Cross-Entropy Method

Dirk P. Kroese, School of Mathematics and Physics, The University of Queensland, Brisbane 4072, Australia, kroese@maths.uq.edu.au.

Reuven Y. Rubinstein, Faculty of Industrial Engineering and Management, Technion, Haifa, Israel, ierrr01@ie.technion.ac.il

Izack Cohen, Faculty of Industrial Engineering and Management, Technion, Haifa, Israel, izik68@tx.technion.ac.il.

Sergey Porotsky, A.L.D. Ltd., 52 Manachem Begin Road, Tel-Aviv 67137, Israel, Sergey.Porotsky@ald.co.il.

Thomas Taimre, School of Mathematics and Physics, The University of Queensland, Brisbane 4072, Australia, ttaimre@maths.uq.edu.au.

Abstract

The cross-entropy method is a powerful heuristic tool for solving difficult estimation and optimization problems, based on Kullback–Leibler (or cross-entropy) minimization.

1 Introduction

The *cross-entropy (CE) method* is a versatile Monte Carlo technique introduced by Rubinstein (1999; 2001), extending earlier work on variance minimization (Rubinstein 1997). A tutorial on the CE method is given in de Boer et al. (2005). A comprehensive treatment can be found in Rubinstein and Kroese (2004); see also Rubinstein and Kroese (2007; Chapter 8). The CE method homepage is `www.cemethod.org`.

The CE method can be applied to two types of problems:

1. **Estimation**: Estimate $\ell = \mathbb{E}[H(\mathbf{X})]$, where $\mathbf{X}$ is a random object taking values in some set $\mathscr{X}$ and H is a function on $\mathscr{X}$. An important special case is the estimation of a probability $\ell = \mathbb{P}(S(\mathbf{X}) \geqslant \gamma)$, where S is another function on $\mathscr{X}$.

2. **Optimization**: Optimize (that is, maximize or minimize) $S(\mathbf{x})$ over all $\mathbf{x} \in \mathscr{X}$, where S is some objective function on $\mathscr{X}$.

In the estimation setting, the CE method can be viewed as an adaptive *importance sampling* procedure that uses the *cross-entropy* or *Kullback–Leibler divergence* as a measure of closeness between two sampling distributions. In the optimization setting, the optimization problem is first translated into a rare-event estimation problem, and then the CE method for estimation is used as an adaptive algorithm to locate the optimum.

2 Estimation

Consider the estimation of

$$\ell = \mathbb{E}_f[H(\mathbf{X})] = \int H(\mathbf{x})\, f(\mathbf{x})\, \mathrm{d}\mathbf{x}\ , \tag{1}$$

where H is a real-valued function and f is the probability density function (pdf) of the random vector $\mathbf{X}$. It is assumed, for simplicity, that $\mathbf{X}$ is a continuous random variable. For the discrete case, replace the integral in (1) by a sum. Let g be another pdf — which must be non-zero for every $\mathbf{x}$ for which $H(\mathbf{x})\, f(\mathbf{x}) \neq 0$. Using the pdf g, ℓ can be represented as

$$\ell = \int H(\mathbf{x})\, \frac{f(\mathbf{x})}{g(\mathbf{x})}\, g(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}_g \left[H(\mathbf{X})\, \frac{f(\mathbf{X})}{g(\mathbf{X})} \right]\ , \tag{2}$$

where the subscript g indicates that the expectation is taken with respect to g rather than f. Consequently, if $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are independent random vectors with pdf g, written as $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\text{iid}} g$, then

$$\widehat{\ell} = \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k)\, \frac{f(\mathbf{X}_k)}{g(\mathbf{X}_k)} \tag{3}$$

is an unbiased estimator of ℓ: a so-called *importance sampling estimator*. The optimal importance sampling pdf, that is, the pdf g^* for which the variance of $\widehat{\ell}$ is minimal, is proportional to $|H|\, f$ (see, e.g., Rubinstein and Kroese (2007; Page 132)), but is in general difficult to evaluate. The idea of the CE method is to choose the importance sampling pdf g in a specified class of pdfs such that the Kullback–Leibler divergence between the optimal importance sampling pdf g^* and g is minimal. The Kullback–Leibler divergence between two pdfs g and h is given by

$$\begin{aligned} \mathcal{D}(g, h) &= \mathbb{E}_g \left[\ln \frac{g(\mathbf{X})}{h(\mathbf{X})} \right] = \int g(\mathbf{x}) \ln \frac{g(\mathbf{x})}{h(\mathbf{x})}\, \mathrm{d}\mathbf{x} \\ &= \int g(\mathbf{x}) \ln g(\mathbf{x})\, \mathrm{d}\mathbf{x} - \int g(\mathbf{x}) \ln h(\mathbf{x})\, \mathrm{d}\mathbf{x}\ . \end{aligned} \tag{4}$$

In most cases of interest the function H is non-negative, and the "nominal" pdf f is parameterized by a finite-dimensional vector $\mathbf{u}$; that is, $f(\mathbf{x}) = f(\mathbf{x}; \mathbf{u})$. It is then customary to choose the importance sampling pdf g in the *same* family of pdfs; thus, $g(\mathbf{x}) = f(\mathbf{x}; \mathbf{v})$ for some *reference parameter* $\mathbf{v}$. The CE minimization procedure then reduces to finding an optimal reference parameter vector, $\mathbf{v}^*$ say, by cross-entropy minimization:

$$\begin{aligned} \mathbf{v}^* &= \operatorname*{argmin}_{\mathbf{v}} \mathcal{D}(g^*, f(\cdot; \mathbf{v})) \\ &= \operatorname*{argmax}_{\mathbf{v}} \int H(\mathbf{x}) f(\mathbf{x}; \mathbf{u}) \ln f(\mathbf{x}; \mathbf{v})\, \mathrm{d}\mathbf{x} \\ &= \operatorname*{argmax}_{\mathbf{v}} \mathbb{E}_{\mathbf{u}} H(\mathbf{x}) \ln f(\mathbf{X}; \mathbf{v}) \\ &= \operatorname*{argmax}_{\mathbf{v}} \mathbb{E}_{\mathbf{w}} H(\mathbf{x}) \ln f(\mathbf{X}; \mathbf{v}) \frac{f(\mathbf{X}; \mathbf{u})}{f(\mathbf{X}; \mathbf{w})}\ , \end{aligned} \tag{5}$$

where $\mathbf{w}$ is any reference parameter. This $\mathbf{v}^*$ can be estimated via the stochastic counterpart of (5):

$$\widehat{\mathbf{v}} = \underset{\mathbf{v}}{\operatorname{argmax}} \frac{1}{N} \sum_{k=1}^{N} H(\mathbf{X}_k) \frac{f(\mathbf{X}_k; \mathbf{u})}{f(\mathbf{X}_k; \mathbf{w})} \ln f(\mathbf{X}_k; \mathbf{v}) \,, \tag{6}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\text{iid}} f(\cdot; \mathbf{w})$. The optimal parameter $\widehat{\mathbf{v}}$ in (6) can often be obtained in explicit form, in particular when the class of sampling distributions forms an *exponential family*; see, for example, Rubinstein and Kroese (2007; Pages 319–320). Indeed, analytical updating formulas can be found whenever explicit expressions for the *maximal likelihood estimators* of the parameters can be found, cf. de Boer et al. (2005; Page 36).

Example: Exponential Random Variables

Consider the case where $\mathbf{X}_1 = (X_1, \ldots, X_n)$ is a vector of independent exponential random variables with expectations $u_1, \ldots, u_n$. Let $\mathbf{u} = (u_1, \ldots, u_n)$ and let $\mathbf{v} = (v_1, \ldots, v_n)$ be the reference parameter of the importance sampling pdf $f(\mathbf{x}; \mathbf{v})$, given by

$$f(\mathbf{x}; \mathbf{v}) = \prod_{i=1}^{n} \frac{\mathrm{e}^{-x_i/v_i}}{v_i} \,.$$

Hence, under this importance sampling pdf, $X_1, \ldots, X_n$ are again independent and exponentially distributed, but now with expectations $v_1, \ldots, v_n$. Writing $H_k = H(\mathbf{X}_k)$ and the *likelihood ratio* $W_k = f(\mathbf{X}_k; \mathbf{u})/f(\mathbf{X}_k; \mathbf{w})$ in (6), the optimal parameter $\widehat{\mathbf{v}}$ is found by maximizing

$$\sum_{i=1}^{n} \sum_{k=1}^{N} H_k W_k \ln f(\mathbf{X}_k; \mathbf{u}) = \sum_{i=1}^{n} \sum_{k=1}^{N} H_k W_k \left(\frac{-X_{ki}}{v_i} - \ln v_i \right) \,, \tag{7}$$

where X_{ki} is the i-th component of $\mathbf{X}_k$. This maximum can be found by differentiating and equating to zero the righthand side of (7) for each v_i, resulting in the equations

$$\sum_{k=1}^{N} H_k W_k \left(\frac{X_{ki}}{v_i^2} - \frac{1}{v_i} \right) = 0 \,, \quad i = 1, \ldots, n \,,$$

from which it follows that

$$\widehat{v}_i = \frac{\sum_{k=1}^{N} H_k W_k X_{ki}}{\sum_{k=1}^{N} H_k W_k}, \quad i = 1, \ldots, n \,. \tag{8}$$

Often $\ell = \mathbb{P}(S(\mathbf{X}) \geqslant \gamma)$ for some function S and *level* γ, in which case $H(\mathbf{x})$ takes the form of an *indicator function*: $H(\mathbf{x}) = \mathrm{I}_{\{S(\mathbf{X}) \geqslant \gamma\}}$; that is, $H(\mathbf{x}) = 1$ if $S(\mathbf{x}) \geqslant \gamma$, and 0 otherwise. A complication in solving (6) occurs when ℓ is a *rare-event probability*; that is, a very small probability (say less than 10^{-4}). Then, for moderate sample size N most or all of the values $H(\mathbf{X}_k)$ in (6) are

zero, and the maximization problem becomes useless. In that case a *multi-level* CE procedure is used, where a sequence of reference parameters and levels is constructed with the goal that the former converges to $\mathbf{v}^*$ and the latter to γ. This leads to the following algorithm; see, e.g., Rubinstein and Kroese (2007; Page 238).

Algorithm 2.1 (CE Algorithm for Rare-Event Estimation)

1. *Define* $\widehat{\mathbf{v}}_0 = \mathbf{u}$. *Let* $N^{\mathrm{e}} = \lceil \varrho N \rceil$. *Set* $t = 1$ *(iteration counter).*

2. *Generate* $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\mathrm{iid}} f(\cdot; \widehat{\mathbf{v}}_{t-1})$. *Calculate* $S_i = S(\mathbf{X}_i)$ *for all* i, *and order these from smallest to largest:* $S_{(1)} \leqslant \ldots \leqslant S_{(N)}$. *Let* $\widehat{\gamma}_t$ *be the sample* $(1-\varrho)$*-quantile of performances; that is,* $\widehat{\gamma}_t = S_{(N-N^{\mathrm{e}}+1)}$. *If* $\widehat{\gamma}_t > \gamma$, *reset* $\widehat{\gamma}_t$ *to* γ.

3. *Use the* **same** *sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *to solve the stochastic program* (6), *with* $\mathbf{w} = \widehat{\mathbf{v}}_{t-1}$. *Denote the solution by* $\widehat{\mathbf{v}}_t$.

4. *If* $\widehat{\gamma}_t < \gamma$, *set* $t = t + 1$ *and reiterate from Step 2; otherwise, proceed with Step 5.*

5. *Let* T *be the final iteration counter. Generate* $\mathbf{X}_1, \ldots, \mathbf{X}_{N_1} \sim_{\mathrm{iid}} f(\cdot; \widehat{\mathbf{v}}_T)$ *and estimate* ℓ *via importance sampling, as in* (3).

Apart from specifying the family of sampling pdfs, the sample sizes N and N_1, and the rarity parameter ϱ (typically between 0.01 and 0.1), the algorithm is completely self-tuning. The sample size N for determining a good reference parameter can usually be chosen much smaller than the sample size N_1 for the final importance sampling estimation, say $N = 1000$ versus $N_1 = 100{,}000$. Under certain technical conditions the deterministic version of Algorithm 2.1 is guaranteed to terminate (reach level γ) provided that ϱ is chosen small enough; see Section 3.5 of Rubinstein and Kroese (2004).

Example: Rare-Event Probability Estimation

A *stochastic activity network* is a frequently used tool in project management to schedule concurrent activities. Each arc corresponds to an activity, and is weighted by the duration of that activity. The maximal project duration corresponds to the length of the longest path in the graph. Figure 1 shows a stochastic activity network with eight activities. Suppose the durations of the activities are independent exponential random variables $X_1, \ldots, X_8$, each with mean 1.

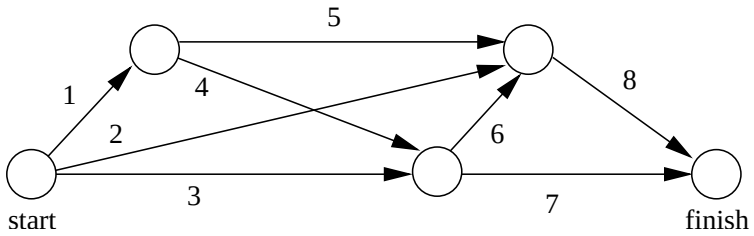

Figure 1: A stochastic activity network.

Let $S(\mathbf{X})$ denote length of the longest path in the graph; that is,

$$S(\mathbf{X}) = \max\{X_1 + X_4 + X_6 + X_8,\ X_1 + X_4 + X_7,\ X_1 + X_5 + X_8,\\ X_2 + X_8,\ X_3 + X_6 + X_8,\ X_3 + X_7\}\ .$$

Suppose the objective is to estimate the rare-event probability $\mathbb{P}(S(\mathbf{X}) \geqslant 20)$ using importance sampling where the random vector $\mathbf{X} = (X_1, \ldots, X_8)$ has independent exponentially distributed components with mean vector $\mathbf{v} = (v_1, \ldots, v_{10})$. Note that the nominal pdf is obtained by setting $v_i = 1$ for all i. At the t-th iteration of the multilevel CE Algorithm 2.1, the solution to (6) with $H(\mathbf{X}) = \mathrm{I}_{\{S(\mathbf{X}) \geqslant \widehat{\gamma}_t\}}$ is, using (8), given by

$$\widehat{v}_{t,i} = \frac{\sum_{k=1}^{N} \mathrm{I}_{\{S(\mathbf{X}_k) \geqslant \widehat{\gamma}_t\}} W_k X_{ki}}{\sum_{k=1}^{N} \mathrm{I}_{\{S(\mathbf{X}_k) \geqslant \widehat{\gamma}_t\}} W_k}\ , \tag{9}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\text{iid}} f(\cdot; \widehat{\mathbf{v}}_{t-1})$, $W_k = f(\mathbf{X}_k; \mathbf{u})/f(\mathbf{X}_k; \widehat{\mathbf{v}}_{t-1})$, and X_{ki} is the i-th element of $\mathbf{X}_k$.

Table 1 lists the successive estimates for the optimal importance sampling parameters obtained from the multilevel CE algorithm, using $N = 10^5$ and $\varrho = 0.1$.

Table 1: Convergence of the sequence $\{(\widehat{\gamma}_t, \widehat{\mathbf{v}}_t)\}$.

t	$\widehat{\gamma}_t$	$\widehat{\mathbf{v}}_t$							
0	–	1	1	1	1	1	1	1	1
1	7.32	1.93	1.12	1.39	1.83	1.32	1.81	1.37	1.96
2	12.01	3.33	1.09	1.58	2.98	1.50	2.95	1.58	3.32
3	20	5.03	1.00	1.88	4.63	1.51	4.73	1.47	5.14

The last step in Algorithm 2.1 gives an estimate of $4.15 \cdot 10^{-6}$ with an estimated relative error of 1%, using a sample size of $N_1 = 10^6$. A typical crude Monte Carlo estimate (that is, taking $\mathbf{v} = \mathbf{u} = (1, 1, \ldots, 1)$) using the same sample size is $3 \cdot 10^{-6}$, with an estimated relative error of 60%, and is therefore of little use.

For large-size activity networks the accurate estimation of the optimal parameters via (9) runs into problems due to the degeneracy behavior of the likelihood ratio; cf. Rubinstein and Kroese (2007; Page 133). For such systems it is recommended to estimate the optimal CE parameters by drawing samples directly from g^*, e.g., via Markov chain Monte Carlo; see Chan (2010).

3 Optimization

Let $\mathscr{X}$ be an arbitrary set of *states* and let S be a real-valued performance function on $\mathscr{X}$. Suppose the goal is to find the maximum of S over $\mathscr{X}$, and the corresponding maximizer $\mathbf{x}^*$ (assuming, for simplicity, that there is only one). Denote the maximum by γ^*, so that

$$S(\mathbf{x}^*) = \gamma^* = \max_{\mathbf{x}\in\mathscr{X}} S(\mathbf{x}) \; . \tag{10}$$

Associate with the above problem the estimation of the probability $\ell = \mathbb{P}(S(\mathbf{X}) \geqslant \gamma)$, where $\mathbf{X}$ has some probability density $f(\mathbf{x};\mathbf{u})$ on $\mathscr{X}$ (for example corresponding to the uniform distribution on $\mathscr{X}$) and γ is some level. Thus, for optimization problems randomness is purposely introduced in order to make the model stochastic, as in the estimation setting. If γ is chosen close to the unknown γ^*, then ℓ is typically a rare-event probability, and the CE approach of Section 2 can be used to find an importance sampling distribution close to the theoretically optimal importance sampling density, which concentrates all its mass on the point $\mathbf{x}^*$. Sampling from such a distribution thus produces optimal or near-optimal states. Note that the final level $\gamma = \gamma^*$ is generally not known in advance, in contrast to the rare-event simulation setting. The CE method for optimization produces a sequence of levels $\{\widehat{\gamma}_t\}$ and reference parameters $\{\widehat{\mathbf{v}}_t\}$ such that the former tends to the optimal γ^* and the latter to the optimal reference vector $\mathbf{v}^*$ corresponding to the point mass at $\mathbf{x}^*$; see, e.g., (Rubinstein and Kroese 2007; Page 251).

Algorithm 3.1 (CE Algorithm for Optimization)

1. *Choose an initial parameter vector* $\widehat{\mathbf{v}}_0$. *Let* $N^{\mathrm{e}} = \lceil \varrho N \rceil$. *Set* $t = 1$ *(level counter).*

2. *Generate* $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim_{\mathrm{iid}} f(\cdot; \widehat{\mathbf{v}}_{t-1})$. *Calculate the performances* $S(\mathbf{X}_i)$ *for all* i, *and order them from smallest to largest:* $S_{(1)} \leqslant \ldots \leqslant S_{(N)}$. *Let* $\widehat{\gamma}_t$ *be the sample* $(1-\varrho)$*-quantile of performances; that is,* $\widehat{\gamma}_t = S_{(N-N^{\mathrm{e}}+1)}$.

3. *Use the* **same** *sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *and solve the stochastic program*

$$\max_{\mathbf{v}} \frac{1}{N} \sum_{k=1}^{N} I_{\{S(X_k)\geqslant\widehat{\gamma}_t\}} \ln f(\mathbf{X}_k; \mathbf{v}) \; . \tag{11}$$

   Denote the solution by $\widehat{\mathbf{v}}_t$.

4. *If some stopping criterion is met, stop; otherwise, set* $t = t+1$, *and return to Step 2.*

To run the algorithm, one needs to provide the class of sampling pdfs, the initial vector $\widehat{\mathbf{v}}_0$, the sample size N, the rarity parameter ϱ, and the stopping criterion. Any CE algorithm for optimization involves thus the following two main iterative phases:

1. **Generate** a random sample of objects in the search space $\mathscr{X}$ (trajectories, vectors, etc.) according to a specified probability distribution.

2. **Update** the parameters of that distribution, based on the N^{e} best performing samples (the so-called *elite samples*), using CE minimization.

Note that Step 5 of Algorithm 2.1 is missing in Algorithm 3.1. Another main difference between the two algorithms is that the likelihood ratio term $f(\mathbf{X}_k; \mathbf{u})/f(\mathbf{X}_k; \widehat{\mathbf{v}}_{t-1})$ in (6) is missing in (11).

Often a smoothed updating rule is used, in which the parameter vector $\widehat{\mathbf{v}}_t$ is taken as

$$\widehat{\mathbf{v}}_t = \alpha\, \widetilde{\mathbf{v}}_t + (1-\alpha)\, \widehat{\mathbf{v}}_{t-1}, \tag{12}$$

where $\widetilde{\mathbf{v}}_t$ is the solution to (11) and $0 \leqslant \alpha \leqslant 1$ is a smoothing parameter. Many other modifications can be found in Kroese et al. (2006), Rubinstein and Kroese (2004), and Rubinstein and Kroese (2007). When there are two or more optimal solutions the CE algorithm typically "fluctuates" between the solutions before focusing in on one of the solutions. The effect that smoothing has on convergence is discussed in detail in Costa et al. (2007). In particular, it is shown that with appropriate smoothing the CE method converges and finds the optimal solution with probability arbitrarily close to 1. Necessary conditions and sufficient conditions under which the optimal solution is generated eventually with probability 1 are also given. Other convergence results, including a proof of convergence along the lines of the convergence proof for simulated annealing can be found in Margolin (2005). The CE method is also effective for solving *noisy* optimization problems, for example when the objective function value is obtained via simulation. Typical examples may be found in Alon et al. (2005) and Cohen et al. (2007).

3.1 Combinatorial Optimization

When the state space $\mathscr{X}$ is finite, the optimization problem (10) is often referred to as a *discrete* or *combinatorial optimization* problem. For example, $\mathscr{X}$ could be the space of combinatorial objects such as binary vectors, trees, paths through graphs, permutations, etc. To apply the CE method, one needs to first specify a convenient parameterized random mechanism to generate objects $\mathbf{X}$ in $\mathscr{X}$. An important example is where $\mathbf{X} = (X_1, \ldots, X_n)$ has independent components such that $X_i = j$ with probability $p_{ij}, i = 1, \ldots, n, j = 1, \ldots, m$. In that case, the CE updating rule (see de Boer et al. (2005; Page 56)) at the t-th iteration is

$$\widehat{p}_{t,ij} = \frac{\sum_{k=1}^{N} \mathrm{I}_{\{S(\mathbf{X}_k) \geqslant \widehat{\gamma}_t\}} \mathrm{I}_{\{X_{ki}=j\}}}{\sum_{k=1}^{N} \mathrm{I}_{\{S(\mathbf{X}_k) \geqslant \widehat{\gamma}_t\}}}\ , i = 1, \ldots, n,\ j = 1, \ldots, m\ , \tag{13}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are independent copies of $\mathbf{X} \sim \{\widehat{p}_{t-1,ij}\}$ and X_{ki} is the i-th element of $\mathbf{X}_k$. Thus, the updated probability $\widehat{p}_{t,ij}$ is simply the number of elite samples for which the i-th component is equal to j, divided by the total number of elite samples.

A possible stopping rule for combinatorial optimization problems is to stop when the overall best objective value does not change over a number of iterations. Alternatively, one could stop when the sampling distribution has "degenerated" enough. For example, when in (13) the $\{\widehat{p}_{t,ij}\}$ differ less than some small $\varepsilon > 0$ from the $\{\widehat{p}_{t-1,i,j}\}$.

Example: Max-Cut Problem

The max-cut problem in a graph can be formulated as follows. Given a weighted graph $G(V, E)$ with node set $V = \{1, \ldots, n\}$ and edge set E, partition the nodes of the graph into two subsets V_1 and V_2 such that the sum of the (nonnegative) weights of the edges going from one subset to the other is maximized. Let $C = (C(i, j))$ be the matrix of weights. The objective is to maximize

$$\sum_{(i,j)\in V_1\times V_2} (C(i,j) + C(j,i)) \tag{14}$$

over all *cuts* $\{V_1, V_2\}$. Such a cut can be conveniently represented by a binary *cut vector* $\mathbf{x} = (1, x_2, \ldots, x_n)$, where $x_i = 1$ indicates that $i \in V_1$. Let $\mathscr{X}$ be the set of cut vectors and let $S(\mathbf{x})$ be the value of the cut represented by $\mathbf{x}$, as given in (14).

To maximize S via the CE method one can generate the random cut vectors by drawing each component (except the first one, which is set to 1) independently from a Bernoulli distribution, that is, $\mathbf{X} = (1, X_2, \ldots, X_n) \sim \mathsf{Ber}(\mathbf{p})$, where $\mathbf{p} = (1, p_2, \ldots, p_n)$. Given an elite sample set $\mathscr{E}$, with size N^{e}, the updating formula (13) is then:

$$\widehat{p}_{t,i} = \frac{\sum_{\mathbf{X}\in\mathscr{E}} X_i}{N^{\mathrm{e}}}, \quad i = 2, \ldots, n\,. \tag{15}$$

That is, the updated success probability for the i-th component is the mean of the i-th components of the vectors in the elite set.

Figure 2 illustrates the evolution of the Bernoulli parameters for a max-cut problem from de Boer et al. (2005) of dimension $n = 400$, for which the optimal solution is given by $\mathbf{x}^* = (1, \ldots, 1, 0, \ldots, 0)$.

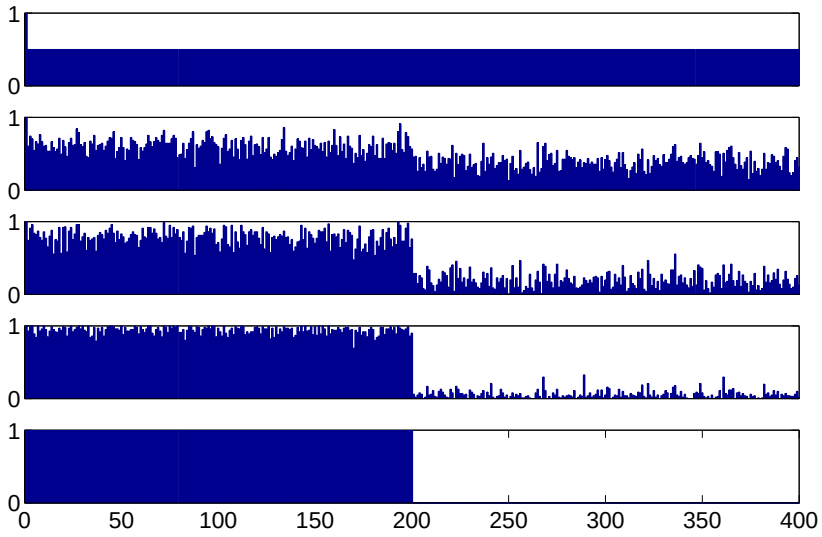

Figure 2: Sequence of reference vectors for a synthetic max-cut problem with 400 nodes. Iterations 0, 5, 10, 15, and 20 are displayed.

3.2 Continuous Optimization

When the state space is continuous, in particular when $\mathscr{X} = \mathbb{R}^n$, the optimization problem is often referred to as a *continuous optimization* problem. The sampling distribution on $\mathbb{R}^n$ can be quite arbitrary, and does not need to be related to the function that is being optimized. The generation of a random vector $\mathbf{X} = (X_1, \ldots, X_n) \in \mathbb{R}^n$ is most easily performed by drawing the coordinates independently from some 2-parameter distribution. In most applications a normal (Gaussian) distribution is employed for each component. Thus, the sampling distribution for $\mathbf{X}$ is characterized by a vector of means $\boldsymbol{\mu}$ and a vector of standard deviations $\boldsymbol{\sigma}$. At each iteration of the CE algorithm these parameter vectors are updated simply as the vectors of sample means and sample standard deviations of the elements in the elite set; see, for example, Kroese et al. (2006).

Algorithm 3.2 (CE for Continuous Optimization: Normal Updating)

1. ***Initialize:*** *Choose* $\widehat{\boldsymbol{\mu}}_0$ *and* $\widehat{\boldsymbol{\sigma}}_0^2$. *Set* $t = 1$.
2. ***Draw:*** *Generate a random sample* $\mathbf{X}_1, \ldots, \mathbf{X}_N$ *from the* $\mathsf{N}(\widehat{\boldsymbol{\mu}}_{t-1}, \widehat{\boldsymbol{\sigma}}_{t-1}^2)$ *distribution.*
3. ***Select:*** *Let* $\mathcal{I}$ *be the indices of the* N^{e} *best performing (=elite) samples.* ***Update:*** *For all* $j = 1, \ldots, n$ *let*

$$\widetilde{\mu}_{t,j} = \sum_{i \in \mathcal{I}} X_{ij} / N^{\mathrm{e}} \tag{16}$$

and

$$\widetilde{\sigma}_{t,j}^2 = \sum_{i \in \mathcal{I}} (X_{ij} - \widetilde{\mu}_{t,j})^2 / N^{\mathrm{e}}. \tag{17}$$

4. ***Smooth:***

$$\widehat{\boldsymbol{\mu}}_t = \alpha \widetilde{\boldsymbol{\mu}}_t + (1-\alpha)\widehat{\boldsymbol{\mu}}_{t-1}, \quad \widehat{\boldsymbol{\sigma}}_t = \alpha \widetilde{\boldsymbol{\sigma}}_t + (1-\alpha)\widehat{\boldsymbol{\sigma}}_{t-1} \tag{18}$$

5. *If* $\max_j\{\widehat{\sigma}_{t,j}\} < \varepsilon$ **stop** *and return* $\boldsymbol{\mu}_t$ *as an approximate solution. Otherwise, increase* t *by* 1 *and return to Step 2.*

For *constrained* continuous optimization problems, where the samples are restricted to a subset $\mathscr{X} \subset \mathbb{R}^n$, it is often possible to replace the normal sampling with sampling from a truncated normal distribution while retaining the updating formulas (16)–(17). An alternative is to use a beta distribution. Instead of returning $\widehat{\boldsymbol{\mu}}_t$ as the final solution, one often returns the overall best solution generated by the algorithm.

Smoothing, as in Step 4, is often crucial to prevent premature shrinking of the sampling distribution. Instead of using a single smoothing factor, it is often useful to use separate smoothing factors for $\widehat{\boldsymbol{\mu}}_t$ and $\widehat{\boldsymbol{\sigma}}_t$. An alternative is to use *dynamic* smoothing for $\widehat{\boldsymbol{\sigma}}_t$:

$$\alpha_t = \beta - \beta \left(1 - \frac{1}{t}\right)^q, \tag{19}$$

where q is an integer (typically between 5 and 10) and β is a smoothing constant (typically between 0.8 and 0.99). Another approach is to *inject* extra variance into the sampling distribution, for example by increasing the components of $\boldsymbol{\sigma}$, once the distribution has degenerated; see Botev and Kroese (2004). Finally, significant speed up can be achieved by using a *parallel* implementation of CE; see, for example, Evans et al. (2007).

Example: Parameter Estimation for Differential Equations

Consider the *FitzHugh–Nagumo* differential equations:

$$\begin{aligned} \frac{\mathrm{d}V_t}{\mathrm{d}t} &= c\left(V_t - \frac{V_t^3}{3} + R_t\right), \\ \frac{\mathrm{d}R_t}{\mathrm{d}t} &= -\frac{1}{c}(V_t - a + bR_t), \end{aligned} \tag{20}$$

which model the behavior of certain types of neurons (Nagumo et al. 1962). Ramsay et al. (2007) consider estimating the parameters a, b, and c from noisy observations of (V_t) by using a generalized smoothing approach. The simulated data in Figure 3 correspond to the values of V_t obtained from (20) at times $0, 0.05, \ldots, 20.0$, adding Gaussian noise with standard deviation 0.5. The true parameter values are $a = 0.2$, $b = 0.2$, and $c = 3$. The initial conditions are $V_0 = -1$ and $R_0 = 1$.

Estimation of the parameters via the CE method can be established by minimizing the least-squares performance

$$S(\mathbf{x}) = \sum_{i=0}^{400} \left(y_i - V_{0.05i}(\mathbf{x})\right)^2 \ ,$$

where $\{y_i\}$ are the simulated data, $\mathbf{x} = (a, b, c, V_0, R_0)$, and $V_t(\mathbf{x})$ is the solution to (20) for parameter vector $\mathbf{x}$. Algorithm 3.2 was implemented with $\widehat{\boldsymbol{\mu}}_0 = (0, 0, 5, 0, 0), \widehat{\boldsymbol{\sigma}}_0 = (1, 1, 1, 1, 1)$, $N = 100$, $N^{\mathrm{e}} = 10$, and $\varepsilon = 0.001$. Constant smoothing parameters $\alpha_1 = 0.9$ and $\alpha_2 = 0.5$ were used for the $\{\widehat{\boldsymbol{\mu}}_t\}$ and the $\{\widehat{\boldsymbol{\sigma}}_t\}$, respectively. The following solution was found (notice that the initial condition was assumed to be unknown): $\widehat{a} = 0.19, \widehat{b} = 0.21, \widehat{c} = 3.00, \widehat{V}_0 = -1.02$, and $\widehat{R}_0 = 1.02$. The smooth curve in Figure 3 gives the corresponding estimated curve, which is practically indistinguishable from the true one.

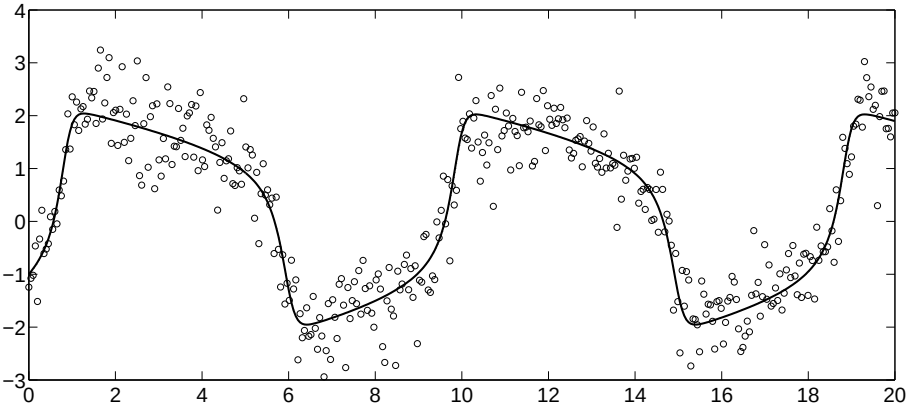

Figure 3: Simulated data for the FitzHugh–Nagumo model and a fitted curve obtained via the CE method.

References

G. Alon, D. P. Kroese, T. Raviv, and R. Y. Rubinstein. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, 134(1):137–151, 2005.

Z. I. Botev and D. P. Kroese. Global likelihood optimization via the cross-entropy method with an application to mixture models. In *Proceedings of the 36th Winter Simulation Conference*, pages 529–535, Washington, D.C., 2004.

J. C. C. Chan. *Advanced Monte Carlo Methods with Applications in Finance*. PhD thesis, University of Queensland, 2010.

I. Cohen, B. Golany, and A. Shtub. Resource allocation in stochastic, finite-capacity, multi-project systems through the cross entropy methodology. *Journal of Scheduling*, 10(1):181–193, 2007.

A. Costa, J. Owen, and D. P. Kroese. Convergence properties of the cross-entropy method for discrete optimization. *Operations Research Letters*, 35 (5):573–580, 2007.

P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

G. E. Evans, J. M. Keith, and D. P. Kroese. Parallel cross-entropy optimization. In *Proceedings of the 2007 Winter Simulation Conference*, pages 2196–2202, Washington, D.C., 2007.

D. P. Kroese, S. Porotsky, and R. Y. Rubinstein. The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, 8(3):383–407, 2006.

L. Margolin. On the convergence of the cross-entropy method. *Annals of Operations Research*, 134(1):201–214, 2005.

J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, October 1962.

J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: A generalized smoothing approach. *Journal of the Royal Statistical Society, Series B*, 69(5):741–796, 2007.

R. Y. Rubinstein. Combinatorial optimization, cross-entropy, ants and rare events. In S. Uryasev and P. M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, pages 304–358, Dordrecht, 2001. Kluwer.

R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

R. Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999.

R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, second edition, 2007.

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, 2004.