The Cross-Entropy Method Optimizes for Quantiles

Sergiu Goschin SGOSCHIN@CS.RUTGERS.EDU
Ari Weinstein AWEINST@CS.RUTGERS.EDU
Rutgers University, Piscataway, NJ 08854 USA

Michael L. Littman MLITTMAN@CS.BROWN.EDU
Brown University, Providence, RI 02912 USA

Abstract

Cross-entropy optimization (CE) has proven to be a powerful tool for search in control environments. In the basic scheme, a distribution over proposed solutions is repeatedly adapted by evaluating a sample of solutions and refocusing the distribution on a percentage of those with the highest scores. We show that, in the kind of noisy evaluation environments that are common in decision-making domains, this percentage-based refocusing does not optimize the expected utility of solutions, but instead a quantile metric. We provide a variant of CE (Proportional CE) that effectively optimizes the expected value. We show using variants of established noisy environments that Proportional CE can be used in place of CE and can improve solution quality.

1. Introduction

Originally designed as a technique for the simulation of rare events in networks (Rubinstein, 1996), the cross-entropy (CE) method was later adapted to the task of optimization by casting optimal events as the rare events of interest (Rubinstein, 1999). As an optimization strategy, CE has been successfully applied in a variety of tasks, such as doing policy search in reinforcement learning (RL) (Mannor et al., 2003) or performing supervised classification (Mannor et al., 2005). Although its theoretical properties are understood only in limited settings (Margolin, 2005; Costa et al., 2007), CE's empirical success in a wide range of applications has put it among the state of the art methods for global optimization.

Most prior work applies CE directly with no modifications when optimizing a stochastic evaluation function. Usually, the underlying assumption is that the noise is "well behaved" (input-independent Gaussian noise for example) and that CE performs well on average. The main focus of this paper will be to show that this assumption of well behaved noise is commonly violated, and leads to solutions with poor expected value (sometimes worse than chance). Formally, it will be shown that this failure is due to the fact that CE optimizes for quantiles instead of expectation. Therefore, in domains where the ordering determined by the expectations is different than the one determined by the targeted quantile, performance is poor. We will propose a simple, alternative algorithm that accomplishes the correct task. Our formal proof will be based on a model used to establish similar properties for selection rules in simple genetic algorithms (Vose, 1998; Goschin et al., 2011).

Empirically, we show that noise distributions with the above property occur naturally in a variety of commonly studied stochastic optimization problems. In particular, we will use domains from operations research (Inventory Control), policy search in RL (Tetris) and games (Blackjack) to demonstrate this claim. We show that even in small Markov Decision Processes, the noise distributions over the returns of policies can have a wide variety of shapes, supports, and variances.

2. Related Work

As mentioned, CE has had significant empirical success in a number of settings, among them buffer allocation (Alon et al., 2005), scheduling, and vehicle routing. More references and applications are described in the standard CE tutorial (Boer et al., 2005) or in

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

the detailed monographs on the topic (Rubinstein & Kroese, 2004; Chang et al., 2007). The first paper to apply the CE method in the context of RL for policy search was Mannor et al. (2005). The idea of using CE to search in a parameterized policy space was subsequently used to obtain results that were orders of magnitude better than previous approaches in a challenging RL domain—Tetris (Szita & Lörincz, 2006; Szita & Szepesvári, 2010a), which we will also address here. More recent papers compare CE with standard RL techniques (Kalyanakrishnan & Stone, 2009) and establish interesting connections with other policy-search algorithms like CMA-ES and PI^2, generalizing several design choices made in standard CE (Stulp & Sigaud, 2012).

On the theoretical side, several initial proofs (Rubinstein & Kroese, 2004; Margolin, 2005) established convergence properties of modified versions of CE under certain assumptions. In a more recent paper, Costa et al. (2007) prove the asymptotic convergence of the standard version of CE for discrete optimization. An underlying assumption of the results above is that there is no noise in the function to be optimized. Previous empirical evidence (Rubinstein & Kroese, 2004) suggests that standard CE behaves well in noisy settings at least for certain domains. To the best of our knowledge, the only theoretical result that discusses the convergence of a modified CE algorithm in a noisy setting (Chang et al., 2007), proposes sufficient, but impractical modifications to CE to address arbitrary noise, in addition to adding extra parameters to the algorithm. (See Section 3 for more details.)

3. Algorithms

The key idea of CE is to maintain a distribution over a space of inputs and update that distribution iteratively so that its support focuses only on the optimal solutions.

The Cross-Entropy Method

For a fixed iteration t, a distribution $\mathcal{D}_t$ over an input space $\mathcal{X}$, and query access to a (possibly noisy) function $F : \mathcal{X} \to \mathbb{R}$ to be optimized, CE proceeds in three phases that are executed iteratively. In the **first phase**, it samples a set of N inputs $x_i \sim \mathcal{D}_t, i \in [1, N]$ and evaluates them. In the **second phase**, it ranks the inputs according to their values $F(x_i)$ and selects a size ρN top (or "elite") subset (for some $\rho \in (0\%, 100\%)$) or, equivalently, the inputs with higher evaluations than the $1-\rho$ sample quantile. Finally, in the **third step**, it uses the elite subset to set the new parameters for $\mathcal{D}_{t+1}$, most commonly by determining the maximum likelihood estimators for the elite set. CE is executed either for a fixed number of iterations or until the distribution is concentrated on a small subregion of the input space. In the RL setting, solutions are encodings of policies, and F executes the policy in the domain, yielding the return of that trajectory.

The algorithm is parameterized by the choice of N, ρ, the parameters of the initial distribution $\mathcal{D}_0$ over the input space and the family of distributions $\mathcal{D}_t$ (which includes the initial distribution). The distributions $\mathcal{D}_t, t \geq 0$ are usually part of the natural exponential family and the standard choice is the normal distribution (for continuous inputs) or the Bernoulli (or multinomial) distribution (for discrete inputs). To instantiate the algorithm for a particular distribution, one needs to specify the update rule for the third stage. In general the rule is determined by solving a stochastic program (for the general version the reader is referred to algorithm 2.1 in Boer et al. (2005)). We will give an example of update rules for the case of $\mathcal{D}_t$ being multi-variate Bernoulli distributions over $\{0, 1\}^n$ (for the normal distribution see Stulp & Sigaud (2012) for example). In this case, $\mathcal{D}_t$ are thus parameterized by a vector of elements $p_i^t \in [0, 1], t \geq 0, i \in \{1, ..., n\}$ (where p_i^t is the parameter for the ith Bernoulli distribution at generation t).

In the first stage of generation t, CE samples N Bernoulli vectors $\mathbf{x}_j$ and evaluates them. In the second stage CE computes the "elite" or the $1-\rho$ sample quantile F_t^ρ based on the evaluations and in the third stage it updates p_i's using the formula: $p_i^{t+1} = \frac{\sum_{j=1}^N I[F(\mathbf{x}_j) \geq F_t^\rho]\ I[x_{j,i}=1]}{\sum_{j=1}^N I[F(\mathbf{x}_j) \geq F_t^\rho]}$, where I is the identity function and $x_{j,i}$ is the ith component of the jth vector. So each component p_i^{t+1} is set to reflect the ratio of 1 values of the bits at position i among the elite sample.

A number of techniques have been used to address various practical observations regarding the behavior of the algorithm. One of the most common problems is that the distribution sometimes prematurely converges to a single point. This is because in practice, the variance of the "elite" population is much smaller than the population at large, leading to a decrease in the variance of $\mathcal{D}_t$. One solution is to artificially maintain the variance of the population high, which was one of the key ideas leading to the empirical success of CE in Tetris (Szita & Lörincz, 2006). Another common technique is to smooth the updates of the parameters of the distribution over generations.

It is important to note that the algorithm is often applied "as is" in settings where the evaluation of the function F is corrupted by an arbitrary noise process.

The key point of the paper is that the algorithm optimizes a quantile measure that, in certain situations of practical interest, is different from optimizing for the expected value of the function.

The m-Cross-Entropy Method (mCE)

An intuitive way to mitigate the impact that the optimization for quantiles has on the expectation of the solution is to take the mean of m samples for each queried input (for example this was applied by Mannor et al. (2003) to address the noise in evaluations). Then, the standard version of CE can be applied considering the value of an individual $\hat{F}(x) = \frac{\sum_{i=1}^{m} F_i(x)}{m}$ (where $F_i(x)$ are i.i.d samples from $F(x)$). By the central limit theorem, as m increases, the noise distribution for each evaluated input will become concentrated around its mean, thus eliminating (in the limit) the problem of inconsistent orderings for the mean and any quantile values.

One obvious problem with mCE is the need to choose a reasonable value for m when not enough information is available about the noise distributions. If m is too small, the undesired phenomenon can still occur. If, on the other hand, m is too large, for a fixed number of function evaluations per generation (thus counting the repeated evaluations of the same point), two problems can occur. On one hand, it is possible that not enough inputs are evaluated for mCE to succeed in finding the optimal (or a reasonable) solution. On the other hand, improvements in the expected value from resampling come at the cost of increased variance in the quality of the final population, as issues of early convergence are exacerbated when the set of sampled points shrinks. As has been observed in early work on evolutionary methods (Fitzpatrick & Grefenstette, 1988), the tradeoff between m, N, and the total number of generations is a complex one with no universally "right" answer. We will discuss an example in Section 5.4 to illustrate these tradeoffs in the context of mCE.

A different (but related to mCE) approach to modifying CE for optimizing in stochastic environments was proposed by Chang et al. (2007) under the name of "Model Reference Adaptive Search 2" (MRAS_2). In addition to other modifications, the algorithm requires the designer to specify a rule m_k for the number of times each input is evaluated at each generation k. For the algorithm to converge (under certain assumptions), m_k is required to increase with k and the authors suggest $m_k = \Omega(c^k)$ (for some $c > 1$) or $m_k = \Omega(k)$ as possible rules for several classes of noise. (See Section 4.2.3 in Chang et al. (2007).) While the above rules are sufficient for convergence in certain scenarios, they lead to impractical algorithms for all but the simplest domains, in addition to adding the need to specify the correct m_k sequence.

Proportional Cross-Entropy

We will now propose a variant of the standard CE method that seeks the input that optimizes the expected value of the evaluation function. In addition to seeking high expected value solutions, the method has the additional benefit of not requiring the parameter ρ. The main modification is a change to the second phase of the CE algorithm: Instead of selecting a subset of the samples from $\mathcal{D}_t$, the algorithm weights each input according to its value (normalizing with respect to the difference between the minimum and the maximum value to address negative evaluations). Then, in the third stage, it sets the parameters of distribution $\mathcal{D}_{t+1}$ according to these weighted inputs.

Concretely, for the same case of the multivariate Bernoulli distributions and using the same notation as for CE, the weights for the sampled inputs are $w_j = \frac{F(\mathbf{x}_j)-m}{M-m}$, where $m = \min_{j=1}^{N}\{F(x_j)\}, M = \max_{j=1}^{N}\{F(x_j)\}$ (the case of $M = m$ can be handled by setting all weights to be equal). Then, in the final stage, the new parameters for the distributions are set according to the equations: $p_i^{t+1} = \frac{\sum_{j=1}^{N} w_j I[x_{j,i}=1]}{\sum_{j=1}^{N} w_j}$.

The idea of modifying the definition of what an "elite" set represents is not new. In CMA-ES, the mechanism for deciding the relative importance of the top ρN samples can be chosen by the algorithm designer (Hansen & Ostermeier, 2001), but it is still the case that the samples outside the "elite" set have no influence in shaping the distribution for the next iteration. In PI^2 (Stulp & Sigaud, 2012), an exponential decay scheme weights all inputs according to their evaluation. Thus, the algorithm we propose can be viewed as being an instantiation of a general template for designing CE-like algorithms. Our contribution is to link the "elite" set selection mechanism (phase two of the CE algorithm) to the optimization objective of the algorithm. To simplify comparison and analysis, we keep all the other design choices unchanged and focus only on comparing the standard algorithm with Proportional CE.

A Simple Example

To illustrate the main point of the paper, we will describe a simple optimization example. We ran an experiment with a constrained version of the video game Tetris using a setup similar to Szita & Szepesvári (2010a). We used a 10×8 board, the feature set from Bertsekas & Ioffe (1996), and allowed only the "S", "Z," and "I" tetrominoes to appear with probabilities of $45\%, 45\%$ and 10%. The score bonuses for clear-

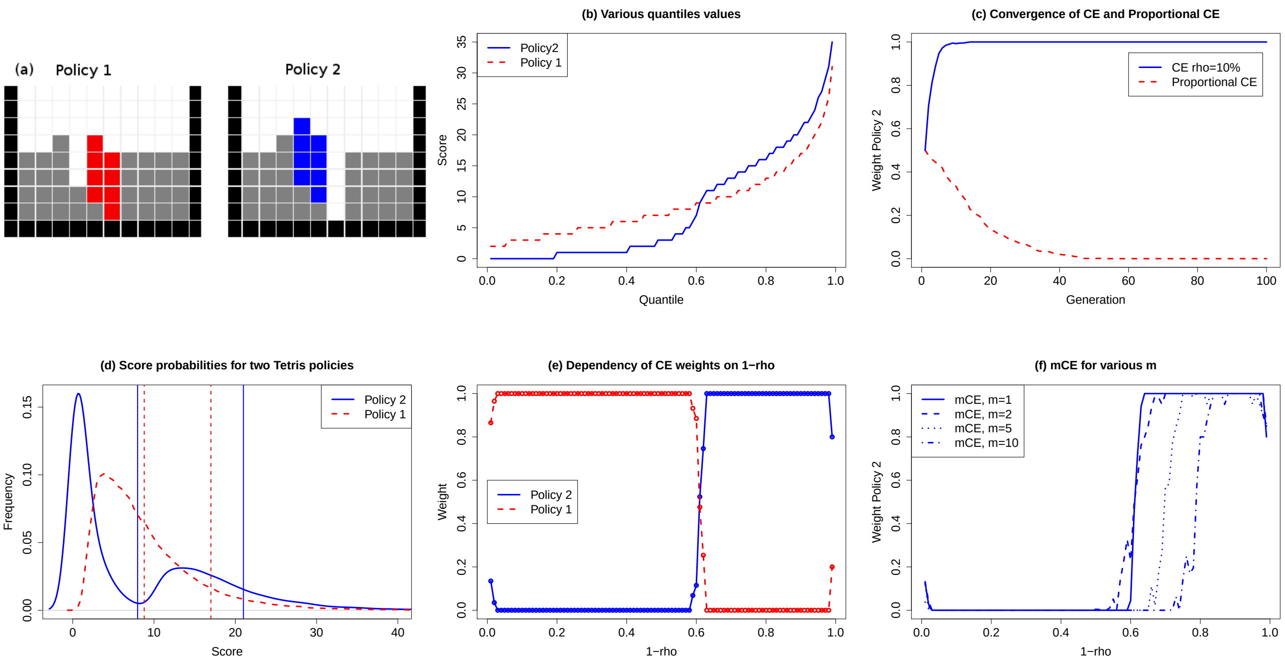

Figure 1. A simple experiment for two policies in Tetris.

ing $1, 2, 3$, and 4 lines were $1, 2, 3$ and 10, respectively. The optimization problem is to pick among two policies the one that has the best average score. Policy 1 is less "risky" and it is represented in Fig. 1(a) (left). It receives a pair of S tetrominoes that it positions as shown in the diagram. Policy 2 positions the two S tetraminoes as in Fig. 1(a) (right). Both policies continue by executing the same fixed strategy (that was obtained offline by running a CE algorithm as in section 5.3) when exposed to random tetrominoes arriving according to the distribution specified above. We executed each policy for 50k steps and plotted the distribution over scores in Fig. 1(d). The ordering by the means is different from the ordering determined by the 90% quantile and in fact the same holds true for a wide range of quantiles (as can be observed in Fig. 1(b), where we plot the empirical quantile functions for the scores of Policy 1 & 2).

We ran CE and Proportional CE with a Bernoulli distribution with just one component that captures the binary choice between Policy 1 and 2. Every experiment was executed 100 iterations and was repeated 20 times with $N = 200$. The result in Fig. 1(c) shows the two algorithms converging to different solutions, with CE($\rho = 10\%$) converging to Policy 2 (which has a higher 90% quantile) and Proportional CE converging to Policy 1 (which has a higher expectation). To verify the phenomenon for a wide range of ρ values, we ran CE with $\rho \in (1\%, 99\%)$ and plotted the results in Fig. 1(e). The solution for CE(ρ) is consistent with the ordering of the quantiles from Fig. 1(b) with a transition stage for $\rho \approx 40\%$ (i.e. for the 60%th quantile), where the quantiles values for the two policies are similar.

To illustrate the behavior of mCE, we repeated the experiment for various values of m. For a fixed m, we varied $\rho \in (1\%, 99\%)$, keeping the number of evaluations per generation fixed and plotted the results in Fig. 1(f). For increasing values of m, although the range of ρ values for which the suboptimal policy is chosen shrinks, the phenomenon does not disappear.

4. Theoretical Results

In this section we will formally show that CE indeed optimizes for the quantiles of a function while Proportional CE optimizes for expectation. We will study the properties of the algorithms in a discrete, stochastic optimization setting under the assumption that an infinity of evaluations are available at any iteration (we note that the algorithms are fixed and don't take advantage of such knowledge). The model is well known in the evolutionary optimization community under the name of "infinite population model" (Vose, 1998). And, while obviously unrealizable, it is a reasonable model for studying qualitative properties of evolutionary methods and is consistent with results for situations where the number of evaluations is large. Thus, we opted for model simplicity with the goal of providing insights about the empirical results.

We assume that given a set $X = \{1, 2, ..., n\}$ and a noisy function $F : X \to [0, 1]$, whenever $F(x), x \in X$

is evaluated it will return a sample from a distribution $\mathcal{P}^x$ (that depends on x) over $[0,1]$. Since our results are based on the technical tools from Goschin et al. (2011), we will make similar assumptions. We will assume that $\mathcal{P}^x$ have strictly increasing cumulative distribution functions $\mathcal{H}^x$ and common support $[0,1]$. Since the cdf's $\mathcal{H}^x$ are strictly increasing, the quantiles $q^x(\tau)$ are uniquely defined for any probability $\tau \in [0\%, 100\%]$: $q^x(\tau) = (\mathcal{H}^x)^{-1}(\tau)$. The standard goal for optimization in the above model is to find an $x^* = \arg\max_x \mathbb{E}_{F(x)\sim\mathcal{P}^x}[F(x)]$. An alternative goal is to optimize for quantiles: for a fixed τ, find $x^\tau = \arg\max_x q^x(\tau)$. It is possible that the two optimization objectives are aligned ($x^* = x^\tau, \forall\tau$) as in the case of an additive noise distribution like $\text{Beta}(\alpha,\alpha)$ that is input-independent for example. But, for general $\mathcal{P}^x$ and τ, the objectives are different.

We will study the optimization objectives of CE and Proportional CE assuming they are allowed an infinite number of evaluations at every iteration. Informally, the idea of having an infinite number of evaluations is to allow the entire distribution over $F(x)$ values for a particular value x to be "present" in the set of samples for a fixed generation. This assumption naturally removes the need for a parameter N. But we still need to study the convergence properties for a particular family of distributions $\mathcal{D}_t$ and for a particular setting of the initial parameters $\mathcal{D}_0$. The reason is that in general, without fixing $\mathcal{D}_0$, there will always be a setting for which the algorithms are guaranteed not to converge:

Proposition 1. *In the setting above, there exists an initial distribution $\mathcal{D}_0$ that forces both CE and Proportional CE to never find x^*.*

Proof. Assume x^* is unique in maximizing $\mathbb{E}[F]$. Let's consider a binary encoding of the input space in $\log(n)$ bits and consider $\mathcal{D}_t$ to be multivariate Bernoulli distributions over vectors of size $\log(n)$. Let's assume wlog that bit 0 of x^* is set to 1. Let's now choose the initial distribution $\mathcal{D}_0$ to have a $\text{Bernoulli}(p=0)$ distribution on the first bit of the representation. Then x^* will never be sampled from $\mathcal{D}_t, \forall t \geq 0$. □

Since the results are distribution-dependent, we will prove the convergence properties for a natural choice of a distribution and initial parameters and conjecture that the results can be extended to other distributions as well. Both algorithms will use a multinomial distribution $\mathcal{M}_t$ over the input space with all parameters $p_0^x = \frac{1}{n}, x \in X$ initially (hence the subscript 0). Using a multinomial distribution is reasonable in this context since it encodes the degree of "belief" the algorithm has in a particular x being the optimal argument of the function. We will prove that[1]:

Theorem 1. *When running CE($\rho, \mathcal{M}_0(p_0^x = \frac{1}{n})$) to optimize a function F, the algorithm will asymptotically converge to $x^{1-\rho}$ (i.e. to a multinomial with $p_\infty^{x^{1-\rho}} = 1$). When running Proportional CE ($\mathcal{M}_0(p_i = \frac{1}{n})$), the algorithm will asymptotically converge to x^*.*

Proof. The first idea of the proof is to use the fact that the third stage of the CE algorithm is maximum likelihood estimation for the multinomial distribution based on the top ρ percent (or equivalently, the $1-\rho$ quantile) of the evaluated, infinite population. For a motivation of the maximum likelihood claim, the reader is referred to Boer et al. (2005) (Remark 2.5 in particular). This perspective automatically provides closed form solutions for the updates of the parameters for $\mathcal{M}_t$ (as opposed to solving a potentially complicated stochastic program as it is the case in general). In particular, $p_{t+1}^x = p_t^x \frac{1-\mathcal{H}^x(a)}{\rho}$ (where a is the threshold value in $[0,1]$ that separates the elite from the rest of the population). In words, each component's new weight p_{t+1}^x is proportional to the relative tail probability mass (which also takes into consideration the previous weight p_t^x) with respect to the other components among the elite sample. The key insight is to observe that the parameter update rule above coincides with the weight update rules for a genetic algorithm using a truncation selection operator thus reducing the proof of convergence to the result from Goschin et al. (2011) (Theorem 4.2).

The proof for Proportional CE is similar, but with different updates and a reduction to Theorem 4.1 from Goschin et al. (2011) instead. □

5. Experiments

The goal of this section is to present empirical results in support of the claim that CE fails to optimize for expectation in naturally occurring noisy environments.

5.1. Die4

Die4 is a game introduced in Goschin et al. (2011) to study genetic algorithms in the context of optimization under risk. The game is played with a regular die. At each point in time the player can decide to roll the die or to stop and accumulate the sum of all die values until the current time. If, however, the die comes up 4 at any roll, the game ends and the player gets 0 points. Depending on the attitude towards risk, policies can

[1] We note that given the usual values of ρ ($\rho < 0.5$), CE is thus risk-seeking which can be dangerous in practice.

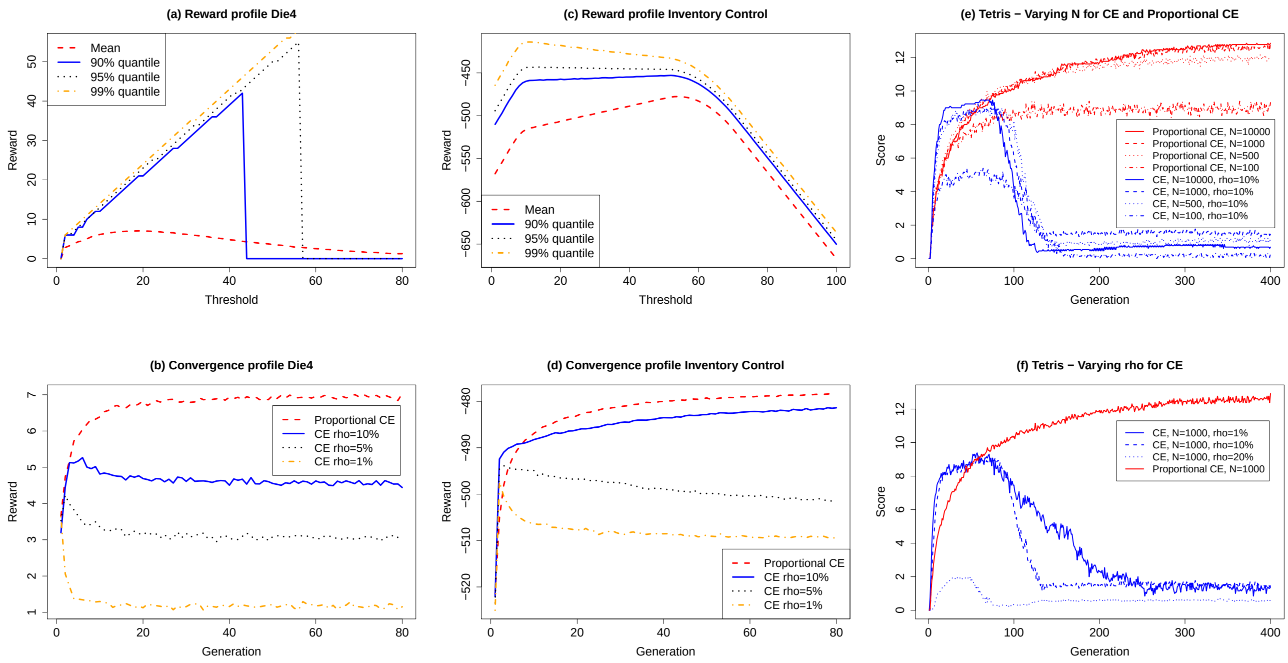

Figure 2. Die4, Inventory Control and Tetris experiments

stop earlier and have a good chance of gaining a non-0 reward or stop later with a high risk of gaining nothing.

Model. The states are the possible sum values for a die (natural numbers >= 2), the actions are *roll* and *stop* (both can be terminal), the rewards are 0 for failure and the value of the state for success. The transitions for the *roll* action are dictated by the roll of the die according to the definition of the game while the transition for *stop* is to the same state (and the game stops). We follow Goschin et al. (2011) and define the policy space to be parameterized by a threshold x encoding a simple rule: "*stop* as soon as the sum of die values is at least x or *roll* otherwise". The (expected) optimal value is ≈ 7.2 and it is obtained by setting $x^* = 17$. In Fig. 2(a), we plot the curve for the mean scores for all the policies with thresholds in $\{2, ..., 80\}$. We also plot the curves corresponding to the $90, 95, 99\%$th quantiles.

Setup. We relax $x \in \mathbb{R}^+$ (even though the sums are discrete) so as to be able to apply CE easily. Both algorithms start with a normal distribution $N(\mu = 50, \sigma^2 = 100)$ over the set of thresholds, $N = 1000$ and are executed for 80 iterations. Each experiment is repeated 50 times and the average scores are reported.

Results. The results in Fig. 2(b) show Proportional CE converging to the optimal expected value. The distribution over the thresholds after 80 iterations is concentrated around the optimal threshold. On the other hand, CE with $\rho = 10\%, 5\%$ or 1% converged to sub-optimal values and actually finds solutions that are optimal according to the corresponding quantiles. The results are consistent with what the theory predicts for such a scenario where the input that yields a maximum expected value is different from inputs that determine optimal quantile values.

5.2. Inventory Control

Inventory Control is a standard benchmark problem from operations research. It was also used as an experimental domain in the first paper that utilized CE for policy search in RL (Mannor et al., 2003). We will describe the simplest version of the problem, which models a shop owner having to make decisions about ordering one product.

Model. The state space consists of possible stock values at the beginning of each day: $s_t \in \mathbb{R}$ (with $t > 0$ denoting the day), and negative stock possible due to under-ordering. For each state, the action space $a_t \in [0, s_{\max} - s_t]$ is the amount of stock the owner can order at the beginning of day t (with s_{max} being the maximum stock). The transition function is determined by i.i.d. requests from clients $d_t \sim \mathcal{P}$ from a fixed, but unknown probability distribution with the next state being determined by $s_{t+1} = s_t + a_t - d_t$. The costs for "holding"(h) too much stock, "backlogging" (b) due to insufficient ordering and the price for one unit of stock c are fixed and known. We will use the same reward function as Mannor et al. (2003) $r_t(s_t, a_t, d_t) = -h \max\{0, s_t\} - b \max\{0, -s_t\} - ca_t$.

We will also use the same policy space as Mannor et al.

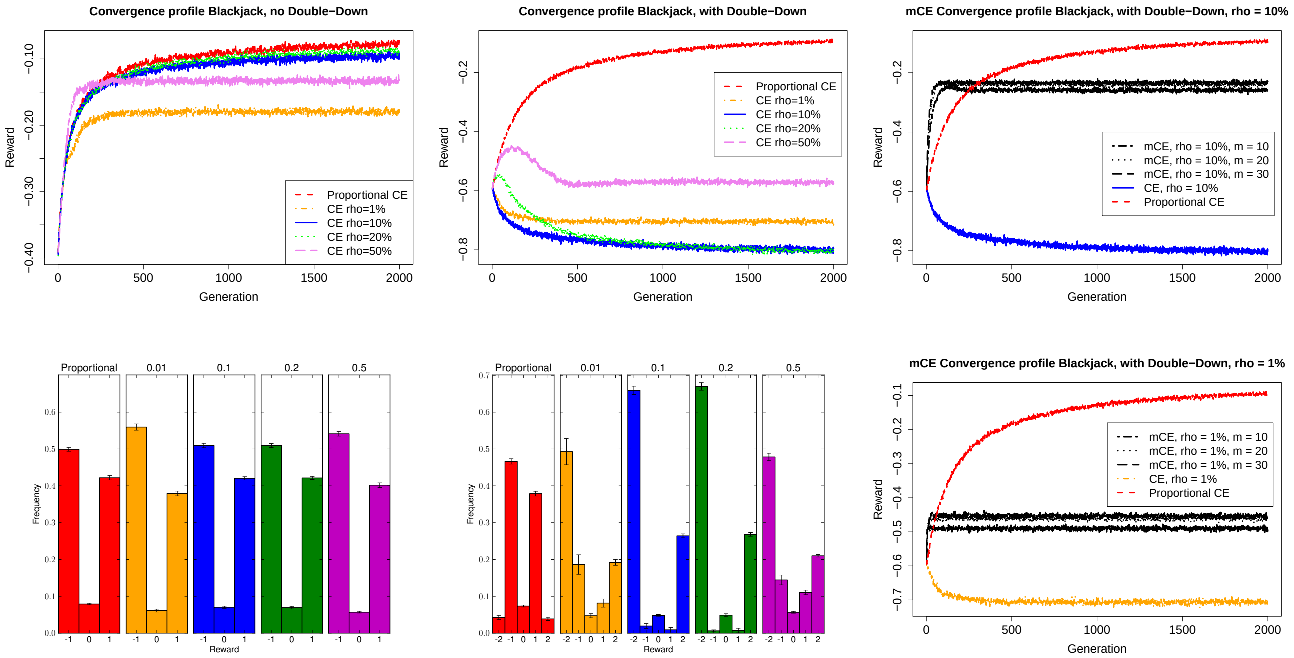

Figure 3. Blackjack experiments. Subfigures top (left to right): (a), (b), (c), down (left to right): (d), (e), (f)

(2003): each policy is determined by a threshold x that sets the stock order as a function of the current stock, meaning at every time t, $a_t = \max\{x - s_t, 0\}$. Searching in this policy space is thus equivalent to finding the best threshold x.

Setup. For the experiments, we instantiated an inventory control problem with the following characteristics: $h = 5, b = 6, c = 10, s_{\max} = 100$ and with $\mathcal{P}$ being a mixture of two normal distributions with equal weights ($N(\mu = 5, \sigma^2 = 5)$ and $N(\mu = 50, \sigma^2 = 20)$) in an attempt to model a mix of small and large requests. Similarly to Die4, both algorithms start with a Normal distribution $N(\mu = 50, \sigma^2 = 100)$ over the set of thresholds, $N = 1000$ and are executed for 80 iterations. Each experiment is repeated 50 times. In Fig. 2(c), we plotted the curves for the means and the same set of quantiles as for Die4. The optimal expected value is around -477 and it is obtained by setting $x = 53$ while the maximum 99% quantile corresponds to an expected value of -514 and is obtained by setting $x = 10$.

Results In Fig. 2(d), we plot the algorithms' convergence curves. It can be observed that Proportional CE converges to a value close to the expected optimal value while CE($\rho = 1\%$) for example converges, as expected, to the value corresponding to the optimal 99% quantile. As in the case of Die4, the solutions distributions over inputs that the algorithms converge to are centered around the input values that are predicted by the theoretical results.

5.3. Tetris

The video game Tetris is well known for being a difficult benchmark for policy search in RL and one where CE performed very well in the past (Szita & Lörincz, 2006). In our experiments we followed closely the setup from Szita & Szepesvári (2010a) that defines a simpler version of Tetris (*Stochastic SZTetris*) that only allows the S and Z tetraminoes with the goal of maintaining the difficulty of the game and make it more efficient to simulate. We used the code base from Szita & Szepesvári (2010b) and extended it to parameterize the domain. We chose the feature representation defined in Bertsekas & Ioffe (1996). We refer the reader to Szita & Szepesvári (2010a) for an excellent presentation of the challenges of SZTetris.

In an attempt to do simulations more efficiently (so that we could run parameter search in reasonable time), we decreased the height of the SZTetris board to from 20 to 5 (the problem is far from trivial even in this modified setup). Moreover we decided to give a bonus of 10 points for clearing two lines (as compared to the default value of 2) with the goal of "infusing risk" in the game. We ran a parameter search for a reasonable value of N (convergence results can be seen in Fig. 2(e) where ρ is fixed to 10%) and for a good ρ value for CE (Fig. 2(f) with fixed $N = 1000$). Every experiment is repeated 15 times and the results are averaged. The first observation is that while Proportional CE converges slower than CE, it will converge to better solutions than the maximum performance of

CE throughout its execution. The second observation is that after its performance plateaus, CE is degrading no matter how we set the initial parameters. We made significant efforts to find a setting of the algorithm where the divergence phenomenon doesn't happen (including setting various smoothing parameters, initial variance etc.) but we were unable to eliminate it. To verify that this was not a direct cause of our setup, we ran the original code base with the original algorithm and SZTetris parameters and found the same phenomenon occurring around generation 2000 (for reasonable tractability reasons, Szita & Szepesvári (2010a) only ran the algorithm up to generation 50).

While the experiments above offer a less clear-cut perspective with respect to the main goal of the paper, we believe they are interesting enough to report. Even in the region where CE converges, its performance is worse than what Proportional CE can achieve. We note that this seems to be a direct cause of the increased bonus for clearing two lines. In experiments with the original scores for clearing lines, the average best performances of the two algorithms are essentially the same (even though CE still degrades). This suggests that the phenomenon of optimizing for different objectives affects this "risky" version of Tetris.

5.4. Blackjack

In this section, we discuss two variants of blackjack and describe how differences in mechanics can lead to changes in policies when optimizing for quantiles or expectation. The first variant of the game reduces the game to its most important dynamics, as described in Sutton & Barto (1998). We also adopt the policy representation of Sutton & Barto (1998). The state is represented by the dealer's showing card, the sum of the player's hand, and whether or not the player holds a usable ace. On hand values less than 12, the player automatically hits, because there is no chance of busting. Therefore, the game can be represented with $n = 200$ states with 2 actions (the distribution over which is binomial).

The experiment is run for 2,000 generations, with $N = 10000$. Each experiment is repeated 10 times. Fig. 3(a) shows the average reward per generation over each of the 10 executions of CE with various selection methods. As can be seen, policy improvement occurs most rapidly with $\rho = 50\%$, but levels off quite rapidly. It is then surpassed by CS Proportional, which produces the highest quality policies for the rest of the experiment. The distribution of rewards according to strategy is depicted in Fig. 3(d), with error bars displaying the standard deviation of the average of the 10 final populations in each experiment. While Proportional CE produces the best policy, the difference between Proportional CE and CE is minimal.

In the second variant tested, the option to *double* is introduced. This action causes the player to double the wager (after which payoffs can be only $-2, 0,$ or 2), hit, and then stick. All other details are identical to the first setting, and the dealer is not able to apply this action. The performance of the various CE variants is rendered in Fig. 3(b). While in the original variant, CE improved all policies over time, only the proportional strategy resulted in consistent improvement over time when doubling was allowed. Both CE with $\rho = 20\%, 50\%$ initially improved, but later degraded, with $\rho = 50\%$ being essentially equal to chance performance by the end of the experiment, and all other policies produced by non-proportional selection being worse than chance. As can be seen in the distributions over rewards in Fig. 3(e), Proportional CE exercises the double action less than 10% of the time, and has a PDF markedly different from the other strategies. In particular, CE with $\rho = 10\%, 20\%$ both performed the worst, and doubled the most (almost 95% of the time), and lost almost 1/3 of all bets where doubling was used, resulting in very poor performance.

mCE. We also ran experiments to verify the behavior of mCE in the context of the second variant of Blackjack. As rendered in Fig. 3(c) and (f) (for two values of ρ), even with as high as $m = 30$ samples per individual, the quality of the mCE algorithm doesn't match Proportional CE. The performance improves (as compared to CE) as more samples per input are added (up to a point due to the finite size of the population), but it is not clear how to set m and N such that mCE performs at least as good as Proportional CE.

6. Conclusion

The goal of the paper was to discuss the impact of naturally occurring evaluation noise on the performance of a well known optimization algorithm: the cross-entropy method. We proved that sometimes CE optimizes for a different criterion than the maximum expected value of a function, namely a quantile metric. We proposed an algorithm that has the same structure but optimizes for the correct objective. We also described a variety of naturally occurring optimization problems which determine CE to behave sub-optimally in a way consistent with the theoretical results.

Acknowledgements This research was supported in part by National Science Foundation DGE-0549115.

References

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

Bertsekas, Dimitri P. and Ioffe, Sergey. Temporal differences-based policy iteration and applications in neuro-dynamic programming. Technical report, MIT, 1996.

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

Chang, H.S., Fu, M.C., Hu, J., and Marcus, S.I. *Simulation-based Algorithms for Markov Decision Processes*. Springer-Verlag New York, Inc., 2007.

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

Fitzpatrick, J.M. and Grefenstette, J.J. Genetic algorithms in noisy environments. *Machine Learning*, 3: 101–120, 1988.

Goschin, S., Littman, M.L., and Ackley, D.H. The effects of selection on noisy fitness optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2011.

Hansen, N. and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolution Computation*, 9(2):159–195, 2001.

Kalyanakrishnan, S. and Stone, P. An empirical analysis of value function-based and policy search reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 2009.

Mannor, S., Rubinstein, R., and Gat, Y. The cross entropy method for fast policy search. In *International Conference on Machine Learning*, 2003.

Mannor, S., Peleg, D., and Rubinstein, R. The cross entropy method for classification. In *International Conference on Machine learning*, 2005.

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

Rubinstein, R. Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112, 1996.

Rubinstein, R. The cross-entropy method for combinatorial and continuous optimization. *Methodology And Computing In Applied Probability*, 1:127–190, 1999.

Rubinstein, R. and Kroese, D.P. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer, 2004.

Stulp, F. and Sigaud, O. Path integral policy improvement with covariance matrix adaptation. In *International Conference on Machine learning*, 2012.

Sutton, Richard S. and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Szita, I. and Lörincz, A. Learning Tetris using the noisy cross-entropy method. *Neural Computation*, 18(12):2936–2941, 2006.

Szita, I. and Szepesvári, C. SZ-Tetris as a benchmark for studying key problems of reinforcement learning. In *ICML 2010 Workshop on Machine Learning and Games*, 2010a.

Szita, I. and Szepesvári, C. sztetris-rl Library. http://code.google.com/p/sztetris-rl/, 2010b.

Vose, M.D. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, 1998.