gradient update (15) with Bernoulli measures now reads

$$\tilde{\theta}_i^{t+\delta t} = \tilde{\theta}_i^t + \delta t (1+\exp(\tilde{\theta}_i^t)) \left(-\bar{w} + (1+\exp(-\tilde{\theta}_i^t)) \sum_{j=1}^{N} w_j [x_{j:N}]_i \right) \ .$$

To better compare the update with the previous representation, note that $\theta_i = \frac{1}{1+\exp(-\tilde{\theta}_i)}$ and thus we can rewrite

$$\tilde{\theta}_i^{t+\delta t} = \tilde{\theta}_i^t + \frac{\delta t}{\theta_i^t (1-\theta_i^t)} \sum_{j=1}^{N} w_j \left([x_{j:N}]_i - \theta_i^t \right) \ .$$

So the direction of the update is the same as before and is given by the proportion of bits set to 0 or 1 in the sample, compared to its expected value under the current distribution. The magnitude of the update is different since the parameter $\tilde{\theta}$ ranges from $-\infty$ to $+\infty$ instead of from 0 to 1. We did not find this algorithm in the literature.

These updates illustrate the influence of setting the sum of weights to 0 or not (Section 2.7). If, at some time, the first bit is set to 1 both for a majority of good points and for a majority of bad points, then the original PBIL will increase the probability of setting the first bit to 1, which is counterintuitive. If the weights w_i are chosen to sum to 0 this noise effect disappears; otherwise, it disappears only on average.

[2] Note that the pseudocode for the algorithm in [BC95, Figure 4] is slightly erroneous since it gives smaller weights to better individuals. The error can be fixed by updating the probability in reversed order, looping from `NUMBER_OF_VECTORS_TO_UPDATE_FROM` to 1. This was confirmed by S. Baluja in personal communication. We consider here the corrected version of the algorithm.

4.2 Multivariate normal distributions (Gaussians)

Evolution strategies [Rec73, Sch95, BS02] are black-box optimization algorithms for the *continuous* search domain, $X \subseteq \mathbb{R}^d$ (for simplicity we assume $X = \mathbb{R}^d$ in the following). They sample new solutions from a multivariate normal distribution. In the context of continuous black-box optimization, *Natural Evolution Strategies* (NES) introduced the idea of using a natural gradient update of the distribution parameters [WSPS08, SWSS09, GSS+10]. Surprisingly, also the well-known *Covariance Matrix Adaption Evolution Strategy*, CMA-ES [HO96, HO01, HMK03, HK04, JA06], turns out to conduct a natural gradient update of distribution parameters [ANOK10, GSS+10].

Let $x \in \mathbb{R}^d$. As the most prominent example, we use mean vector $m = \mathbb{E}x$ and covariance matrix $C = \mathbb{E}(x-m)(x-m)^{\mathrm{T}} = \mathbb{E}(xx^{\mathrm{T}}) - mm^{\mathrm{T}}$ to parametrize the distribution via $\theta = (m, C)$. The IGO update in (14) or (15) depends on the chosen parametrization, but can now be entirely reformulated without the (inverse) Fisher matrix, compare (18). The complexity of the update is linear in the number of parameters (size of $\theta = (m, C)$, where $(d^2 - d)/2$ parameters are redundant). We discuss known algorithms that implement variants of this update.

CMA-ES. The CMA-ES implements the equations[3]

$$m^{t+1} = m^t + \eta_{\mathrm{m}} \sum_{i=1}^{N} \widehat{w}_i (x_i - m^t) \tag{29}$$

$$C^{t+1} = C^t + \eta_{\mathrm{c}} \sum_{i=1}^{N} \widehat{w}_i ((x_i - m^t)(x_i - m^t)^T - C^t) \tag{30}$$

where $\widehat{w}_i$ are the weights based on ranked f-values, see (12) and (14).

When $\eta_{\mathrm{c}} = \eta_{\mathrm{m}}$, Equations (30) and (29) coincide with the IGO update (14) expressed in the parametrization (m, C) [ANOK10, GSS+10][4]. Note, however, that the learning rates η_{m} and η_{c} take essentially different values in CMA-ES, if $N \ll \dim \Theta$[5]: this is in deviation from an IGO algorithm. (Remark that the Fisher information matrix is block-diagonal in m and C [ANOK10], so that application of the different learning rates and of the inverse Fisher matrix commute.)

Natural evolution strategies. Natural evolution strategies (NES) [WSPS08, SWSS09] implement (29) as well, but use a Cholesky decomposition of C as parametrization for the update of the variance parameters. The resulting update that replaces (30) is neither particularly elegant nor numerically efficient. However, the most recent xNES [GSS+10] chooses an "exponential" parametrization depending on the current parameters. This leads to an elegant formulation where the additive update in exponential parametrization becomes a multiplicative update for C in (30). With $C = AA^{\mathrm{T}}$, the matrix

[3] The CMA-ES implements these equations given the parameter setting $c_1 = 0$ and $c_\sigma = 0$ (or $d_\sigma = \infty$, see e.g. [Han09]) that disengages the effect of both so-called evolution paths.

[4] In these articles the result has been derived for $\theta \leftarrow \theta + \eta \widetilde{\nabla}_\theta \mathbb{E}_{P_\theta} f$, see (9), leading to $f(x_i)$ in place of $\widehat{w}_i$. No assumptions on f have been used besides that it does not depend on θ. By replacing f with $W^f_{\theta^t}$, where θ^t is fixed, the derivation holds equally well for $\theta \leftarrow \theta + \eta \widetilde{\nabla}_\theta \mathbb{E}_{P_\theta} W^f_{\theta^t}$.

[5] Specifically, let $\sum |\widehat{w}_i| = 1$, then $\eta_{\mathrm{m}} = 1$ and $\eta_{\mathrm{c}} \approx 1 \wedge 1/(d^2 \sum \widehat{w}_i^2)$.

update reads

$$A \leftarrow A \times \exp\left(\frac{\eta_c}{2}\sum_{i=1}^{N}\widehat{w}_i(A^{-1}(x_i-m)(A^{-1}(x_i-m))^T - \mathrm{I}_d)\right) \tag{31}$$

where I_d is the identity matrix. (From (31) we get $C \leftarrow A \times \exp^2(\dots) \times A^{\mathrm{T}}$.) Remark that in the representation $\theta = (A^{-1}m, A^{-1}CA^{\mathrm{T}^{-1}}) = (A^{-1}m, \mathrm{I}_d)$, the Fisher information matrix becomes diagonal.

The update has the advantage over (30) that even negative weights always lead to feasible values. By default, $\eta_m \neq \eta_c$ in xNES in the same circumstances as in CMA-ES (most parameter settings are borrowed from CMA-ES), but contrary to CMA-ES the past evolution path is not taken into account [GSS+10].

When $\eta_c = \eta_m$, xNES is consistent with the IGO flow (6), and implements a slightly generalized IGO algorithm (14) using a θ-dependent parametrization.

Cross-entropy method and EMNA. *Estimation of distribution algorithms* (EDA) and the *cross-entropy method* (CEM) [Rub99, RK04] estimate a new distribution from a censored sample. Generally, the new parameter value can be written as

$$\begin{aligned}\theta_{\mathrm{maxLL}} &= \arg\max_\theta \sum_{i=1}^{N}\widehat{w}_i \ln P_\theta(x_i) \\ &\longrightarrow_{N\to\infty} \arg\max_\theta \mathbb{E}_{P_{\theta^t}} W^f_{\theta^t} \ln P_\theta\end{aligned} \tag{32}$$

For positive weights, θ_{maxLL} maximizes the weighted log-likelihood of $x_1 \dots x_N$. The argument under the arg max in the RHS of (32) is the negative cross-entropy between P_θ and the distribution of censored (elitist) samples $P_{\theta^t}W^f_{\theta^t}$ or of N realizations of such samples. The distribution $P_{\theta_{\mathrm{maxLL}}}$ has therefore minimal cross-entropy and minimal KL-divergence to the distribution of the μ best samples.[6] As said above, (32) recovers the *cross-entropy method* (CEM) [Rub99, RK04].

For Gaussian distributions, equation (32) can be explicitly written in the form

$$m^{t+1} = m^* = \sum_{i=1}^{N}\widehat{w}_i x_i \tag{33}$$

$$C^{t+1} = C^* = \sum_{i=1}^{N}\widehat{w}_i (x_i - m^*)(x_i - m^*)^{\mathrm{T}} \tag{34}$$

the empirical mean and variance of the elite sample. The weights $\widehat{w}_i$ are equal to $1/\mu$ for the μ best points and 0 otherwise.

Equations (33) and (34) also define the most fundamental continuous domain EDA, the *estimation of multivariate normal algorithm* ($\mathrm{EMNA}_{\mathrm{global}}$, [LL02]). It might be interesting to notice that (33) and (34) only differ from (29) and (30) in that the new mean m^{t+1} is used in the covariance matrix update.

[6] Let P_w denote the distribution of the weighted samples: $\Pr(x = x_i) = \widehat{w}_i$ and $\sum_i \widehat{w}_i = 1$. Then the cross-entropy between P_w and P_θ reads $\sum_i P_w(x_i) \ln 1/P_\theta(x_i)$ and the KL-divergence reads $\mathrm{KL}(P_w \,\|\, P_\theta) = \sum_i P_w(x_i) \ln 1/P_\theta(x_i) - \sum_i P_w(x_i) \ln 1/P_w(x_i)$. Minimization of both terms in θ result in θ_{maxLL}.

Let us compare IGO-ML (23), CMA (29)–(30), and smoothed CEM (25) in the parametrization with mean and covariance matrix. For learning rate $\delta t = 1$, IGO-ML and smoothed CEM/EMNA realize θ_{maxLL} from (32)–(34). For $\delta t < 1$ these algorithms all update the distribution mean in the same way; the update of mean and covariance matrix is computed to be

$$\begin{aligned} m^{t+1} &= (1-\delta t)\, m^t + \delta t\, m^* \\ C^{t+1} &= (1-\delta t)\, C^t + \delta t\, C^* + \delta t (1-\delta t)^j\, (m^* - m^t)(m^* - m^t)^{\mathrm{T}}, \end{aligned} \tag{35}$$

for different values of j, where m^* and C^* are the mean and covariance matrix computed over the elite sample (with weights $\widehat{w}_i$) as above. For CMA we have $j = 0$, for IGO-ML we have $j = 1$, and for smoothed CEM we have $j = \infty$ (the rightmost term is absent). The case $j = 2$ corresponds to an update that uses m^{t+1} instead of m^t in (30) (compare [Han06b]). For $0 < \delta t < 1$, the larger j, the smaller C^{t+1}.

The rightmost term of (35) resembles the so-called rank-one update in CMA.

For $\delta t \to 0$, the update is independent of j at first order in δt if $j < \infty$: this reflects compatibility with the IGO flow of CMA and of IGO-ML, but not of smoothed CEM.

Critical δt. Let us assume that the weights $\widehat{w}_i$ are non-negative, sum to one, and $\mu < N$ weights take a value of $1/\mu$, so that the selection quantile is $q = \mu/N$.

There is a critical value of δt depending on this quantile q, such that above the critical δt the algorithms given by IGO-ML and smoothed CEM are prone to premature convergence. Indeed, let f be a linear function on $\mathbb{R}^d$, and consider the variance in the direction of the gradient of f. Assuming further $N \to \infty$ and $q \leq 1/2$, then the variance C^* of the elite sample is smaller than the current variance C^t, by a constant factor. Depending on the precise update for C^{t+1}, if δt is too large, the variance C^{t+1} is going to be smaller than C^t by a constant factor as well. This implies that the algorithm is going to stall. (On the other hand, the continuous-time IGO flow corresponding to $\delta t \to 0$ does not stall, see Section 4.3.)

We now study the critical δt under which the algorithm does not stall. For IGO-ML, ($j = 1$ in (35), or equivalently for the smoothed CEM in the expectation parameters $(m, C + mm^{\mathrm{T}})$, see Section 3), the variance increases if and only if δt is smaller than the critical value $\delta t_{\text{crit}} = qb\sqrt{2\pi}e^{b^2/2}$ where b is the percentile function of q, i.e. b is such that $q = \int_b^\infty e^{-x^2/2}/\sqrt{2\pi}$. This value δt_{crit} is plotted as solid line in Fig. 1. For $j = 2$, δt_{crit} is smaller, related to the above by $\delta t_{\text{crit}} \leftarrow \sqrt{1 + \delta t_{\text{crit}}} - 1$ and plotted as dashed line in Fig. 1. For CEM ($j = \infty$), the critical δt is zero. For CMA-ES ($j = 0$), the critical δt is infinite for $q < 1/2$. When the selection quantile q is above $1/2$, for all algorithms the critical δt becomes zero.

We conclude that, despite the principled approach of ascending the natural gradient, the choice of the weighting function w, the choice of δt, and possible choices in the update for $\delta t > 0$, need to be taken with great care in relation to the choice of parametrization.

4.3 Computing the IGO flow for some simple examples

In this section we take a closer look at the IGO flow solutions of (6) for some simple examples of fitness functions, for which it is possible to obtain exact information about these IGO trajectories.

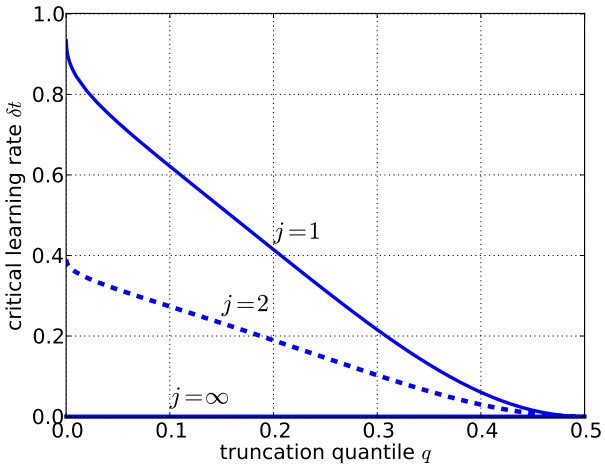

Figure 1: Critical δt versus truncation quantile q. Above the critical δt, the variance decreases systematically when optimizing a linear function. For CMA-ES/NES, the critical δt for $q < 0.5$ is infinite.

We start with the discrete search space $X = \{0,1\}^d$ and linear functions (to be minimized) defined as $f(x) = c - \sum_{i=1}^d \alpha_i x_i$ with $\alpha_i > 0$. Note that the onemax function to be maximized $f_{\text{onemax}}(x) = \sum_{i=1}^d x_i$ is covered by setting $\alpha_i = 1$. The differential equation (6) for the Bernoulli measures $P_\theta(x) = p_{\theta_1}(x_1) \dots p_{\theta_d}(x_d)$ defined on X can be derived taking the limit of the IGO-PBIL update given in (28):

$$\frac{\mathrm{d}\theta_i^t}{\mathrm{d}t} = \int W_{\theta^t}^f(x)(x_i - \theta_i^t) P_{\theta^t}(dx) =: g_i(\theta^t) \ . \tag{36}$$

Though finding the analytical solution of the differential equation (36) for any initial condition seems a bit intricate we show that the equation admits one critical stable point, $(1, \dots, 1)$, and one critical unstable point $(0, \dots, 0)$. In addition we prove that the trajectory decreases along f in the sense $\frac{\mathrm{d}f(\theta^t)}{\mathrm{d}t} \leqslant 0$. To do so we establish the following result:

Lemma 17. *On $f(x) = c - \sum_{i=1}^d \alpha_i x_i$ the solution of (36) satisfies $\sum_{i=1}^d \alpha_i \frac{\mathrm{d}\theta_i}{\mathrm{d}t} \geqslant 0$; moreover $\sum \alpha_i \frac{\mathrm{d}\theta_i}{\mathrm{d}t} = 0$ if and only if $\theta = (0, \dots, 0)$ or $\theta = (1, \dots, 1)$.*

Proof. We compute $\sum_{i=1}^d \alpha_i g_i(\theta^t)$ and find that

$$\begin{aligned}
\sum_{i=1}^d \alpha_i \frac{\mathrm{d}\theta_i^t}{\mathrm{d}t} &= \int W_{\theta^t}^f(x) \left(\sum_{i=1}^d \alpha_i x_i - \sum_{i=1}^d \alpha_i \theta_i^t \right) P_{\theta^t}(\mathrm{d}x) \\
&= \int W_{\theta^t}^f(x)(f(\theta^t) - f(x)) P_{\theta^t}(\mathrm{d}x) \\
&= \mathbb{E}[W_{\theta^t}^f(x)]\, \mathbb{E}[f(x)] - \mathbb{E}[W_{\theta^t}^f(x) f(x)]
\end{aligned}$$

In addition, $-W_{\theta^t}^f(x) = -w(\Pr(f(x') < f(x)))$ is a nondecreasing bounded function in the variable $f(x)$ such that $-W_{\theta^t}^f(x)$ and $f(x)$ are positively correlated (see [Tho00, Chapter 1] for a proof of this result), i.e.

$$\mathbb{E}[-W_{\theta^t}^f(x) f(x)] \geqslant \mathbb{E}[-W_{\theta^t}^f(x)] \mathbb{E}[f(x)]$$

with equality if and only if $\theta^t = (0, \dots, 0)$ or $\theta^t = (1, \dots, 1)$. Thus $\sum_{i=1}^d \alpha_i \frac{\mathrm{d}\theta_i^t}{\mathrm{d}t} \geqslant 0$. □

The previous result implies that the positive definite function $V(\theta) = \sum_{i=1}^d \alpha_i - \sum_{i=1}^d \alpha_i \theta_i$ in $(1, \dots, 1)$ satisfies $V^*(\theta) = \nabla V(\theta) \cdot g(\theta) \leqslant 0$ (such a function is called a Lyapunov function). Consequently $(1, \dots, 1)$ is stable. Similarly $V(\theta) = \sum_{i=1}^d \alpha_i \theta_i$ is a Lyapunov function for $(0, \dots, 0)$ such that $\nabla V(\theta) \cdot g(\theta) \geqslant 0$. Consequently $(0, \dots, 0)$ is unstable [AO08].

We now consider on $\mathbb{R}^d$ the family of multivariate normal distributions $P_\theta = \mathcal{N}(m, \sigma^2 I_d)$ with covariance matrix equal to $\sigma^2 I_d$. The parameter θ thus has $d+1$ components $\theta = (m, \sigma) \in \mathbb{R}^d \times \mathbb{R}$. The natural gradient update using this family was derived in [GSS+10]; from it we can derive the IGO differential equation which reads:

$$\frac{\mathrm{d}m^t}{\mathrm{d}t} = \int_{\mathbb{R}^d} W^f_{\theta^t}(x)(x - m^t) p_{\mathcal{N}(m^t, (\sigma^t)^2 I_d)}(x) dx \tag{37}$$

$$\frac{\mathrm{d}\tilde{\sigma}^t}{\mathrm{d}t} = \int_{\mathbb{R}^d} \frac{1}{2d} \left\{ \sum_{i=1}^d \left(\frac{x_i - m_i^t}{\sigma^t} \right)^2 - 1 \right\} W^f_{\theta^t}(x) p_{\mathcal{N}(m^t, (\sigma^t)^2 I_d)}(x) dx \tag{38}$$

where σ^t and $\tilde{\sigma}^t$ are linked via $\sigma^t = \exp(\tilde{\sigma}^t)$ or $\tilde{\sigma}^t = \ln(\sigma^t)$. Denoting $\mathcal{N}$ a random vector following a centered multivariate normal distribution with covariance matrix identity we write equivalently the gradient flow as

$$\frac{\mathrm{d}m^t}{\mathrm{d}t} = \sigma^t \mathbb{E}\left[W^f_{\theta^t}(m^t + \sigma^t \mathcal{N}) \mathcal{N} \right] \tag{39}$$

$$\frac{\mathrm{d}\tilde{\sigma}^t}{\mathrm{d}t} = \mathbb{E}\left[\frac{1}{2} \left(\frac{\|\mathcal{N}\|^2}{d} - 1 \right) W^f_{\theta^t}(m^t + \sigma^t \mathcal{N}) \right] \ . \tag{40}$$

Let us analyze the solution of the previous system on linear functions. Without loss of generality (because of invariance) we can consider the linear function $f(x) = x_1$. We have

$$W^f_{\theta^t}(x) = w(\Pr(m_1^t + \sigma^t Z_1 < x_1))$$

where Z_1 follows a standard normal distribution and thus

$$W^f_{\theta^t}(m^t + \sigma^t \mathcal{N}) = w(\Pr_{Z_1 \sim \mathcal{N}(0,1)}(Z_1 < \mathcal{N}_1)) \tag{41}$$

$$= w(\mathcal{F}(\mathcal{N}_1)) \tag{42}$$

with $\mathcal{F}$ the cumulative distribution of a standard normal distribution. The differential equation thus simplifies into

$$\frac{\mathrm{d}m^t}{\mathrm{d}t} = \sigma^t \begin{pmatrix} \mathbb{E}\left[w(\mathcal{F}(\mathcal{N}_1)) \mathcal{N}_1 \right] \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{43}$$

$$\frac{\mathrm{d}\tilde{\sigma}^t}{\mathrm{d}t} = \frac{1}{2d} \mathbb{E}\left[(|\mathcal{N}_1|^2 - 1) w(\mathcal{F}(\mathcal{N}_1)) \right] \ . \tag{44}$$

Consider, for example, the weight function associated with truncation selection, i.e. $w(q) = 1_{q \leqslant q_0}$ where $q_0 \in]0, 1]$—also called intermediate recombination. We find that

$$\frac{\mathrm{d}m_1^t}{\mathrm{d}t} = \sigma^t \mathbb{E}[\mathcal{N}_1 1_{\{\mathcal{N}_1 \leqslant \mathcal{F}^{-1}(q_0)\}}] =: \sigma^t \beta \tag{45}$$

$$\frac{\mathrm{d}\tilde{\sigma}^t}{\mathrm{d}t} = \frac{1}{2d} \left(\int_0^{q_0} \mathcal{F}^{-1}(u)^2 du - q_0 \right) =: \alpha \ . \tag{46}$$

The solution of the IGO flow for the linear function $f(x) = x_1$ is thus given by

$$m_1^t = m_0^t + \frac{\sigma^0 \beta}{\alpha} \exp(\alpha t) \tag{47}$$

$$\sigma^t = \sigma^0 \exp(\alpha t) \ . \tag{48}$$

The coefficient β is strictly negative for any $q_0 < 1$. The coefficient α is strictly positive if and only if $q_0 < 1/2$ which corresponds to selecting less than half of the sampled points in an ES. In this case the step-size σ^t grows exponentially fast to infinity and the mean vectors moves along the gradient direction towards minus ∞ at the same rate. If more than half of the points are selected, $q_0 \geqslant 1/2$, the step-size will decrease to zero exponentially fast and the mean vector will get stuck (compare also [Han06a]).

5 Multimodal optimization using restricted Boltzmann machines

We now illustrate experimentally the influence of the natural gradient, versus vanilla gradient, on diversity over the course of optimization. We consider a very simple situation of a fitness function with two distant optima and test whether different algorithms are able to reach both optima simultaneously or only converge to one of them. This provides a practical test of Proposition 1 stating that the natural gradient minimizes loss of diversity.

The IGO method allows to build a natural search algorithm from an arbitrary probability distribution on an arbitrary search space. In particular, by choosing families of probability distributions that are richer than Gaussian or Bernoulli, one may hope to be able to optimize functions with complex shapes. Here we study how this might help optimize multimodal functions.

Both Gaussian distributions on $\mathbb{R}^d$ and Bernoulli distributions on $\{0,1\}^d$ are unimodal. So at any given time, a search algorithm using such distributions concentrates around a given point in the search space, looking around that point (with some variance). It is an often-sought-after feature for an optimization algorithm to handle multiple hypotheses simultaneously.

In this section we apply our method to a multimodal distribution on $\{0,1\}^d$: restricted Boltzmann machines (RBMs). Depending on the activation state of the *latent variables*, values for various blocks of bits can be switched on or off, hence multimodality. So the optimization algorithm derived from these distributions will, hopefully, explore several distant zones of the search space at any given time. A related model (Boltzmann machines) was used in [Ber02] and was found to perform better than PBIL on some functions.

Our study of a bimodal RBM distribution for the optimization of a bimodal function confirms that the natural gradient does indeed behave in a more natural way than the vanilla gradient: when initialized properly, the natural gradient is able to maintain diversity by fully using the RBM distribution to learn the two modes, while the vanilla gradient only converges to one of the two modes.

Although these experiments support using a natural gradient approach, they also establish that complications can arise for estimating the inverse Fisher matrix in the case of complex distributions such as RBMs: estimation errors may lead to a singular or unreliable estimation of the Fisher

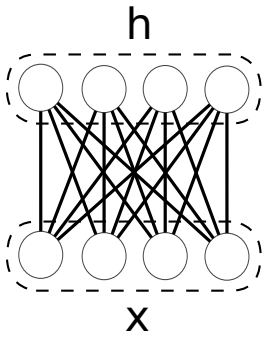

Figure 2: The RBM architecture with the observed ($\mathbf{x}$) and latent ($\mathbf{h}$) variables. In our experiments, a single hidden unit was used.

matrix, especially when the distribution becomes singular. Further research is needed for a better understanding of this issue.

The experiments reported here, and the fitness function used, are extremely simple from the optimization viewpoint: both algorithms using the natural and vanilla gradient find an optimum in only a few steps. The emphasis here is on the specific influence of replacing the vanilla gradient with the natural gradient, and the resulting influence on diversity and multimodality, in a simple situation.

5.1 IGO for restricted Boltzmann machines

Restricted Boltzmann machines. A restricted Boltzmann machine (RBM) [Smo86, AHS85] is a kind of probability distribution belonging to the family of undirected graphical models (also known as a Markov random fields). A set of observed variables $\mathbf{x} \in \{0,1\}^{n_x}$ are given a probability using their joint distribution with unobserved latent variables $\mathbf{h} \in \{0,1\}^{n_h}$ [Gha04]. The latent variables are then marginalized over. See Figure 2 for the graph structure of a RBM.

The probability associated with an observation $\mathbf{x}$ and latent variable $\mathbf{h}$ is given by

$$P_\theta(\mathbf{x},\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{\sum_{\mathbf{x}',\mathbf{h}'} e^{-E(\mathbf{x}',\mathbf{h}')}}, \qquad P_\theta(\mathbf{x}) = \sum_{\mathbf{h}} P_\theta(\mathbf{x},\mathbf{h}). \tag{49}$$

where $E(\mathbf{x},\mathbf{h})$ is the so-called energy function and $\sum_{\mathbf{x}',\mathbf{h}'} e^{-E(\mathbf{x}',\mathbf{h}')}$ is the partition function denoted Z in Section 2.3. The energy E is defined by

$$E(\mathbf{x},\mathbf{h}) = -\sum_i a_i x_i - \sum_j b_j h_j - \sum_{i,j} w_{ij} x_i h_j \,. \tag{50}$$

The distribution is fully parametrized by the bias on the observed variables $\mathbf{a}$, the bias on the latent variables $\mathbf{b}$ and the weights $\mathbf{W}$ which account for pairwise interactions between observed and latent variables: $\theta = (\mathbf{a},\mathbf{b},\mathbf{W})$.

RBM distributions are a special case of exponential family distributions with latent variables (see (21) in Section 2.3). The RBM IGO equations stem from Equations (16), (17) and (18) by identifying the statistics $T(x)$ with x_i, h_j or $x_i h_j$.

For these distributions, the gradient of the log-likelihood is well-known [Hin02]. Although it is often considered intractable in the context of machine learning where a lot of variables are required, it becomes tractable for smaller

RBMs. The gradient of the log-likelihood consists of the difference of two expectations (compare (16)):

$$\frac{\partial \ln P_\theta(x)}{\partial w_{ij}} \quad = \quad \mathbb{E}_{P_\theta}[x_i h_j | x] - \mathbb{E}_{P_\theta}[x_i h_j] \tag{51}$$

The Fisher information matrix is given by (20)

$$I_{w_{ab} w_{cd}}(\theta) = \mathbb{E}[x_a h_b x_c h_d] - \mathbb{E}[x_a h_b]\mathbb{E}[x_c h_d] \tag{52}$$

where $I_{w_{ab}w_{cd}}$ denotes the entry of the Fisher matrix corresponding to the components w_{ab} and w_{cd} of the parameter θ. These equations are understood to encompass the biases $\mathbf{a}$ and $\mathbf{b}$ by noticing that the bias can be replaced in the model by adding two variables x_k and h_k always equal to one.

Finally, the IGO update rule is taken from (14):

$$\theta^{t+\delta t} = \theta^t + \delta t \, I^{-1}(\theta^t) \sum_{k=1}^{N} \widehat{w}_k \frac{\partial \ln P_\theta(\mathbf{x}_k)}{\partial \theta}\bigg|_{\theta=\theta^t} \tag{53}$$

Implementation. In this experimental study, the IGO algorithm for RBMs is directly implemented from Equation (53). At each optimization step, the algorithm consists in (1) finding a reliable estimate of the Fisher matrix (see Eq. 52) which is then inverted using the QR-Algorithm if it is invertible; (2) computing an estimate of the vanilla gradient which is then weighted according to W_θ^f; and (3) updating the parameters, taking into account the gradient step size δt. In order to estimate both the Fisher matrix and the gradient, samples must be drawn from the model P_θ. This is done using Gibbs sampling (see for instance [Hin02]).

Fisher matrix imprecision. The imprecision incurred by the limited sampling size may sometimes lead to a singular estimation of the Fisher matrix (see p. 11 for a lower bound on the number of samples needed). Although having a singular Fisher estimation happens rarely in normal conditions, it occurs with certainty when the probabilities become too concentrated over a few points. This situation arises naturally when the algorithm is allowed to continue the optimization after the optimum has been reached. For this reason, in our experiments, we stop the optimization as soon as both optima have been sampled, thus preventing P_θ from becoming too concentrated.

In a variant of the same problem, the Fisher estimation can become close to singular while still being numerically invertible. This situation leads to unreliable estimates and should therefore be avoided. To evaluate the reliability of the inverse Fisher matrix estimate, we use a cross-validation method: (1) making two estimates $\hat{F}_1$ and $\hat{F}_2$ of the Fisher matrix on two distinct sets of points generated from P_θ, and (2) making sure that the eigenvalues of $\hat{F}_1 \times \hat{F}_2^{-1}$ are close to 1. In practice, at all steps we check that the average of the eigenvalues of $\hat{F}_1 \times \hat{F}_2^{-1}$ is between $1/2$ and 2. If at some point during the optimization the Fisher estimation becomes singular or unreliable according to this criterion, the corresponding run is stopped and considered to have failed.

When optimizing on a larger space helps. A restricted Boltzmann machine defines naturally a distribution P_θ on both visible and hidden units $(\mathbf{x}, \mathbf{h})$, whereas the function to optimize depends only on the visible units

$\mathbf{x}$. Thus we are faced with a choice. A first possibility is to decide that the objective function $f(\mathbf{x})$ is really a function of $(\mathbf{x}, \mathbf{h})$ where $\mathbf{h}$ is a dummy variable; then we can use the IGO algorithm to optimize over $(\mathbf{x}, \mathbf{h})$ using the distributions $P_\theta(\mathbf{x}, \mathbf{h})$. A second possibility is to marginalize $P_\theta(\mathbf{x}, \mathbf{h})$ over the hidden units $\mathbf{h}$ as in (49), to define the distribution $P_\theta(\mathbf{x})$; then we can use the IGO algorithm to optimize over $\mathbf{x}$ using $P_\theta(\mathbf{x})$.

These two approaches yield slightly different algorithms. Both were tested and found to be viable. However the first approach is numerically more stable and requires less samples to estimate the Fisher matrix. Indeed, if $I_1(\theta)$ is the Fisher matrix at θ in the first approach and $I_2(\theta)$ in the second approach, we always have $I_1(\theta) \geqslant I_2(\theta)$ (in the sense of positive-definite matrices). This is because probability distributions on the pair $(\mathbf{x}, \mathbf{h})$ carry more information than their projections on $\mathbf{x}$ only, and so computed Kullback–Leibler distances will be larger.

In particular, there are (isolated) values of θ for which the Fisher matrix I_2 is not invertible whereas I_1 is. For this reason, we selected the first approach.

5.2 Experimental results

In our experiments, we look at the optimization of the two-min function defined below with a bimodal RBM: an RBM with only one latent variable. Such an RBM is bimodal because it has two possible configurations of the latent variable: $\mathbf{h} = 0$ or $\mathbf{h} = 1$, and given $\mathbf{h}$, the observed variables are independent and distributed according to two Bernoulli distributions.

Set a parameter $\mathbf{y} \in \{0, 1\}^d$. The two-min function is defined as follows:

$$f_{\mathbf{y}}(\mathbf{x}) = \min \left(\sum_i |x_i - y_i| \, , \sum_i |(1 - x_i) - y_i)| \right) \tag{54}$$

This function of $\mathbf{x}$ has two optima: one at $\mathbf{y}$, the other at its binary complement $\bar{\mathbf{y}}$.

For the quantile rewriting of f (Section 1.2), we chose the function w to be $w(q) = \mathbb{1}_{q \leqslant 1/2}$ so that points which are below the median are given the weight 1, whereas other points are given the weight 0. Also, in accordance with (3), if several points have the same fitness value, their weight W_θ^f is set to the average of w over all those points.

For initialization of the RBMs, the weights $\mathbf{W}$ are sampled from a normal distribution centered around zero and of standard deviation $1/\sqrt{n_x \times n_h}$, where n_x is the number of observed variables (dimension d of the problem) and n_h is the number of latent variables ($n_h = 1$ in our case), so that initially the energies E are not too large. Then the bias parameters are initialized as $b_j \leftarrow -\sum_i \frac{w_{ij}}{2}$ and $a_i \leftarrow -\sum_j \frac{w_{ij}}{2} + \mathcal{N}(\frac{0.01}{n_x^2})$ so that each variable (observed or latent) has a probability of activation close to $1/2$.

In the following experiments, we show the results of IGO optimization and vanilla gradient optimization for the two-min function in dimension 40, for various values of the step size δt. For each δt, we present the median of the quantity of interest over 100 runs. Error bars indicate the 16th percentile and the 84th percentile (this is the same as mean$\pm$stddev for a Gaussian variable, but is invariant by f-reparametrization). For each run, the parameter $\mathbf{y}$ of the two-max function is sampled randomly in order to ensure that the presented results are not dependent on a particular choice of optima.

The number of sample points used for estimating the Fisher matrix is set to $10,000$: large enough (by far) to ensure the stability of the estimates.

The same points are used for estimating the integral of (14), therefore there are $10,000$ calls to the fitness function at each gradient step. These rather comfortable settings allow for a good illustration of the theoretical properties of the $N = \infty$ IGO flow limit.

The numbers of runs that fail after the occurrence of a singular matrix or an unreliable estimate amount for less than 10% for $\delta t \leqslant 2$ (as little as 3% for the smallest learning rate), but can increase up to 30% for higher learning rates.

5.2.1 Convergence

We first check that both vanilla and natural gradient are able to converge to an optimum.

Figures 3 and 4 show the fitness of the best sampled point for the IGO algorithm and for the vanilla gradient at each step. Predictably, both algorithms are able to optimize the two-min function in a few steps.

The two-min function is extremely simple from an optimization viewpoint; thus, convergence speed is not the main focus here, all the more since we use a large number of f-calls at each step.

Note that the values of the parameter δt for the two gradients used are not directly comparable from a theoretical viewpoint (they correspond to parametrizations of different trajectories in Θ-space, and identifying vanilla δt with natural δt is meaningless). We selected larger values of δt for the vanilla gradient, in order to obtain roughly comparable convergence speeds in practice.

5.2.2 Diversity

As we have seen, the two-min function is equally well optimized by the IGO and vanilla gradient optimization. However, the methods fare very differently when we look at the distance from the sample points to *both* optima. From (54), the fitness gives the distance of sample points to the closest optimum. We now study how close sample points come to the *other*, opposite optimum. The distance of sample points to the second optimum is shown in Figure 5 for IGO, and in Figure 6 for the vanilla gradient.

Figure 5 shows that IGO also reaches the second optimum most of the time, and is often able to find it only a few steps after the first. This property of IGO is of course dependent on the initialization of the RBM with enough diversity. When initialized properly so that each variable (observed and latent) has a probability $1/2$ of being equal to 1, the initial RBM distribution has maximal diversity over the search space and is at equal distance from the two optima of the function. From this starting position, the natural gradient is then able to increase the likelihood of the two optima at the same time.

By stark contrast, the vanilla gradient is not able to go towards both optima at the same time as shown in Fig. 6. In fact, the vanilla gradient only converges to one optimum at the expense of the other. For all values of δt, the distance to the second optimum increases gradually and approaches the maximum possible distance.

As mentioned earlier, each state of the latent variable $\mathbf{h}$ corresponds to a mode of the distribution. In Figures 7 and 8, we look at the average value of $\mathbf{h}$ for each gradient step. An average value close to $1/2$ means that the distribution samples from both modes: $\mathbf{h} = 0$ or $\mathbf{h} = 1$ with a comparable probability. Conversely, average values close to 0 or 1 indicate

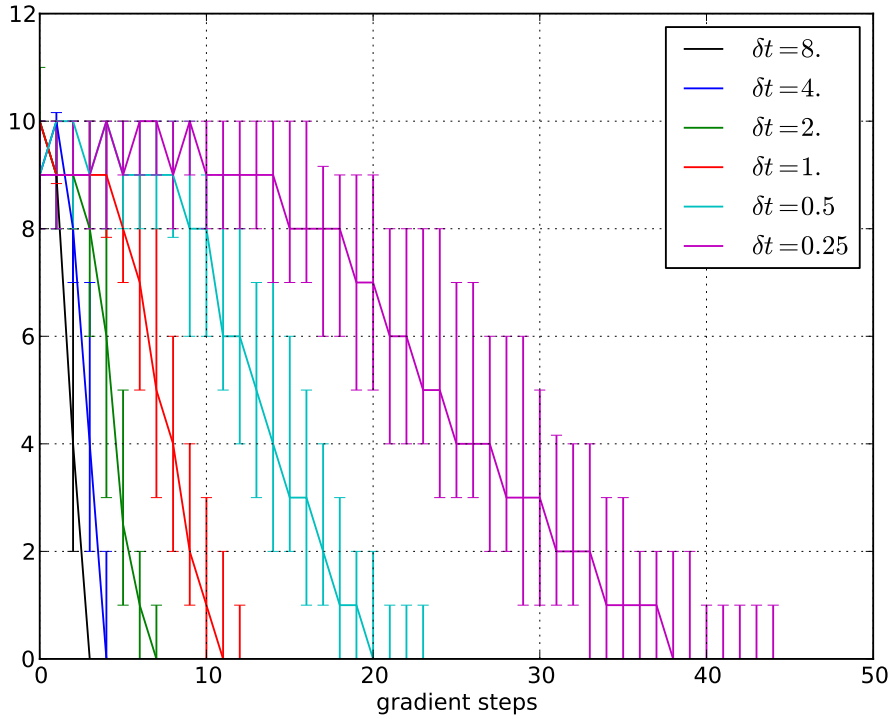

Figure 3: Fitness of sampled points during IGO optimization.

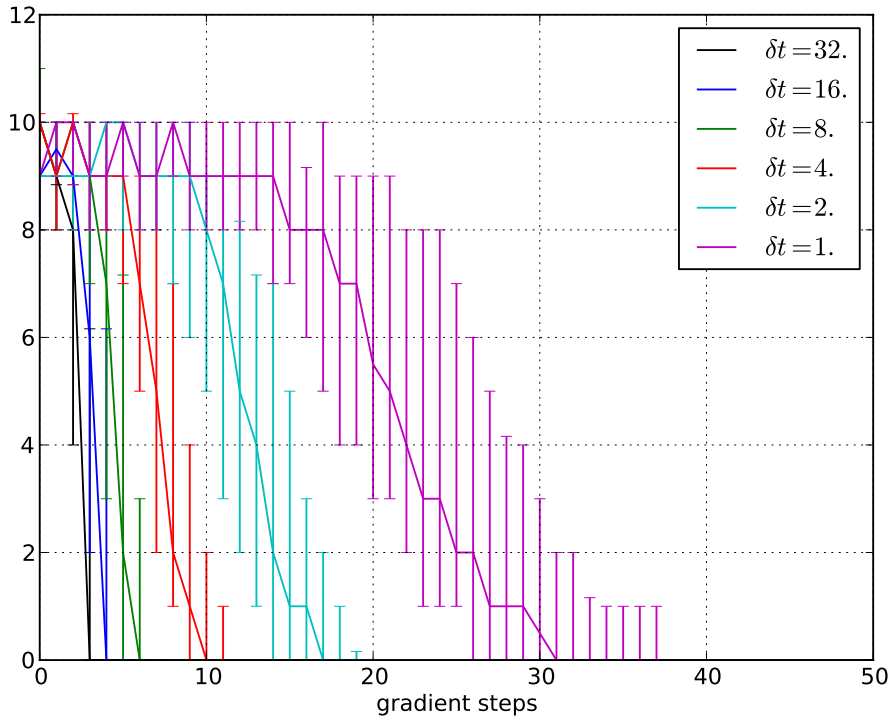

Figure 4: Fitness of sampled points during vanilla gradient optimization.

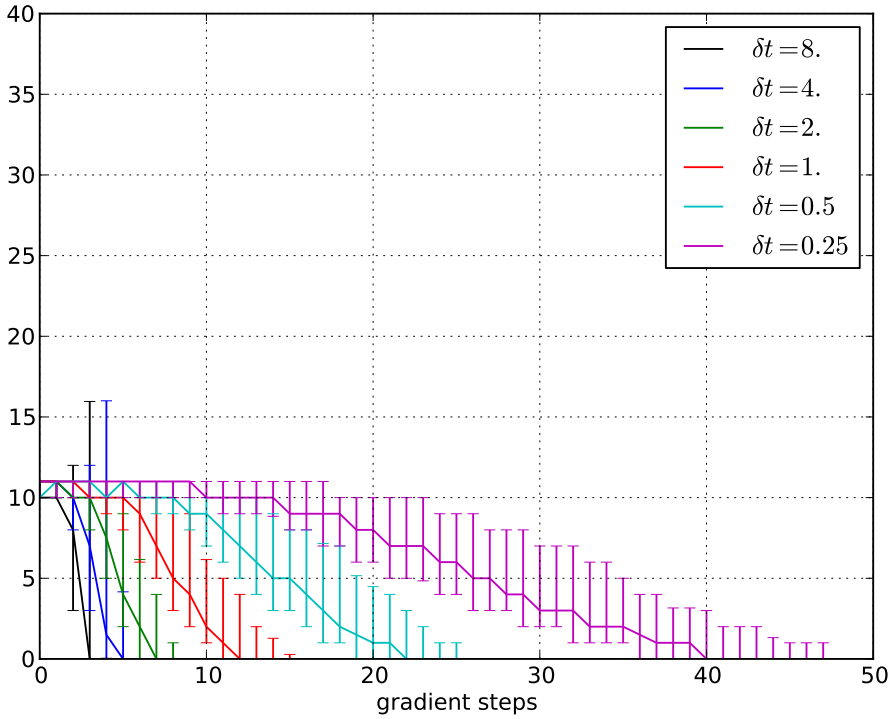

Figure 5: Distance to the second optimum during IGO optimization.

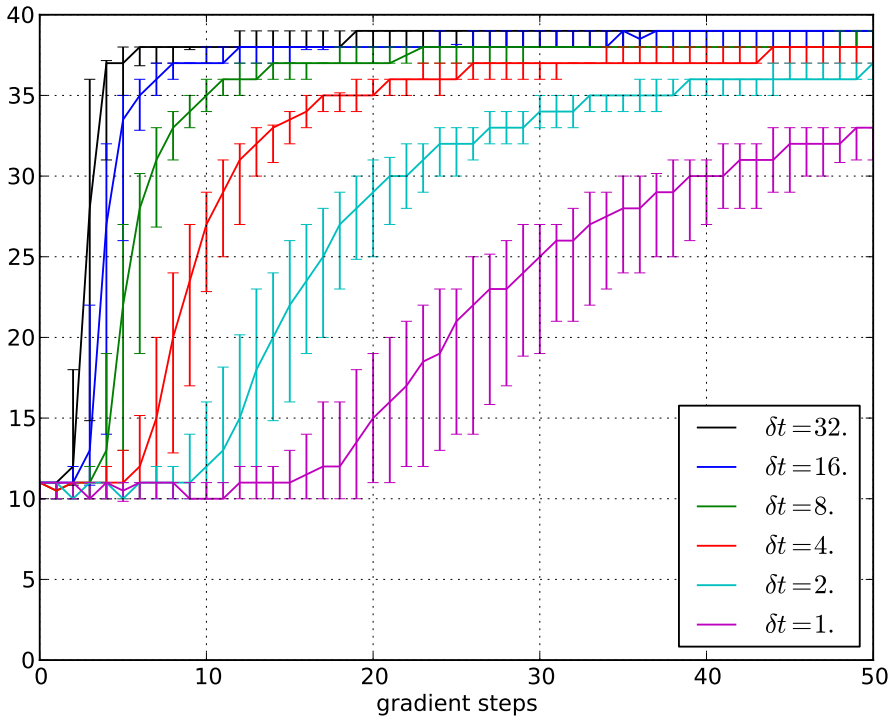

Figure 6: Distance to second optimum during vanilla gradient optimization.

that the distribution gives most probability to one mode at the expense of the other. In Figure 7, we can see that with IGO, the average value of $\mathbf{h}$ is close to 1/2 during the whole optimization procedure for most runs: the distribution is initialized with two modes and stays bimodal[7]. As for the vanilla gradient, statistics for $\mathbf{h}$ are depicted in Figure 8 and we can see that they converge to 1: one of the two modes of the distribution has been lost during optimization.

Hidden breach of symmetry by the vanilla gradient. The experiments reveal a curious phenomenon: the vanilla gradient loses multimodality by always setting the hidden variable h to 1, not to 0. (We detected no obvious asymmetry on the visible units x, though.)

Of course, exchanging the values 0 and 1 for the hidden variables in a restricted Boltzmann machine still gives a distribution of another Boltzmann machine. More precisely, changing h_j into $1 - h_j$ is equivalent to resetting $a_i \leftarrow a_i + w_{ij}$, $b_j \leftarrow -b_j$, and $w_{ij} \leftarrow -w_{ij}$. IGO and the natural gradient are impervious to such a change by Proposition 9.

The vanilla gradient implicitly relies on the Euclidean norm on parameter space, as explained in Section 1.1. For this norm, the distance between the RBM distributions (a_i, b_j, w_{ij}) and (a'_i, b'_j, w'_{ij}) is simply $\sum_i |a_i - a'_i|^2 + \sum_j \left|b_j - b'_j\right|^2 + \sum_{ij} \left|w_{ij} - w'_{ij}\right|^2$. However, the change of variables $a_i \leftarrow a_i + w_{ij}$, $b_j \leftarrow -b_j$, $w_{ij} \leftarrow -w_{ij}$ does *not* preserve this Euclidean metric. Thus, exchanging 0 and 1 for the hidden variables will result in two different vanilla gradient ascents. The observed asymmetry on h is a consequence of this implicit asymmetry.

The same asymmetry exists for the visible variables x_i; but this does not prevent convergence to an optimum in our situation, since any gradient descent eventually reaches some local optimum.

Of course it is possible to cook up parametrizations for which the vanilla gradient will be more symmetric: for instance, using $-1/1$ instead of $0/1$ for the variables, or defining the energy by

$$E(\mathbf{x}, \mathbf{h}) = -\sum_i A_i(x_i - \tfrac{1}{2}) - \sum_j B_j(h_j - \tfrac{1}{2}) - \sum_{i,j} W_{ij}(x_i - \tfrac{1}{2})(h_j - \tfrac{1}{2}) \quad (55)$$

with "bias-free" parameters A_i, B_j, W_{ij} related to the usual parametrization by $w_{ij} = W_{ij}$, $a_i = A_i - \frac{1}{2}\sum_j w_{ij}$ $b_j = B_j - \frac{1}{2}\sum_i w_{ij}$. The vanilla gradient might perform better in these parametrizations.

However, we chose a naive approach: we used a family of probability distributions found in the literature, with the parametrization found in the literature. We then use the vanilla gradient and the natural gradient on these distributions. This directly illustrates the specific influence of the chosen gradient (the two implementations only differ by the inclusion of the Fisher matrix). It is remarkable, we think, that the natural gradient is able to recover symmetry where there was none.

5.3 Convergence to the continuous-time limit

In the previous figures, it looks like changing the parameter δt only results in a time speedup of the plots.

[7] In Fig. 7, we use the following adverse setting: runs are interrupted once they reach both optima, therefore the statistics are taken only over those runs which have *not yet* converged and reached both optima, which results in higher variation around 1/2. The plot has been stopped when less than half the runs remain. The error bars are relative only to the remaining runs.

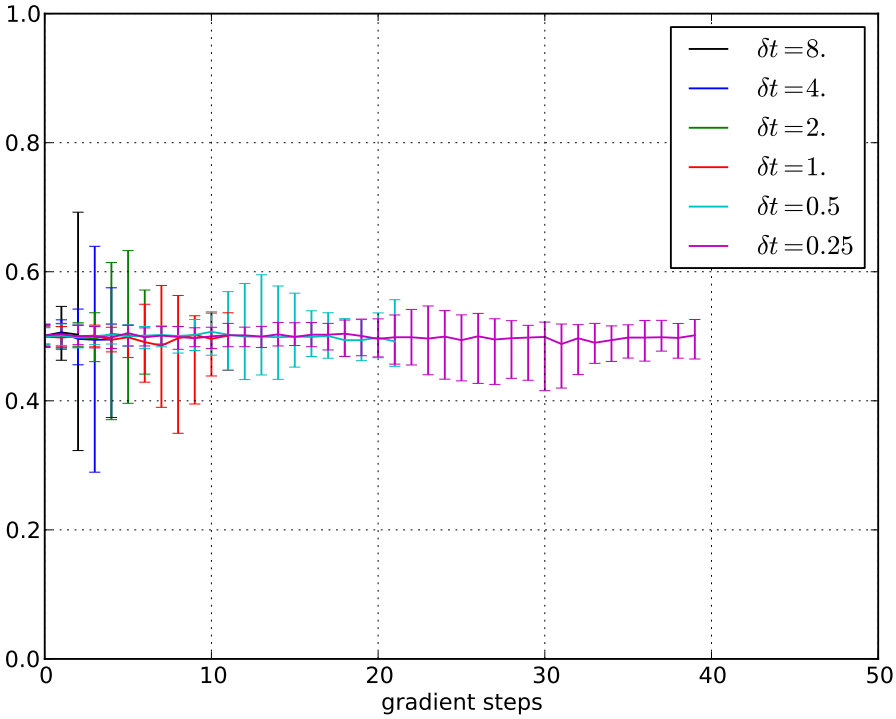

Figure 7: Average value of $\mathbf{h}$ during IGO optimization.

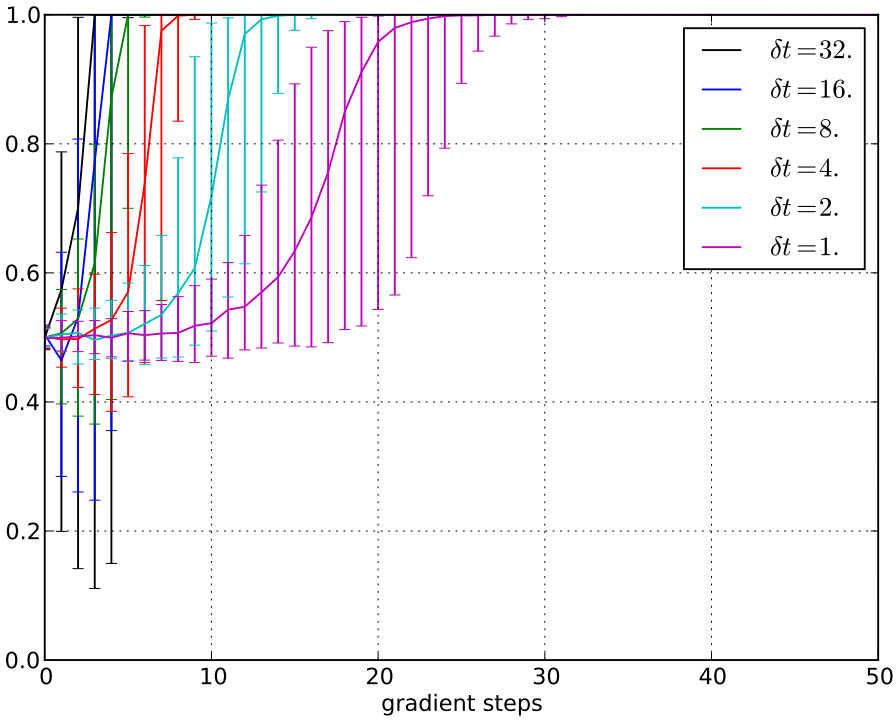

Figure 8: Average value of $\mathbf{h}$ during vanilla gradient optimization.

Update rules of the type $\theta \leftarrow \theta + \delta t \, \nabla_\theta g$ (for either gradient) are Euler approximations of the continuous-time ordinary differential equation $\frac{\mathrm{d}\theta}{\mathrm{d}t} = \nabla_\theta g$, with each iteration corresponding to an increment δt of the time t. Thus, it is expected that for small enough δt, the algorithm after k steps approximates the IGO flow or vanilla gradient flow at time $t = k.\delta t$.

Figures 9 and 10 illustrate this convergence: we show the fitness w.r.t to δt times the number of gradient steps. An asymptotic trajectory seems to emerge when δt decreases. For the natural gradient, it can be interpreted as the fitness of samples of the continuous-time IGO flow.

Thus, for this kind of optimization algorithms, it makes theoretical sense to plot the results according to the "intrinsic time" of the underlying continuous-time object, to illustrate properties that do not depend on the setting of the parameter δt. (Still, the raw number of steps is more directly related to algorithmic cost.)

6 Further discussion

A single framework for optimization on arbitrary spaces. A strength of the IGO viewpoint is to automatically provide optimization algorithms using any family of probability distributions on any given space, discrete or continuous. This has been illustrated with restricted Boltzmann machines. IGO algorithms also feature good invariance properties and make a least number of arbitrary choices.

In particular, IGO unifies several well-known optimization algorithms into a single framework. For instance, to the best of our knowledge, PBIL has never been described as a natural gradient ascent in the literature[8].

For Gaussian measures, algorithms of the same form (14) had been developed previously [HO01, WSPS08] and their close relationship with a natural gradient ascent had been recognized [ANOK10, GSS+10].

The wide applicability of natural gradient approaches seems not to be widely known in the optimization community, though see [MMS08].

About quantiles. The IGO flow, to the best of our knowledge, has not been defined before. The introduction of the quantile-rewriting (3) of the objective function provides the first rigorous derivation of quantile- or rank-based natural optimization from a gradient ascent in θ-space.

Indeed, NES and CMA-ES have been claimed to maximize $-\mathbb{E}_{P_\theta} f$ via natural gradient ascent [WSPS08, ANOK10]. However, we have proved that when the number of samples is large and the step size is small, the NES and CMA-ES updates converge to the IGO flow, not to the similar flow with the gradient of $\mathbb{E}_{P_\theta} f$ (Theorem 4). So we find that in reality these algorithms maximize $\mathbb{E}_{P_\theta} W^f_{\theta^t}$, where $W^f_{\theta^t}$ is a decreasing transformation of the f-quantiles under the current sample distribution.

Also in practice, maximizing $-\mathbb{E}_{P_\theta} f$ is a rather unstable procedure and has been discouraged, see for example [Whi89].

About choice of P_θ: learning a model of good points. The choice of the family of probability distributions P_θ plays a double role.

First, it is analogous to a *mutation operator* as seen in evolutionary algorithms: indeed, P_θ encodes possible moves according to which new sample points are explored.

[8] Thanks to Jonathan Shapiro for an early argument confirming this property (personal communication).

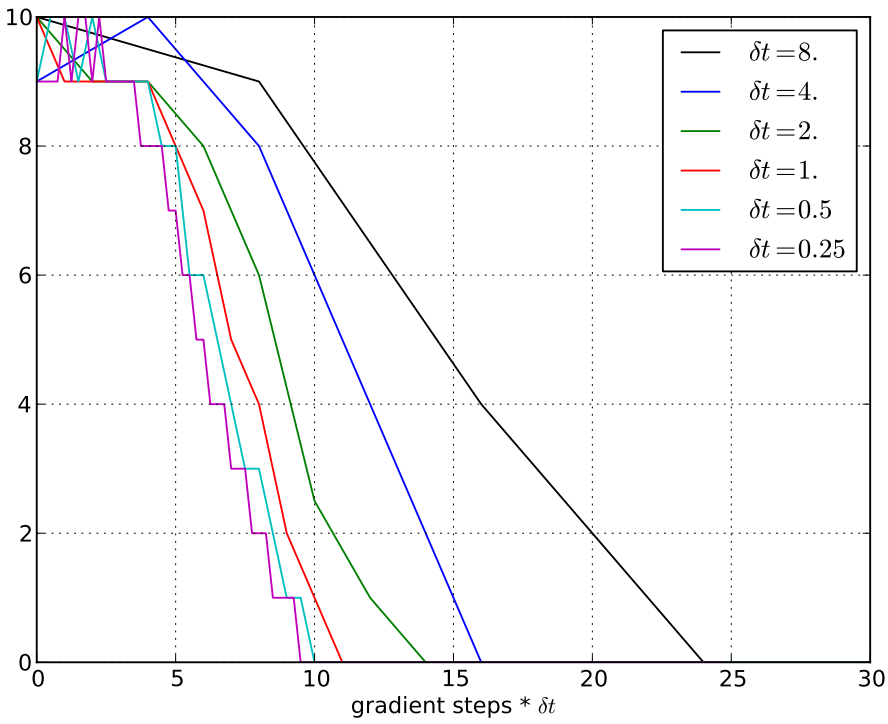

Figure 9: Fitness of sampled points w.r.t. "intrinsic time" during IGO optimization.

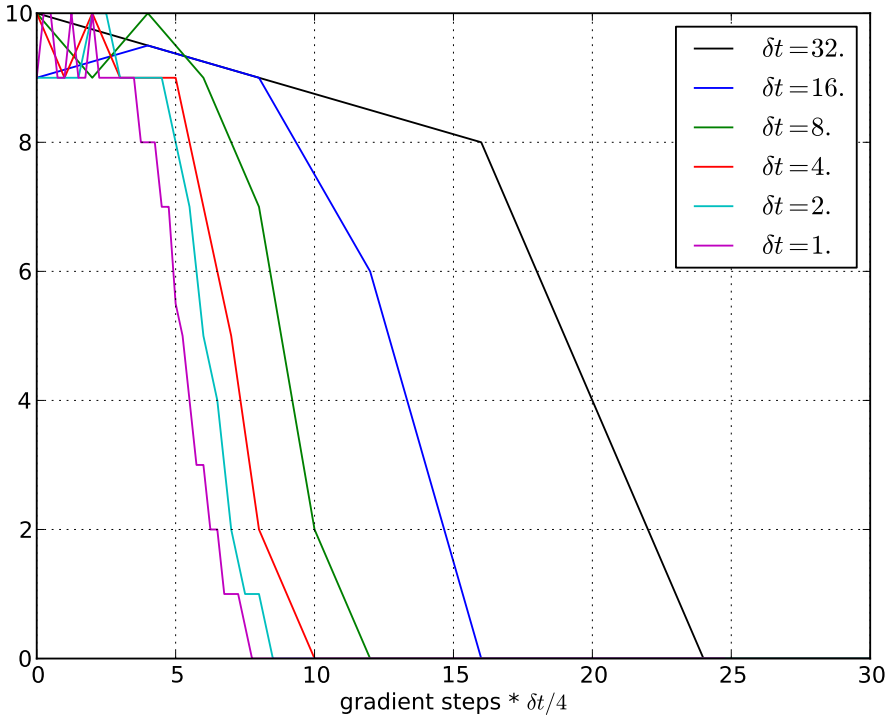

Figure 10: Fitness of sampled points w.r.t. "intrinsic time" during vanilla gradient optimization.

Second, optimization algorithms using distributions can be interpreted as learning a probabilistic model of where the points with good values lie in the search space. With this point of view, P_θ describes *richness of this model*: for instance, restricted Boltzmann machines with h hidden units can describe distributions with up to 2^h modes, whereas the Bernoulli distribution used in PBIL is unimodal. This influences, for instance, the ability to explore several valleys and optimize multimodal functions in a single run.

Natural gradient and parametrization invariance. Central to IGO is the use of natural gradient, which follows from θ-invariance and makes sense on any search space, discrete or continuous.

While the IGO flow is exactly θ-invariant, for any practical implementation of an IGO algorithm, a parametrization choice has to be made. Still, since all IGO algorithms approximate the IGO flow, two parametrizations of IGO will differ less than two parametrizations of another algorithm (such as the vanilla gradient or the smoothed CEM method)—at least if the learning rate δt is not too large.

On the other hand, natural evolution strategies have never strived for θ-invariance: the chosen parametrization (Cholesky, exponential) has been deemed a relevant feature. In the IGO framework, the chosen parametrization becomes more relevant as the step size δt increases.

IGO, maximum likelihood and cross-entropy. The cross-entropy method (CEM) [dBKMR05] can be used to produce optimization algorithms given a family of probability distributions on an arbitrary space, by performing a jump to a maximum likelihood estimate of the parameters.

We have seen (Corollary 16) that the standard CEM is an IGO algorithm in a particular parametrization, with a learning rate δt equal to 1. However, it is well-known, both theoretically and experimentally [BLS07, Han06b, WAS04], that standard CEM loses diversity too fast in many situations. The usual solution [dBKMR05] is to reduce the learning rate (smoothed CEM, (25)), but this breaks the reparametrization invariance.

On the other hand, the IGO flow can be seen as a *maximum likelihood update with infinitesimal learning rate* (Theorem 13). This interpretation allows to define a particular IGO algorithm, the IGO-ML (Definition 14): it performs a maximum likelihood update with an arbitrary learning rate, and keeps the reparametrization invariance. It coincides with CEM when the learning rate is set to 1, but it differs from smoothed CEM by the exchange of the order of argmax and averaging (compare (23) and (25)). We argue that this new algorithm may be a better way to reduce the learning rate and achieve smoothing in CEM.

Standard CEM can lose diversity, yet is a particular case of an IGO algorithm: this illustrates the fact that reasonable values of the learning rate δt depend on the parametrization. We have studied this phenomenon in detail for various Gaussian IGO algorithms (Section 4.2).

Why would a smaller learning rate perform better than a large one in an optimization setting? It might seem more efficient to jump directly to the maximum likelihood estimate of currently known good points, instead of performing a slow gradient ascent towards this maximum.

However, optimization faces a "moving target", contrary to a learning setting in which the example distribution is often stationary. Currently known good points are likely not to indicate the *position* at which the optimum lies, but, rather, the *direction* in which the optimum is to be found.

After an update, the next elite sample points are going to be located somewhere new. So the goal is certainly not to settle down around these currently known points, as a maximum likelihood update does: by design, CEM only tries to reflect status-quo (even for $N = \infty$), whereas IGO tries to move somewhere. When the target moves over time, a progressive gradient ascent is more reasonable than an immediate jump to a temporary optimum, and realizes a kind of time smoothing.

This phenomenon is most clear when the number of sample points is small. Then, a full maximum likelihood update risks losing a lot of diversity; it may even produce a degenerate distribution if the number of sample points is smaller than the number of parameters of the distribution. On the other hand, for smaller δt, the IGO algorithms do, by design, try to maintain diversity by moving as little as possible from the current distribution P_θ in Kullback–Leibler divergence. A full ML update disregards the current distribution and tries to move as close as possible to the elite sample in Kullback–Leibler divergence [dBKMR05], thus realizing maximal diversity loss. This makes sense in a non-iterated scenario but is unsuited for optimization.

Diversity and multiple optima. The IGO framework emphasizes the relation of natural gradient and diversity: we argued that IGO provides minimal diversity change for a given objective function increment. In particular, provided the initial diversity is large, diversity is kept at a maximum. This theoretical relationship has been confirmed experimentally for restricted Boltzmann machines.

On the other hand, using the vanilla gradient does not lead to a balanced distribution between the two optima in our experiments. Using the vanilla gradient introduces hidden arbitrary choices between those points (more exactly between moves in Θ-space). This results in loss of diversity, and might also be detrimental at later stages in the optimization. This may reflect the fact that the Euclidean metric on the space of parameters, implicitly used in the vanilla gradient, becomes less and less meaningful for gradient descent on complex distributions.

IGO and the natural gradient are certainly relevant to the well-known problem of exploration-exploitation balance: as we have seen, arguably the natural gradient realizes the best increment of the objective function with the least possible change of diversity in the population.

More generally, IGO tries to learn a model of where the good points are, representing all good points seen so far rather than focusing only on some good points; this is typical of machine learning, one of the contexts for which the natural gradient was studied. The conceptual relationship of IGO and IGO-like optimization algorithms with machine learning is still to be explored and exploited.

Summary and conclusion

We sum up:

- The information-geometric optimization (IGO) framework derives from invariance principles and allows to build optimization algorithms from any family of distributions on any search space. In some instances (Gaussian distributions on $\mathbb{R}^d$ or Bernoulli distributions on $\{0,1\}^d$) it

recovers versions of known algorithms (CMA-ES or PBIL); in other instances (restricted Boltzmann machine distributions) it produces new, hopefully efficient optimization algorithms.

- The use of a quantile-based, time-dependent transform of the objective function (Equation (3)) provides a rigorous derivation of rank-based update rules currently used in optimization algorithms. Theorem 4 uniquely identifies the infinite-population limit of these update rules.

- The IGO flow is singled out by its equivalent description as an infinitesimal weighted maximal log-likelihood update (Theorem 13). In a particular parametrization and with a step size of 1, it recovers the cross-entropy method (Corollary 16).

- Theoretical arguments suggest that the IGO flow minimizes the change of diversity in the course of optimization. In particular, starting with high diversity and using multimodal distributions may allow simultaneous exploration of multiple optima of the objective function. Preliminary experiments with restricted Boltzmann machines confirm this effect in a simple situation.

Thus, the IGO framework is an attempt to provide sound theoretical foundations to optimization algorithms based on probability distributions. In particular, this viewpoint helps to bridge the gap between continuous and discrete optimization.

The invariance properties, which reduce the number of arbitrary choices, together with the relationship between natural gradient and diversity, may contribute to a theoretical explanation of the good practical performance of those currently used algorithms, such as CMA-ES, which can be interpreted as instantiations of IGO.

We hope that invariance properties will acquire in computer science the importance they have in mathematics, where intrinsic thinking is the first step for abstract linear algebra or differential geometry, and in modern physics, where the notions of invariance w.r.t. the coordinate system and so-called gauge invariance play central roles.

Acknowledgements

The authors would like to thank Michèle Sebag for the acronym and for helpful comments. Y. O. would like to thank Cédric Villani and Bruno Sévennec for helpful discussions on the Fisher metric. A. A. and N. H. would like to acknowledge the Dagstuhl Seminar No 10361 on the Theory of Evolutionary Computation (http://www.dagstuhl.de/10361) for inspiring their work on natural gradients and beyond. This work was partially supported by the ANR-2010-COSI-002 grant (SIMINOLE) of the French National Research Agency.

Appendix: Proofs

Proof of Theorem 4 (Convergence of empirical means and quantiles)

Let us give a more precise statement including the necessary regularity conditions.

Proposition 18. *Let $\theta \in \Theta$. Assume that the derivative $\frac{\partial \ln P_\theta(x)}{\partial \theta}$ exists for P_θ-almost all $x \in X$ and that $\mathbb{E}_{P_\theta} \left| \frac{\partial \ln P_\theta(x)}{\partial \theta} \right|^2 < +\infty$. Assume that the function w is non-decreasing and bounded.*

Let $(x_i)_{i\in\mathbb{N}}$ be a sequence of independent samples of P_θ. Then with probability 1, as $N \to \infty$ we have

$$\frac{1}{N} \sum_{i=1}^{N} \widehat{W}^f(x_i) \frac{\partial \ln P_\theta(x_i)}{\partial \theta} \to \int W_\theta^f(x) \, \frac{\partial \ln P_\theta(x)}{\partial \theta} \, P_\theta(\mathrm{d}x)$$

where

$$\widehat{W}^f(x_i) = w\left(\frac{\mathrm{rk}_N(x_i) + 1/2}{N} \right)$$

with $\mathrm{rk}_N(x_i) = \#\{1 \leqslant j \leqslant N, f(x_j) < f(x_i)\}$. (When there are f-ties in the sample, $W^f(x_i)$ is defined as the average of $w((r+1/2)/N)$ over the possible rankings r of x_i.)

Proof. Let $g : X \to \mathbb{R}$ be any function with $\mathbb{E}_{P_\theta} g^2 < \infty$. We will show that $\frac{1}{N} \sum \widehat{W}^f(x_i) g(x_i) \to \int W_\theta^f(x) g(x) \, P_\theta(\mathrm{d}x)$. Applying this with g equal to the components of $\frac{\partial \ln P_\theta(x)}{\partial \theta}$ will yield the result.

Let us decompose

$$\frac{1}{N} \sum \widehat{W}^f(x_i) g(x_i) = \frac{1}{N} \sum W_\theta^f(x_i) g(x_i) + \frac{1}{N} \sum (\widehat{W}^f(x_i) - W_\theta^f(x_i)) g(x_i).$$

Each summand in the first term involves only one sample x_i (contrary to $\widehat{W}^f(x_i)$ which depends on the whole sample). So by the strong law of large numbers, almost surely $\frac{1}{N} \sum W_\theta^f(x_i) g(x_i)$ converges to $\int W_\theta^f(x) g(x) \, P_\theta(\mathrm{d}x)$. So we have to show that the second term converges to 0 almost surely.

By the Cauchy–Schwarz inequality, we have

$$\left| \frac{1}{N} \sum (\widehat{W}^f(x_i) - W_\theta^f(x_i)) g(x_i) \right|^2 \leqslant \left(\frac{1}{N} \sum (\widehat{W}^f(x_i) - W_\theta^f(x_i))^2 \right) \left(\frac{1}{N} \sum g(x_i)^2 \right)$$

By the strong law of large numbers, the second term $\frac{1}{N} \sum g(x_i)^2$ converges to $E_{P_\theta} g^2$ almost surely. So we have to prove that the first term $\frac{1}{N} \sum (\widehat{W}^f(x_i) - W_\theta^f(x_i))^2$ converges to 0 almost surely.

Since w is bounded by assumption, we can write

$$\begin{aligned} (\widehat{W}^f(x_i) - W_\theta^f(x_i))^2 &\leqslant 2B \left| \widehat{W}^f(x_i) - W_\theta^f(x_i) \right| \\ &= 2B \left| \widehat{W}^f(x_i) - W_\theta^f(x_i) \right|_+ + 2B \left| \widehat{W}^f(x_i) - W_\theta^f(x_i) \right|_- \end{aligned}$$

where B is the bound on $|w|$. We will bound each of these terms.

Let us abbreviate $q_i^- = \Pr_{x' \sim P_\theta}(f(x') < f(x_i))$, $q_i^+ = \Pr_{x' \sim P_\theta}(f(x') \leqslant f(x_i))$, $r_i^- = \#\{j \leqslant N, \, f(x_j) < f(x_i)\}$, $r_i^+ = \#\{j \leqslant N, \, f(x_j) \leqslant f(x_i)\}$.

By definition of $\widehat{W}^f$ we have

$$\widehat{W}^f(x_i) = \frac{1}{r_i^+ - r_i^-} \sum_{k=r_i^-}^{r_i^+ - 1} w((k+1/2)/N)$$

and moreover $W_\theta^f(x_i) = w(q_i^-)$ if $q_i^- = q_i^+$ or $W_\theta^f(x_i) = \frac{1}{q_i^+ - q_i^-} \int_{q_i^-}^{q_i^+} w$ otherwise.

The Glivenko–Cantelli theorem states that $\sup_i \left| q_i^+ - r_i^+/N \right|$ tends to 0 almost surely, and likewise for $\sup_i \left| q_i^- - r_i^-/N \right|$. So let N be large enough so that these errors are bounded by ε.

Since w is non-increasing, we have $w(q_i^-) \leqslant w(r_i^-/N - \varepsilon)$. In the case $q_i^- \neq q_i^+$, we decompose the interval $[q_i^-; q_i^+]$ into $(r_i^+ - r_i^-)$ subintervals. The average of w over each such subinterval is compared to a term in the sum defining $w^N(x_i)$: since w is non-increasing, the average of w over the k^{th} subinterval is at most $w((r_i^- + k)/N - \varepsilon)$. So we get

$$W_\theta^f(x_i) \leqslant \frac{1}{r_i^+ - r_i^-} \sum_{k=r_i^-}^{r_i^+ - 1} w(k/N - \varepsilon)$$

so that

$$W_\theta^f(x_i) - \widehat{W}^f(x_i) \leqslant \frac{1}{r_i^+ - r_i^-} \sum_{k=r_i^-}^{r_i^+ - 1} (w(k/N - \varepsilon) - w((k+1/2)/N)).$$

Let us sum over i, remembering that there are $(r_i^+ - r_i^-)$ values of j for which $f(x_j) = f(x_i)$. Taking the positive part, we get

$$\frac{1}{N} \sum_{i=1}^N \left| W_\theta^f(x_i) - \widehat{W}^f(x_i) \right|_+ \leqslant \frac{1}{N} \sum_{k=0}^{N-1} (w(k/N - \varepsilon) - w((k+1/2)/N)).$$

Since w is non-increasing we have

$$\frac{1}{N} \sum_{k=0}^{N-1} w(k/N - \varepsilon) \leqslant \int_{-\varepsilon - 1/N}^{1 - \varepsilon - 1/N} w$$

and

$$\frac{1}{N} \sum_{k=0}^{N-1} w((k+1/2)/N) \geqslant \int_{1/2N}^{1+1/2N} w$$

(we implicitly extend the range of w so that $w(q) = w(0)$ for $q < 0$). So we have

$$\frac{1}{N} \sum_{i=1}^N \left| W_\theta^f(x_i) - \widehat{W}^f(x_i) \right|_+ \leqslant \int_{-\varepsilon - 1/N}^{1/2N} w - \int_{1 - \varepsilon - 1/N}^{1+1/2N} w \leqslant (2\varepsilon + 3/N)B$$

where B is the bound on $|w|$.

Reasoning symmetrically with $w(k/N + \varepsilon)$ and the inequalities reversed, we get a similar bound for $\frac{1}{N} \sum \left| W_\theta^f(x_i) - \widehat{W}^f(x_i) \right|_-$. This ends the proof. □

Proof of Proposition 5 (Quantile improvement)

Let us use the weight $w(u) = \mathbb{1}_{u \leqslant q}$. Let m be the value of the q-quantile of f under P_{θ^t}. We want to show that the value of the q-quantile of f under $P_{\theta^{t+\delta t}}$ is less than m, unless the gradient vanishes and the IGO flow is stationary.

Let $p_- = \Pr_{x \sim P_{\theta^t}}(f(x) < m)$, $p_m = \Pr_{x \sim P_{\theta^t}}(f(x) = m)$ and $p_+ = \Pr_{x \sim P_{\theta^t}}(f(x) > m)$. By definition of the quantile value we have $p_- + p_m \geqslant q$ and $p_+ + p_m \geqslant 1 - q$. Let us assume that we are in the more complicated case $p_m \neq 0$ (for the case $p_m = 0$, simply remove the corresponding terms).

We have $W_{\theta_t}^f(x) = 1$ if $f(x) < m$, $W_{\theta_t}^f(x) = 0$ if $f(x) > m$ and $W_{\theta_t}^f(x) = \frac{1}{p_m} \int_{p_-}^{p_- + p_m} w(u) \mathrm{d}u = \frac{q - p_-}{p_m}$ if $f(x) = m$.

Using the same notation as above, let $g_t(\theta) = \int W_{\theta^t}^f(x)\, P_\theta(\mathrm{d}x)$. Decomposing this integral on the three sets $f(x) < m$, $f(x) = m$ and $f(x) > m$, we

get that $g_t(\theta) = \Pr_{x\sim P_\theta}(f(x) < m) + \Pr_{x\sim P_\theta}(f(x) = m)\frac{q-p_-}{p_m}$. In particular, $g_t(\theta^t) = q$.

Since we follow a gradient ascent of g_t, for δt small enough we have $g_t(\theta^{t+\delta t}) > g_t(\theta^t)$ unless the gradient vanishes. If the gradient vanishes we have $\theta^{t+\delta t} = \theta^t$ and the quantiles are the same. Otherwise we get $g_t(\theta^{t+\delta t}) > g_t(\theta^t) = q$.

Since $\frac{q-p_-}{p_m} \leqslant \frac{(p_-+p_m)-p_-}{p_m} = 1$, we have $g_t(\theta) \leqslant \Pr_{x\sim P_\theta}(f(x) < m) + \Pr_{x\sim P_\theta}(f(x) = m) = \Pr_{x\sim P_\theta}(f(x) \leqslant m)$.

So $\Pr_{x\sim P_{\theta^{t+\delta t}}}(f(x) \leqslant m) \geqslant g_t(\theta^{t+\delta t}) > q$. This implies that, by definition, the q-quantile value of $P_{\theta^{t+\delta t}}$ is smaller than m.

Proof of Proposition 10 (Speed of the IGO flow)

Lemma 19. *Let X be a centered L^2 random variable with values in $\mathbb{R}^d$ and let A be a real-valued L^2 random variable. Then*

$$\|\mathbb{E}(AX)\| \leqslant \sqrt{\lambda \operatorname{Var} A}$$

where λ is the largest eigenvalue of the covariance matrix of X expressed in an orthonormal basis.

Proof of the lemma. Let v be any vector in $\mathbb{R}^d$; its norm satisfies

$$\|v\| = \sup_{w,\, \|w\|\leqslant 1} (v \cdot w)$$

and in particular

$$\begin{aligned}
\|\mathbb{E}(AX)\| &= \sup_{w,\, \|w\|\leqslant 1} (w \cdot \mathbb{E}(AX)) \\
&= \sup_{w,\, \|w\|\leqslant 1} \mathbb{E}(A\,(w \cdot X)) \\
&= \sup_{w,\, \|w\|\leqslant 1} \mathbb{E}((A - \mathbb{E}A)\,(w \cdot X)) \quad \text{since } (w \cdot X) \text{ is centered} \\
&\leqslant \sup_{w,\, \|w\|\leqslant 1} \sqrt{\operatorname{Var} A}\, \sqrt{\mathbb{E}((w \cdot X)^2)}
\end{aligned}$$

by the Cauchy–Schwarz inequality and using the fact that A is centered.

Now, in an orthonormal basis we have

$$(w \cdot X) = \sum_i w_i X_i$$

so that

$$\begin{aligned}
\mathbb{E}((w \cdot X)^2) &= \mathbb{E}\left((\textstyle\sum_i w_i X_i)(\sum_j w_j X_j)\right) \\
&= \textstyle\sum_i \sum_j \mathbb{E}(w_i X_i w_j X_j) \\
&= \textstyle\sum_i \sum_j w_i w_j \mathbb{E}(X_i X_j) \\
&= \textstyle\sum_i \sum_j w_i w_j C_{ij}
\end{aligned}$$

with C_{ij} the covariance matrix of X. The latter expression is the scalar product $(w \cdot Cw)$. Since C is a symmetric positive-semidefinite matrix, $(w \cdot Cw)$ is at most $\lambda\|w\|^2$ with λ the largest eigenvalue of C. □

For the IGO flow we have $\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = \mathbb{E}_{x\sim P_\theta} W_\theta^f(x) \tilde{\nabla}_\theta \ln P_\theta(x)$.

So applying the lemma, we get that the norm of $\frac{\mathrm{d}\theta}{\mathrm{d}t}$ is at most $\sqrt{\lambda \operatorname{Var}_{x\sim P_\theta} W_\theta^f(x)}$ where λ is the largest eivengalue of the covariance matrix of $\tilde{\nabla}_\theta \ln P_\theta(x)$ (expressed in a coordinate system where the Fisher matrix at the current point θ is the identity).

By construction of the quantiles, we have $\operatorname{Var}_{x\sim P_\theta} W_\theta^f(x) \leqslant \operatorname{Var}_{[0,1]} w$ (with equality unless there are ties). Indeed, for a given x, let $\mathcal{U}$ be a uniform random variable in $[0,1]$ independent from x and define the random variable $Q = q^-(x) + (q^+(x) - q^-(x))\mathcal{U}$. Then Q is uniformly distributed between the upper and lower quantiles $q^+(x)$ and $q^-(x)$ and thus we can rewrite $W_\theta^f(x)$ as $\mathbb{E}(w(Q)|x)$. By the Jensen inequality we have $\operatorname{Var} W_\theta^f(x) = \operatorname{Var} \mathbb{E}(w(Q)|x) \leqslant \operatorname{Var} w(Q)$. In addition when x is taken under P_θ, Q is uniformly distributed in $[0,1]$ and thus $\operatorname{Var} w(Q) = \operatorname{Var}_{[0,1]} w$, i.e. $\operatorname{Var}_{x\sim P_\theta} W_\theta^f(x) \leqslant \operatorname{Var}_{[0,1]} w$.

Besides, consider the tangent space in Θ-space at point θ^t, and let us choose an orthonormal basis in this tangent space for the Fisher metric. Then, in this basis we have $\tilde{\nabla}_i \ln P_\theta(x) = \partial_i \ln P_\theta(x)$. So the covariance matrix of $\tilde{\nabla} \ln P_\theta(x)$ is $\mathbb{E}_{x\sim P_\theta}(\partial_i \ln P_\theta(x) \partial_j \ln P_\theta(x))$, which is equal to the Fisher matrix by definition. So this covariance matrix is the identity, whose largest eigenvalue is 1.

Proof of Proposition 12 (Noisy IGO)

On the one hand, let P_θ be a family of distributions on X. The IGO algorithm (13) applied to a random function $f(x) = \tilde{f}(x, \omega)$ where ω is a random variable uniformly distributed in $[0,1]$ reads

$$\theta^{t+\delta t} = \theta^t + \delta t \sum_{i=1}^N \hat{w}_i \widetilde{\nabla}_\theta \ln P_\theta(x_i) \tag{56}$$

where $x_i \sim P_\theta$ and $\hat{w}_i$ is according to (12) where ranking is applied to the values $\tilde{f}(x_i, \omega_i)$, with ω_i uniform variables in $[0,1]$ independent from x_i and from each other.

On the other hand, for the IGO algorithm using $P_\theta \otimes U_{[0,1]}$ and applied to the deterministic function $\tilde{f}$, $\hat{w}_i$ is computed using the ranking according to the $\tilde{f}$ values of the sampled points $\tilde{x}_i = (x_i, \omega_i)$, and thus coincides with the one in (56).

Besides,

$$\widetilde{\nabla}_\theta \ln P_{\theta\otimes U_{[0,1]}}(\tilde{x}_i) = \widetilde{\nabla}_\theta \ln P_\theta(x_i) + \underbrace{\widetilde{\nabla}_\theta \ln U_{[0,1]}(\omega_i)}_{=0}$$

and thus the IGO algorithm update on space $X \times [0,1]$, using the family of distributions $\tilde{P}_\theta = P_\theta \otimes U_{[0,1]}$, applied to the deterministic function $\tilde{f}$, coincides with (56).

Proof of Theorem 13 (Natural gradient as ML with infinitesimal weights)

We begin with a calculus lemma (proof omitted).

Lemma 20. *Let f be real-valued function on a finite-dimensional vector space E equipped with a definite positive quadratic form $\|\cdot\|^2$. Assume f is smooth and has at most quadratic growth at infinity. Then, for any $x \in E$, we have*

$$\nabla f(x) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \arg\max_h \left\{ f(x+h) - \frac{1}{2\varepsilon} \|h\|^2 \right\}$$

where ∇ is the gradient associated with the norm $\|\cdot\|$. Equivalently,

$$\arg\max_y \left\{ f(y) - \frac{1}{2\varepsilon}\|y-x\|^2 \right\} = x + \varepsilon \nabla f(x) + O(\varepsilon^2)$$

when $\varepsilon \to 0$.

We are now ready to prove Theorem 13. Let W be a function of x, and fix some θ_0 in Θ.

We need some regularity assumptions: we assume that the parameter space Θ is non-degenerate (no two points $\theta \in \Theta$ define the same probability distribution) and proper (the map $P_\theta \mapsto \theta$ is continuous). We also assume that the map $\theta \mapsto P_\theta$ is smooth enough, so that $\int \log P_\theta(x)\, W(x)\, P_{\theta_0}(\mathrm{d}x)$ is a smooth function of θ. (These are restrictions on θ-regularity: this does not mean that W has to be continuous as a function of x.)

The two statements of Theorem 13 using a sum and an integral have similar proofs, so we only include the first. For $\varepsilon > 0$, let θ be the solution of

$$\theta = \arg\max \left\{ (1-\varepsilon) \int \log P_\theta(x)\, P_{\theta_0}(\mathrm{d}x) + \varepsilon \int \log P_\theta(x)\, W(x)\, P_{\theta_0}(\mathrm{d}x) \right\}.$$

Then we have

$$\begin{aligned}
\theta &= \arg\max \left\{ \int \log P_\theta(x)\, P_{\theta_0}(\mathrm{d}x) + \varepsilon \int \log P_\theta(x)\, (W(x)-1)\, P_{\theta_0}(\mathrm{d}x) \right\} \\
&= \arg\max \left\{ \int \log P_\theta(x)\, P_{\theta_0}(\mathrm{d}x) - \int \log P_{\theta_0}(x)\, P_{\theta_0}(\mathrm{d}x) + \varepsilon \int \log P_\theta(x)\, (W(x)-1)\, P_{\theta_0}(\mathrm{d}x) \right\}
\end{aligned}$$

(because the added term does not depend on θ)

$$\begin{aligned}
&= \arg\max \left\{ -\mathrm{KL}(P_{\theta_0} \,\|\, P_\theta) + \varepsilon \int \log P_\theta(x)\, (W(x)-1)\, P_{\theta_0}(\mathrm{d}x) \right\} \\
&= \arg\max \left\{ -\frac{1}{\varepsilon}\mathrm{KL}(P_{\theta_0} \,\|\, P_\theta) + \int \log P_\theta(x)\, (W(x)-1)\, P_{\theta_0}(\mathrm{d}x) \right\}
\end{aligned}$$

When $\varepsilon \to 0$, the first term exceeds the second one if $\mathrm{KL}(P_{\theta_0} \,\|\, P_\theta)$ is too large (because W is bounded), and so $\mathrm{KL}(P_{\theta_0} \,\|\, P_\theta)$ tends to 0. So we can assume that θ is close to θ_0.

When $\theta = \theta_0 + \delta\theta$ is close to θ_0, we have

$$\mathrm{KL}(P_{\theta_0} \,\|\, P_\theta) = \frac{1}{2} \sum I_{ij}(\theta_0)\, \delta\theta_i\, \delta\theta_j + O(\delta\theta^3)$$

with $I_{ij}(\theta_0)$ the Fisher matrix at θ_0. (This actually holds both for $\mathrm{KL}(P_{\theta_0} \,\|\, P_\theta)$ and $\mathrm{KL}(P_\theta \,\|\, P_{\theta_0})$.)

Thus, we can apply the lemma above using the Fisher metric $\sum I_{ij}(\theta_0)\, \delta\theta_i\, \delta\theta_j$, and working on a small neighborhood of θ_0 in θ-space (which can be identified with $\mathbb{R}^{\dim\Theta}$). The lemma states that the argmax above is attained at

$$\theta = \theta_0 + \varepsilon \widetilde{\nabla}_\theta \int \log P_\theta(x)\, (W(x)-1)\, P_{\theta_0}(\mathrm{d}x)$$

up to $O(\varepsilon^2)$, with $\widetilde{\nabla}$ the natural gradient.

Finally, the gradient cancels the constant -1 because $\int (\widetilde{\nabla} \log P_\theta)\, P_{\theta_0} = 0$ at $\theta = \theta_0$. This proves Theorem 13.

Proof of Theorem 15 (IGO, CEM and IGO-ML)

Let P_θ be a family of probability distributions of the form

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp\left(\sum \theta_i T_i(x)\right) \, H(\mathrm{d}x)$$

where $T_1, \dots, T_k$ is a finite family of functions on X and $H(\mathrm{d}x)$ is some reference measure on X. We assume that the family of functions $(T_i)_i$ together with the constant function $T_0(x) = 1$, are linearly independent. This prevents redundant parametrizations where two values of θ describe the same distribution; this also ensures that the Fisher matrix $\mathrm{Cov}(T_i, T_j)$ is invertible.

The IGO update (14) in the parametrization $\bar{T}_i$ is a sum of terms of the form

$$\widetilde{\nabla}_{\bar{T}_i} \ln P(x).$$

So we will compute the natural gradient $\widetilde{\nabla}_{\bar{T}_i}$ in those coordinates. We first need some general results about the Fisher metric for exponential families.

The next proposition gives the expression of the Fisher scalar product between two tangent vectors δP and $\delta' P$ of a statistical manifold of exponential distributions. It is one way to express the duality between the coordinates θ_i and $\bar{T}_i$ (compare [AN00, (3.30) and Section 3.5]).

Proposition 21. *Let $\delta\theta_i$ and $\delta'\theta_i$ be two small variations of the parameters θ_i. Let $\delta P(x)$ and $\delta' P(x)$ be the resulting variations of the probability distribution P, and $\delta\bar{T}_i$ and $\delta'\bar{T}_i$ the resulting variations of $\bar{T}_i$. Then the scalar product, in Fisher information metric, between the tangent vectors δP and $\delta' P$, is*

$$\langle \delta P, \delta' P \rangle = \sum_i \delta\theta_i \, \delta'\bar{T}_i = \sum_i \delta'\theta_i \, \delta\bar{T}_i.$$

Proof. By definition of the Fisher metric:

$$\begin{aligned}
\langle \delta P, \delta' P \rangle &= \sum_{i,j} I_{ij} \, \delta\theta_i \, \delta'\theta_j \\
&= \sum_{i,j} \delta\theta_i \, \delta'\theta_j \int_x \frac{\partial \ln P(x)}{\partial \theta_i} \frac{\partial \ln P(x)}{\partial \theta_j} P(x) \\
&= \int_x \sum_i \frac{\partial \ln P(x)}{\partial \theta_i} \delta\theta_i \sum_j \frac{\partial \ln P(x)}{\partial \theta_j} \delta'\theta_j \, P(x) \\
&= \int_x \sum_i \frac{\partial \ln P(x)}{\partial \theta_i} \, \delta\theta_i \, \delta'(\ln P(x)) \, P(x) \\
&= \int_x \sum_i \frac{\partial \ln P(x)}{\partial \theta_i} \, \delta\theta_i \, \delta' P(x) \\
&= \int_x \sum_i (T_i(x) - \bar{T}_i) \delta\theta_i \, \delta' P(x) \qquad \text{by (16)} \\
&= \sum_i \delta\theta_i \left(\int_x T_i(x) \, \delta' P(x) \right) - \sum_i \delta\theta_i \bar{T}_i \int_x \delta' P(x) \\
&= \sum_i \delta\theta_i \, \delta'\bar{T}_i
\end{aligned}$$

because $\int_x \delta' P(x) = 0$ since the total mass of P is 1, and $\int_x T_i(x) \, \delta' P(x) = \delta'\bar{T}_i$ by definition of $\bar{T}_i$. ∎

Proposition 22. *Let f be a function on the statistical manifold of an exponential family as above. Then the components of the natural gradient w.r.t. the expectation parameters are given by the vanilla gradient w.r.t. the natural parameters:*

$$\widetilde{\nabla}_{\bar{T}_i} f = \frac{\partial f}{\partial \theta_i}$$

and conversely

$$\widetilde{\nabla}_{\theta_i} f = \frac{\partial f}{\partial \bar{T}_i}.$$

(Beware this does *not* mean that the gradient ascent in any of those parametrizations is the vanilla gradient ascent.)

We could not find a reference for this result, though we think it known.

Proof. By definition, the natural gradient $\widetilde{\nabla} f$ of a function f is the unique tangent vector δP such that that, for any other tangent vector $\delta' P$, we have

$$\delta' f = \langle \delta P, \delta' P \rangle$$

with $\langle \cdot, \cdot \rangle$ the scalar product associated with the Fisher metric. We want to compute this natural gradient in coordinates $\bar{T}_i$, so we are interested in the variations $\delta \bar{T}_i$ associated with δP.

By Proposition 21, the scalar product above is

$$\langle \delta P, \delta' P \rangle = \sum \delta \bar{T}_i \, \delta' \theta_i$$

where $\delta \bar{T}_i$ is the variation of $\bar{T}_i$ associated with δP, and $\delta' \theta_i$ the variation of θ_i associated with $\delta' P$.

On the other hand we have $\delta' f = \sum_i \frac{\partial f}{\partial \theta_i} \delta' \theta_i$. So we must have

$$\sum_i \delta \bar{T}_i \, \delta' \theta_i = \sum_i \frac{\partial f}{\partial \theta_i} \delta' \theta_i$$

for any $\delta' P$, which leads to

$$\delta \bar{T}_i = \frac{\partial f}{\partial \theta_i}$$

as needed. The converse relation is proved *mutatis mutandis*. □

Back to the proof of Theorem 15. We can now compute the desired terms:

$$\widetilde{\nabla}_{\bar{T}_i} \ln P(x) = \frac{\partial \ln P(x)}{\partial \theta_i} = T_i(x) - \bar{T}_i$$

by (16). This proves the first statement (27) in Theorem 15 about the form of the IGO update in these parameters.

The other statements follow easily from this together with the additional fact (26) that, for any set of weights a_i with $\sum a_i = 1$, the value $T^* = \sum_i a(i) T(x_i)$ is the maximum likelihood estimate of $\sum_i a(i) \log P(x_i)$.

References

[AHS85] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, *A learning algorithm for Boltzmann machines*, Cognitive Science **9** (1985), no. 1, 147–169.

[Ama98] Shun-Ichi Amari, *Natural gradient works efficiently in learning*, Neural Comput. **10** (1998), 251–276.

[AN00] Shun-ichi Amari and Hiroshi Nagaoka, *Methods of information geometry*, Translations of Mathematical Monographs, vol. 191, American Mathematical Society, Providence, RI, 2000, Translated from the 1993 Japanese original by Daishi Harada.

[ANOK10] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi, *Bidirectional relation between CMA evolution strategies and natural evolution strategies*, Proceedings of Parallel Problem Solving from Nature - PPSN XI, Lecture Notes in Computer Science, vol. 6238, Springer, 2010, pp. 154–163.

[AO08] Ravi P. Agarwal and Donal O'Regan, *An introduction to ordinary differential equations*, Springer, 2008.

[Arn06] D.V. Arnold, *Weighted multirecombination evolution strategies*, Theoretical computer science **361** (2006), no. 1, 18–37.

[Bal94] Shumeet Baluja, *Population based incremental learning: A method for integrating genetic search based function optimization and competitve learning*, Tech. Report CMU-CS-94-163, Carnegie Mellon Report, 1994.

[BC95] Shumeet Baluja and Rich Caruana, *Removing the genetics from the standard genetic algorithm*, Proceedings of ICML'95, 1995, pp. 38–46.

[Ber00] A. Berny, *Selection and reinforcement learning for combinatorial optimization*, Parallel Problem Solving from Nature PPSN VI (Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Merelo, and Hans-Paul Schwefel, eds.), Lecture Notes in Computer Science, vol. 1917, Springer Berlin Heidelberg, 2000, pp. 601–610.

[Ber02] Arnaud Berny, *Boltzmann machine for population-based incremental learning*, ECAI, 2002, pp. 198–202.

[Bey01] H.-G. Beyer, *The theory of evolution strategies*, Natural Computing Series, Springer-Verlag, 2001.

[BLS07] J. Branke, C. Lode, and J.L. Shapiro, *Addressing sampling errors and diversity loss in umda*, Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM, 2007, pp. 508–515.

[BS02] H.G. Beyer and H.P. Schwefel, *Evolution strategies—a comprehensive introduction*, Natural computing **1** (2002), no. 1, 3–52.

[CT06] Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, second ed., Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2006.

[dBKMR05] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein, *A tutorial on the cross-entropy method*, Annals OR **134** (2005), no. 1, 19–67.

[Gha04] Zoubin Ghahramani, *Unsupervised learning*, Advanced Lectures on Machine Learning (Olivier Bousquet, Ulrike von

Luxburg, and Gunnar Rätsch, eds.), Lecture Notes in Computer Science, vol. 3176, Springer Berlin / Heidelberg, 2004, pp. 72–112.

[GSS+10] Tobias Glasmachers, Tom Schaul, Yi Sun, Daan Wierstra, and Jürgen Schmidhuber, *Exponential natural evolution strategies*, GECCO, 2010, pp. 393–400.

[Han06a] N. Hansen, *An analysis of mutative σ-self-adaptation on linear fitness functions*, Evolutionary Computation **14** (2006), no. 3, 255–275.

[Han06b] ———, *The CMA evolution strategy: a comparing review*, Towards a new evolutionary computation. Advances on estimation of distribution algorithms (J.A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, eds.), Springer, 2006, pp. 75–102.

[Han09] ———, *Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed*, Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers (New York, NY, USA), GECCO '09, ACM, 2009, pp. 2389–2396.

[Hin02] G.E. Hinton., *Training products of experts by minimizing contrastive divergence*, Neural Computation **14** (2002), 1771–1800.

[HJ61] R. Hooke and T.A. Jeeves, *"direct search" solution of numerical and statistical problems*, Journal of the ACM **8** (1961), 212–229.

[HK04] N. Hansen and S. Kern, *Evaluating the CMA evolution strategy on multimodal test functions*, Parallel Problem Solving from Nature PPSN VIII (X. Yao et al., eds.), LNCS, vol. 3242, Springer, 2004, pp. 282–291.

[HMK03] N. Hansen, S.D. Müller, and P. Koumoutsakos, *Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)*, Evolutionary Computation **11** (2003), no. 1, 1–18.

[HO96] N. Hansen and A. Ostermeier, *Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation*, ICEC96, IEEE Press, 1996, pp. 312–317.

[HO01] Nikolaus Hansen and Andreas Ostermeier, *Completely derandomized self-adaptation in evolution strategies*, Evolutionary Computation **9** (2001), no. 2, 159–195.

[JA06] G.A. Jastrebski and D.V. Arnold, *Improving evolution strategies through active covariance matrix adaptation*, Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, IEEE, 2006, pp. 2814–2821.

[Jef46] Harold Jeffreys, *An invariant form for the prior probability in estimation problems*, Proc. Roy. Soc. London. Ser. A. **186** (1946), 453–461.

[Kul97] Solomon Kullback, *Information theory and statistics*, Dover Publications Inc., Mineola, NY, 1997, Reprint of the second (1968) edition.

[LL02] P. Larranaga and J.A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*, Springer Netherlands, 2002.

[LRMB07] Nicolas Le Roux, Pierre-Antoine Manzagol, and Yoshua Bengio, *Topmoumoute online natural gradient algorithm*, NIPS, 2007.

[MMS08] Luigi Malagò, Matteo Matteucci, and Bernardo Dal Seno, *An information geometry perspective on estimation of distribution algorithms: boundary analysis*, GECCO (Companion), 2008, pp. 2081–2088.

[NM65] John Ashworth Nelder and R Mead, *A simplex method for function minimization*, The Computer Journal (1965), 308–313.

[PGL02] M. Pelikan, D.E. Goldberg, and F.G. Lobo, *A survey of optimization by building and using probabilistic models*, Computational optimization and applications **21** (2002), no. 1, 5–20.

[Rao45] Calyampudi Radhakrishna Rao, *Information and the accuracy attainable in the estimation of statistical parameters*, Bull. Calcutta Math. Soc. **37** (1945), 81–91.

[Rec73] I. Rechenberg, *Evolutionstrategie: Optimierung technisher systeme nach prinzipien des biologischen evolution*, Frommann-Holzboog Verlag, Stuttgart, 1973.

[RK04] R.Y. Rubinstein and D.P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, monte-carlo simulation, and machine learning*, Springer-Verlag New York Inc, 2004.

[Rub99] Reuven Rubinstein, *The cross-entropy method for combinatorial and continuous optimization*, Methodology and Computing in Applied Probability **1** (1999), 127–190, 10.1023/A:1010091220143.

[SA90] F. Silva and L. Almeida, *Acceleration techniques for the backpropagation algorithm*, Neural Networks (1990), 110–119.

[Sch92] Laurent Schwartz, *Analyse. II*, Collection Enseignement des Sciences [Collection: The Teaching of Science], vol. 43, Hermann, Paris, 1992, Calcul différentiel et équations différentielles, With the collaboration of K. Zizi.

[Sch95] H.-P. Schwefel, *Evolution and optimum seeking*, Sixth-generation computer technology series, John Wiley & Sons, Inc. New York, NY, USA, 1995.

[SHI09] Thorsten Suttorp, Nikolaus Hansen, and Christian Igel, *Efficient covariance matrix update for variable metric evolution strategies*, Machine Learning **75** (2009), no. 2, 167–197.

[Smo86] P. Smolensky, *Information processing in dynamical systems: foundations of harmony theory*, Parallel Distributed Processing (D. Rumelhart and J. McClelland, eds.), vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 194–281.

[SWSS09] Yi Sun, Daan Wierstra, Tom Schaul, and Juergen Schmidhuber, *Efficient natural evolution strategies*, Proceedings of the 11th Annual conference on Genetic and evolutionary computation (New York, NY, USA), GECCO '09, ACM, 2009, pp. 539–546.

[Tho00] H. Thorisson, *Coupling, stationarity, and regeneration*, Springer, 2000.

[Tor97] Virginia Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on optimization **7** (1997), no. 1, 1–25.

[WAS04] Michael Wagner, Anne Auger, and Marc Schoenauer, *EEDA : A New Robust Estimation of Distribution Algorithms*, Research Report RR-5190, INRIA, 2004.

[Whi89] D. Whitley, *The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best*, Proceedings of the third international conference on Genetic algorithms, 1989, pp. 116–121.

[WSPS08] Daan Wierstra, Tom Schaul, Jan Peters, and Jürgen Schmidhuber, *Natural evolution strategies*, IEEE Congress on Evolutionary Computation, 2008, pp. 3381–3387.