NOTE Communicated by Andrew Barto

Learning Tetris Using the Noisy Cross-Entropy Method

István Szita
szityu@eotvos.elte.hu
András Lőrincz
andras.lorincz@elte.hu
Department of Information Systems, Eötvös Loránd University, Budapest, Hungary H-1117

The cross-entropy method is an efficient and general optimization algorithm. However, its applicability in reinforcement learning (RL) seems to be limited because it often converges to suboptimal policies. We apply noise for preventing early convergence of the cross-entropy method, using Tetris, a computer game, for demonstration. The resulting policy outperforms previous RL algorithms by almost two orders of magnitude.

1 Introduction

Tetris is one of the most popular computer games (see, e.g., Fahey, 2003). Despite its simple rules, playing the game well requires a complex strategy and lots of practice. Furthermore, Demaine, Hohenberger, and Liben-Nowell (2003) have shown that Tetris is hard in a mathematical sense as well. Finding the optimal strategy is NP-hard even if the sequence of tetrominoes is known in advance. These properties make Tetris an appealing benchmark problem for testing reinforcement learning (and other machine learning) algorithms.

Reinforcement learning (RL) algorithms are quite effective in solving a variety of complex sequential decision problems. Despite this, RL approaches tried to date on Tetris show surprisingly poor performance. The aim of this note is to show how to improve RL for this hard combinatorial problem. In this article, we put forth a modified version of the cross-entropy (CE) method (de Boer, Kroese, Mannor, & Rubinstein, 2004).

2 Applying the Cross-Entropy Method to Tetris

2.1 Value Function and Action Selection. Following the approach in Bertsekas and Tsitsiklis (1996), we shall learn state-value functions that are linear combination of several basis functions. We use 22 such basis functions: maximal column height, individual column heights, differences of column heights, and the number of holes. More formally, if s denotes a Tetris state and $\phi_i(s)$ is the value of basis function i in this state, then according to weight vector w, the value of state s is

$$V_w(s) := \sum_{i=1}^{22} w_i \phi_i(s). \tag{2.1}$$

During a game, the actual tetromino is test-placed in every legal position, and after erasing full rows (if any), the value of the resulting state is calculated according to V_w. Finally, we choose the column and direction with the highest value.

2.2 The Cross-Entropy Method. The cross-entropy (CE) method is a general algorithm for (approximately) solving global optimization tasks of the form

$$w^* = \arg\max_w S(w), \tag{2.2}$$

where S is a general real-valued objective function, with an optimum value $\gamma^* = S(w^*)$. The main idea of CE is to maintain a distribution of possible solutions and update this distribution at each step (de Boer et al., 2004). Here a very brief overview is provided.

The CE method starts with a parametric family of probability distributions $\mathcal{F}$ and an initial distribution $f_0 \in \mathcal{F}$. Under this distribution, the probability of drawing a high-valued sample (having value near γ^*) is presumably very low; therefore, finding such samples by naive sampling is intractable. For any $\gamma \in \mathbb{R}$, let $g_{\geq\gamma}$ be the uniform distribution over the set $\{w : S(w) \geq \gamma\}$. If one finds the distribution $f_1 \in \mathcal{F}$ closest to $g_{\geq\gamma}$ with regard to the cross-entropy measure, then f_0 can be replaced by f_1 and γ-valued samples will have larger probabilities. For many distribution families, the parameters of f_1 can be estimated from samples of f_0. This estimation is tractable if the probability of the γ-level set is not very low with regard to f_0. Instead of the direct computation of the $\mathcal{F}$-distribution closest to $g_{\geq\gamma^*}$, we can proceed iteratively. We select a γ_0 appropriate for f_0, update the distribution parameters to obtain f_1, select γ_1, and so on, until we reach a sufficiently large γ_k. Below we sketch the special case when w is sampled from a member of the gaussian distribution family.

Let the distribution of the parameter vector at iteration t be $f_t \sim N(\mu_t, \sigma_t^2)$. After drawing n sample vectors $w_1, \ldots, w_n$ and obtaining their value $S(w_1), \ldots, S(w_n)$, we select the best $\lfloor \rho \cdot n \rfloor$ samples, where $0 < \rho < 1$ is the selection ratio. This is equivalent to setting $\gamma_t = S(w_{\lfloor \rho \cdot n \rfloor})$. Denoting the set of indices of the selected samples by $I \subseteq \{1, 2, \ldots, n\}$, the mean and the deviation of the distribution is updated using

$$\mu_{t+1} := \frac{\sum_{i \in I} w_i}{|I|} \tag{2.3}$$

and

$$\sigma^2_{t+1} := \frac{\sum_{i \in I} (w_i - \mu_{t+1})^T (w_i - \mu_{t+1})}{|I|}. \tag{2.4}$$

2.3 The Cross-Entropy Method and Reinforcement Learning. Applications of the CE method to RL include the parameter tuning of radial basis functions (Menache, Mannor, & Shimkin, 2005) and adaptation of a parameterized policy (Mannor, Rubinstein, & Gat, 2003). We apply CE to learn the weights of the basis functions, drawing each weight from an independent gaussian distribution.

2.4 Preventing Early Convergence. Preliminary investigations showed that applicability of CE to RL problems is restricted severely by the phenomenon that the distribution concentrates to a single point too fast. To prevent this, we adapt a trick frequently used in particle filtering: at each iteration, we add some extra noise to the distribution: instead of equation 2.4, we use

$$\sigma^2_{t+1} := \frac{\sum_{i \in I} (w_i - \mu_{t+1})^T (w_i - \mu_{t+1})}{|I|} + Z_{t+1}, \tag{2.5}$$

where Z_{t+1} is a constant vector depending only on t.

3 Experiments

In the experiments we used the standard Tetris game described in Bertsekas and Tsitsiklis (1996), scoring 1 point for each cleared row. Each parameter had an initial distribution of $N(0, 100)$. We set $n = 100$ and $\rho = 0.1$. Each drawn sample was evaluated by playing a single game using the corresponding value function. After each iteration, we updated distribution parameters using equations 2.3 and 2.5 and evaluated the mean performance of the learned parameters. This was accomplished by playing 30 games using V_{μ_t}, and averaging the results. (The large number of evaluation games was necessary because Tetris strategies have large performance deviations; Fahey, 2003.)

In experiment 1 we tested the original CE method (corresponding to $Z_t = 0$). As expected, deviations converge to 0 too fast, so the mean performance settles at about 20,000 points. In experiment 2 we used a constant noise rate

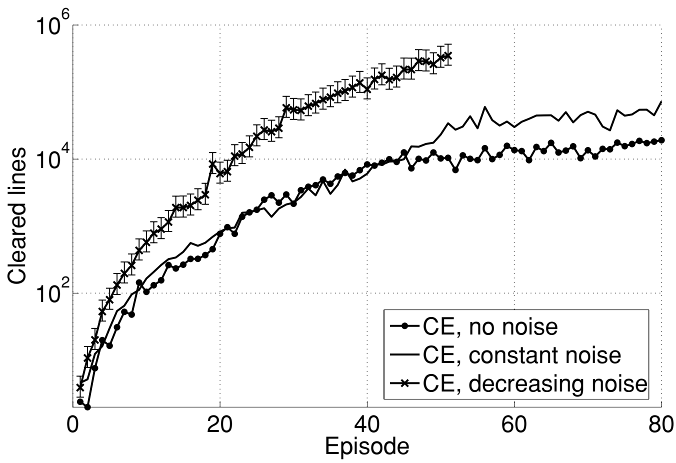

Figure 1: Tetris scores on logarithmic scale. Mean performance of 30 evaluations versus iteration number. Error bars denoting 95% confidence intervals are shown on one of the curves.

of $Z_t = 4$, raising mean performance to a 70,000 point level. Our analysis showed that further improvement was counteracted by the amount of noise, which was too high, and thus prevented convergence of the distributions. Therefore, in experiment 3, we applied a decreasing amount of noise, $Z_t = \max(5 - \frac{t}{10}, 0)$. With this setting, average score exceeded 300,000 points by the end of episode 50, and the best score exceeded 800,000 points. In experiments 2 and 3, noise parameters were selected in an ad hoc manner, and no optimization was carried out. Results are summarized in Figure 1 and in Table 1.

Assuming an exponential score distribution (Fahey, 2003), we calculated the 95% confidence intervals of the mean. For better visibility, confidence intervals have been plotted only for experiment 3.

Learning took more than one month of CPU time on a 1 GHz machine using Matlab. The main reason for the long learning course is the fact that for Tetris, evaluation time of the value function scales linearly with the score, and the score is very noisy.[1] Related to this, the computational overhead of the CE method is negligibly small.

Because of the large running times, experiments consisted of only a single training run. However, preliminary results on simplified Tetris problems show that over multiple trials, the method consistently converges to the same region.

[1] It is conjectured that the length of a game can be approximated from its starting sequence (Fahey, 2003), which could reduce evaluation time considerably.

Table 1: Average Tetris Scores of Various Algorithms.

Method	Mean Score	Reference
Nonreinforcement learning		
Hand-coded	631,167	Dellacherie (Fahey, 2003)
Genetic algorithm	586,103	(Böhm et al., 2004)
Reinforcement learning		
Relational reinforcement learning+kernel-based regression	≈50	Ramon and Driessens (2004)
Policy iteration	3183	Bertsekas and Tsitsiklis (1996)
Least squares policy iteration	<3000	Lagoudakis, Parr, and Littman (2002)
Linear programming + Bootstrap	4274	Farias and van Roy (2006)
Natural policy gradient	≈6800	Kakade (2001)
CE+RL	21,252	
CE+RL, constant noise	72,705	
CE+RL, decreasing noise	348,895	

3.1 Comparison to Previous Work. Tetris has been chosen as a benchmark problem by many researchers to test their RL algorithms. Table 1 summarizes results known to us, comparing them to our algorithm and to two state-of-the-art non-RL algorithms as well.

The comparison shows that our method improves on the performance of the best RL algorithm by almost two orders of magnitude and gets close to the best non-RL algorithms (Fahey, 2003; Böhm, Kókai, & Mandl, 2004).

We think that by applying the performance enhancement techniques in Böhm et al. (2004)—more basis functions, exponential value function—further significant improvement is possible.

References

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Nashua, NH: Athena Scientific.

Böhm, N., Kókai, G., & Mandl, S. (2004). Evolving a heuristic function for the game of Tetris. In T. Scheffer (Ed.), *Proc. Lernen, Wissensentdeckung und Adaptivität LWA—2004* (pp. 118–122). Berlin.

de Boer, P., Kroese, D., Mannor, S., & Rubinstein, R. (2004). A tutorial on the cross-entropy method. *Annals of Operations Research, 134*(1), 19–67.

Demaine, E. D., Hohenberger, S., & Liben-Nowell, D. (2003). Tetris is hard, even to approximate. In *Proc. 9th International Computing and Combinatorics Conference (COCOON 2003)* (pp. 351–363). Berlin: Springer.

Fahey, C. P. (2003). Tetris AI. Available online at http://www.colinfahey.com

Farias, V. F., & van Roy, B. (2006). *Tetris: A study of randomized constraint sampling*. In G. Calafiore & F. Dabbene (Eds.), *Probabilistic and randomized methods for design under uncertainty*. Berlin: Springer-Verlag.

Kakade, S. (2001). A natural policy gradient. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14* (pp. 1531–1538). Cambridge, MA: MIT Press.

Lagoudakis, M. G., Parr, R., & Littman, M. L. (2002). Least-squares methods in reinforcement learning for control. In *SETN '02: Proceedings of the Second Hellenic Conference on AI* (pp. 249–260). Berlin: Springer-Verlag.

Mannor, S., Rubinstein, R. Y., & Gat, Y. (2003). The cross-entropy method for fast policy search. In *Proc. International Conf. on Machine Learning (ICML 2003)*, (pp. 512–519). Menlo Park, CA: AAAI Press.

Menache, I., Mannor, S., & Shimkin, N. (2005). Basis function adaption in temporal difference reinforcement learning. *Annals of Operations Research, 134*(1), 215–238.

Ramon, J., & Driessens, K. (2004). On the numeric stability of gaussian processes regression for relational reinforcement learning. In *ICML-2004 Workshop on Relational Reinforcement Learning* (pp. 10–14). N.p.: Omni press.

Received October 3, 2005; accepted May 15, 2006.

This article has been cited by:

1. Boyi Zhang, Pengjian Shang. 2019. Complexity and uncertainty analysis of financial stock markets based on entropy of scale exponential spectrum. *Nonlinear Dynamics* **98**:3, 2147-2170. [Crossref]
2. Xin Tong, Weiming Liu, Bin Li. Enhancing Rolling Horizon Evolution with Policy and Value Networks 1-8. [Crossref]
3. Hangkai Hu, Shiji Song, C. L. Phillip Chen. 2019. Plume Tracing via Model-Free Reinforcement Learning Method. *IEEE Transactions on Neural Networks and Learning Systems* **30**:8, 2515-2527. [Crossref]
4. Xinghua Qu, Yew-Soon Ong, Yaqing Hou, Xiaobo Shen. Memetic Evolution Strategy for Reinforcement Learning 1922-1928. [Crossref]
5. Yang Yang, Anusha Lalitha, Jinwon Lee, Chris Lott. Automatic Grammar Augmentation for Robust Voice Command Recognition 6376-6380. [Crossref]
6. Zhiyuan Jiang, Sheng Zhou, Zhisheng Niu. Distributed Policy Learning Based Random Access for Diversified QoS Requirements 1-6. [Crossref]
7. Shixun You, Ming Diao, Lipeng Gao. 2019. Completing Explorer Games with a Deep Reinforcement Learning Framework Based on Behavior Angle Navigation. *Electronics* **8**:5, 576. [Crossref]
8. John K. Lindstedt, Wayne D. Gray. 2019. Distinguishing experts from novices by the Mind's Hand and Mind's Eye. *Cognitive Psychology* **109**, 1-25. [Crossref]
9. Sarthak Bhagat, Hritwick Banerjee, Zion Ho Tse, Hongliang Ren. 2019. Deep Reinforcement Learning for Soft, Flexible Robots: Brief Review with Impending Challenges. *Robotics* **8**:1, 4. [Crossref]
10. Abhinav Nagpal, Goldie Gabrani. Strategizing Game Playing Using Evolutionary Approach 481-492. [Crossref]
11. Giacomo Da Col, Erich C. Teppan. Heuristic Search for Tetris: A Case Study 410-423. [Crossref]
12. Catherine Sibert, Wayne D. Gray. 2018. The Tortoise and the Hare: Understanding the Influence of Sequence Length and Variability on Decision-Making in Skilled Performance. *Computational Brain & Behavior* **1**:3-4, 215-227. [Crossref]
13. Guangliang Li, Shimon Whiteson, W. Bradley Knox, Hayley Hung. 2018. Social interaction for efficient agent learning from human reward. *Autonomous Agents and Multi-Agent Systems* **32**:1, 1-25. [Crossref]
14. Benjamin J. Hodel. 2018. Learning to Operate an Excavator via Policy Optimization. *Procedia Computer Science* **140**, 376-382. [Crossref]
15. Benjamin Peherstorfer, Boris Kramer, Karen Willcox. 2018. Multifidelity Preconditioning of the Cross-Entropy Method for Rare Event Simulation and Failure Probability Estimation. *SIAM/ASA Journal on Uncertainty Quantification* **6**:2, 737-761. [Crossref]

16. Shankar Sadasivam, Zhuo Chen, Jinwon Lee, Rajeev Jain. Efficient reinforcement learning for automating human decision-making in SoC design 1-6. [Crossref]
17. Jacob Schrum. Evolving indirectly encoded convolutional neural networks to play tetris with low-level features 205-212. [Crossref]
18. Renan Samuel da Silva, Rafael Stubs Parpinelli. Playing the Original Game Boy Tetris Using a Real Coded Genetic Algorithm 282-287. [Crossref]
19. Murat Altun, Onur Pekcan. 2017. A modified approach to cross entropy method: Elitist stepped distribution algorithm. *Applied Soft Computing* **58**, 756-769. [Crossref]
20. Jose M. Font, Daniel Manrique, Sergio Larrodera, Pablo Ramos Criado. Towards a hybrid neural and evolutionary heuristic approach for playing tile-matching puzzle games 76-79. [Crossref]
21. Aaron Isaksen, Drew Wallace, Adam Finkelstein, Andy Nealen. Simulating strategy and dexterity for puzzle games 142-149. [Crossref]
22. Catherine Sibert, Wayne D. Gray, John K. Lindstedt. 2017. Interrogating Feature Learning Models to Discover Insights Into the Development of Human Expertise in a Real-Time, Dynamic Decision-Making Task. *Topics in Cognitive Science* **9**:2, 374-394. [Crossref]
23. Lauren E. Gillespie, Gabriela R. Gonzalez, Jacob Schrum. Comparing direct and indirect encodings using both raw and hand-designed features in tetris 179-186. [Crossref]
24. Eiji Uchibe, Jiexin Wang. 2017. Deterministic Policy Search Method for Real Robot Control. *The Brain & Neural Networks* **24**:4, 195-203. [Crossref]
25. Guangliang Li, Shimon Whiteson, W. Bradley Knox, Hayley Hung. 2016. Using informative behavior to increase engagement while learning from human reward. *Autonomous Agents and Multi-Agent Systems* **30**:5, 826-848. [Crossref]
26. H. J. Kappen, H. C. Ruiz. 2016. Adaptive Importance Sampling for Control and Inference. *Journal of Statistical Physics* **162**:5, 1244-1266. [Crossref]
27. Matthew Hausknecht, Joel Lehman, Risto Miikkulainen, Peter Stone. 2014. A Neuroevolution Approach to General Atari Game Playing. *IEEE Transactions on Computational Intelligence and AI in Games* **6**:4, 355-366. [Crossref]
28. Guangliang Li, Hayley Hung, Shimon Whiteson, W. Bradley Knox. Learning from human reward benefits from socio-competitive feedback 93-100. [Crossref]
29. Paul Wagner. 2014. Policy oscillation is overshooting. *Neural Networks* **52**, 43-61. [Crossref]
30. Tobias Jung, Louis Wehenkel, Damien Ernst, Francis Maes. 2014. Optimized look-ahead tree policies: a bridge between look-ahead tree policies and direct policy search. *International Journal of Adaptive Control and Signal Processing* **28**:3-5, 255-289. [Crossref]

31. Matthew S. Maxwell, Shane G. Henderson, Huseyin Topaloglu. 2013. Tuning Approximate Dynamic Programming Policies for Ambulance Redeployment via Direct Search. *Stochastic Systems* **3**:2, 322-361. [Crossref]
32. Kyriakos Efthymiadis, Daniel Kudenko. Using plan-based reward shaping to learn strategies in StarCraft: Broodwar 1-8. [Crossref]
33. Olivier Buffet. Policy-Gradient Algorithms 127-152. [Crossref]
34. Dimitri P. Bertsekas. Lambda-Policy Iteration: A Review and a New Implementation 379-409. [Crossref]
35. Vijay V. Desai, Vivek F. Farias, Ciamac C. Moallemi. 2012. Approximate Dynamic Programming via a Smoothed Linear Program. *Operations Research* **60**:3, 655-674. [Crossref]
36. István Szita. Reinforcement Learning in Games 539-577. [Crossref]
37. Hado van Hasselt. Reinforcement Learning in Continuous State and Action Spaces 207-251. [Crossref]
38. Amine Boumaza. Reducing the Learning Time of Tetris in Evolution Strategies 193-204. [Crossref]
39. Mohan Sridharan. Augmented Reinforcement Learning for Interaction with Non-expert Humans in Agent Domains 424-429. [Crossref]
40. Dimitri P. Bertsekas. 2011. Approximate policy iteration: a survey and some new methods. *Journal of Control Theory and Applications* **9**:3, 310-335. [Crossref]
41. Samuel Sarjant, Bernhard Pfahringer, Kurt Driessens, Tony Smith. Using the online cross-entropy method to learn relational policies for playing different games 182-189. [Crossref]
42. Pierre Minvielle, Emilia Tantar, Alexandru-Adrian Tantar, Philippe Berisset. 2011. Sparse Antenna Array Optimization With the Cross-Entropy Method. *IEEE Transactions on Antennas and Propagation* **59**:8, 2862-2871. [Crossref]
43. Shivaram Kalyanakrishnan, Peter Stone. 2011. Characterizing reinforcement learning methods through parameterized learning problems. *Machine Learning* **84**:1-2, 205-247. [Crossref]
44. Marek Petrik, Shlomo Zilberstein. Learning Feature-Based Heuristic Functions 269-305. [Crossref]
45. Dimitri P. Bertsekas. Pathologies of temporal difference methods in approximate dynamic programming 3034-3039. [Crossref]
46. Matthew S. Maxwell, Shane G. Henderson, Huseyin Topaloglu. Identifying effective policies in approximate dynamic programming: Beyond regression 1079-1087. [Crossref]
47. Shimon Whiteson, Matthew E. Taylor, Peter Stone. 2010. Critical factors in the empirical performance of temporal difference and evolutionary methods for reinforcement learning. *Autonomous Agents and Multi-Agent Systems* **21**:1, 1-35. [Crossref]

48. Leo Langenhoven, Willem S. van Heerden, Andries P. Engelbrecht. Swarm Tetris: Applying particle swarm optimization to tetris 1-8. [Crossref]
49. Kokolo Ikeda, Shigenobu Kobayashi, Hajime Kita. 2010. Exemplar-Based Policy with Selectable Strategies and its Optimization Using GA. *Transactions of the Japanese Society for Artificial Intelligence* **25**, 351-362. [Crossref]
50. Amine Boumaza. On the evolution of artificial Tetris players 387-393. [Crossref]
51. Cornelius Weber, Jochen Triesch. Goal-directed feature learning 3319-3326. [Crossref]
52. I. Szita, M. Ponsen, P. Spronck. 2009. Effective and Diverse Adaptive Game AI. *IEEE Transactions on Computational Intelligence and AI in Games* **1**:1, 16-27. [Crossref]
53. W. Bradley Knox, Peter Stone. TAMER: Training an Agent Manually via Evaluative Reinforcement 292-297. [Crossref]
54. Leo Galway, Darryl Charles, Michaela Black. 2008. Machine learning in digital games: a survey. *Artificial Intelligence Review* **29**:2, 123-161. [Crossref]
55. András Lőrincz. Learning and Representation: From Compressive Sampling to the 'Symbol Learning Problem' 445-488. [Crossref]
56. Kokolo Ikeda, Hajime Kita. State evaluation strategy for exemplar-based policy optimization of dynamic decision problems 3685-3691. [Crossref]