POLITECNICO MILANO 1863

informatics mathematics Inria

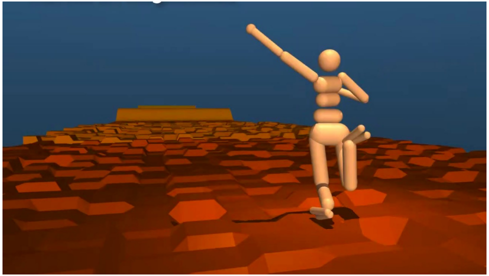
Stochastic Variance-Reduced Policy Gradient

Matteo Papini
Damiano Binaghi   Giuseppe Canonaco
Matteo Pirotta   Marcello Restelli

35th International Conference on Machine Learning, Stockholm, Sweden

Policy Gradient

An effective **Reinforcement Learning (RL)** solution to **continuous** control problems:



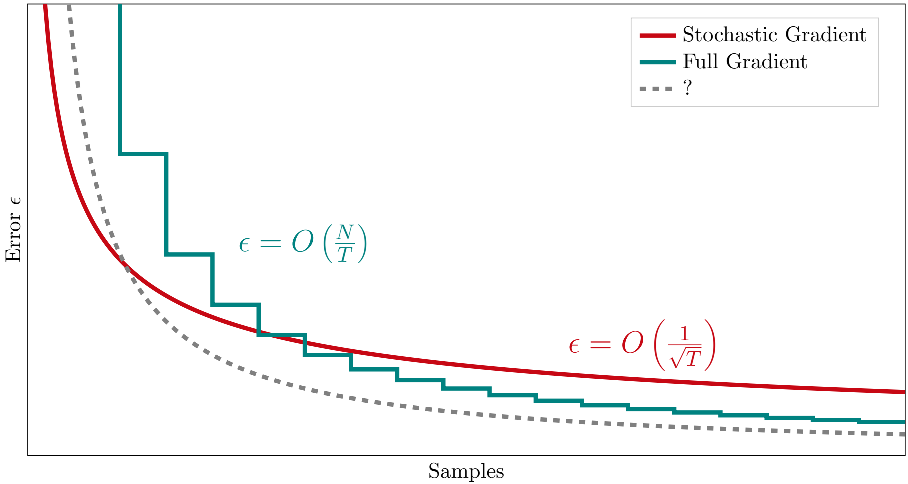
Robotics (Heess et al., 2017)

Video games (OpenAI, 2018)

Mostly based on **Stochastic Gradient Ascent** (Robbins and Monro, 1951)

Full vs Stochastic Gradient

Non-convex optimization



Can we do something better?

Visualization idea from Bach (2016)

SVRG

Stochastic Variance-Reduced Gradient

A solution from **finite-sum optimization** (Johnson and Zhang, 2013):

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_{i=1}^{N} f_i(\boldsymbol{\theta})$$

epoch

iteration

$$\underbrace{\blacktriangledown J(\boldsymbol{\theta})}_{\text{SVRG estimator}} = \underbrace{\nabla J(\widetilde{\boldsymbol{\theta}})}_{\text{FG (snapshot)}} + \underbrace{\nabla f_i(\boldsymbol{\theta})}_{\text{SG in current parameter}} - \underbrace{\nabla f_i(\widetilde{\boldsymbol{\theta}})}_{\text{Correction term}}$$

- Unbiased
- Linear convergence
- More data-efficient than FG
- **Supervised Learning (SL)**

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\tau} p(\tau|\boldsymbol{\theta}) R(\tau)$$

Not trivial! There are three **challenges**:

1. **Non-concavity** of $J(\boldsymbol{\theta})$ (Allen-Zhu and Hazan, 2016; Reddi et al., 2016)
2. **Infinite dataset**: we would need *infinite samples* to compute FG (Harikandeh et al., 2015; Bietti and Mairal, 2017)
3. **Non-stationarity**: $\tau \sim p_{\boldsymbol{\theta}}$ (new!)

RL so far: *policy evaluation* (Du et al., 2017) and *off-policy control* (Xu et al., 2017)

Our work: **on-policy control**

epoch

iteration

$$\underbrace{\blacktriangledown J(\boldsymbol{\theta})}_{\text{SVRPG estimator}} = \underbrace{\widehat{\nabla}_N J(\widetilde{\boldsymbol{\theta}})}_{\substack{\text{Large N} \\ \text{to approximate FG}}} + \underbrace{\widehat{\nabla}_B J(\boldsymbol{\theta})}_{B<<N} - \underbrace{\omega(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}})\widehat{\nabla}_B J(\widetilde{\boldsymbol{\theta}})}_{\substack{\text{Importance weighting} \\ \text{for non-stationarity}}}$$

- Unbiased
- More data-efficient than FG
- **On-policy**: only the correction term is weighted

Convergence Properties

Convergence to **local** optimum:

$$\mathbb{E}\left[\|\nabla J(\boldsymbol{\theta})\|^2\right] \leq \frac{J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_0)}{\psi T} + \underbrace{\frac{\zeta}{N}}_{\text{Infinite dataset}} + \underbrace{\frac{\xi}{B}}_{\text{Nonstationarity}}$$

- Linear convergence + **error** (similar to Harikandeh et al., 2015)

Heuristics

Meta-parameter selection

- **Adaptive step size**: two ADAM (Kingma and Ba, 2014) annealing schedules

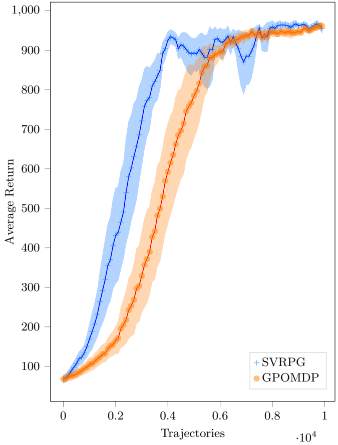
$$\underbrace{\alpha_{FG}}_{\text{used at the snapshot}} \qquad \underbrace{\alpha_{SG}}_{\text{used inside epoch}}$$

- **Adaptive epoch size**: new snapshot when effective step size becomes too small

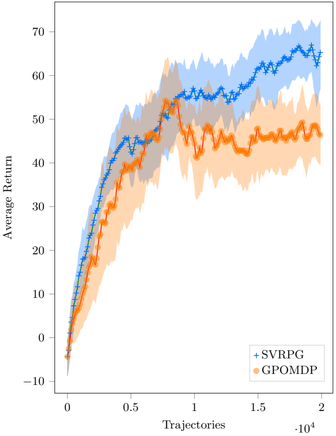
$$\frac{\alpha_{SG}}{B} < \frac{\alpha_{FG}}{N} \implies \textbf{snapshot}$$

Results

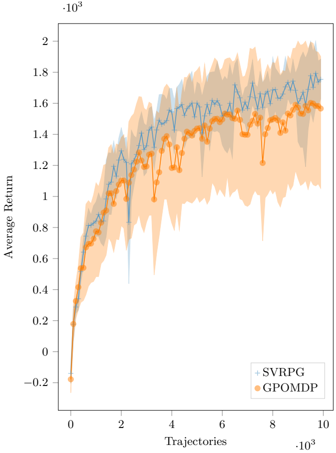
Cart-Pole



Swimmer



Half-Cheetah



Conclusions

- Efficient policy optimization is challenging
- **SVRPG**: on-policy control based on SVRG
- Meta-parameters still crucial to tame different sources of variance
- Future work: adaptive batch size, natural gradient, actor-critic

Thank you for your attention

- Poster: today 06:15 – 09:00 PM @ **Hall B #65**
- Contact: `matteo.papini@polimi.it`
- Online resources: `t3p.github.io`

References

Allen-Zhu, Z. and Hazan, E. (2016). Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707.

Bach, F. (2016). Stochastic optimization: Beyond stochastic gradients and convexity part i.

Baxter, J. and Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350.

Bietti, A. and Mairal, J. (2017). Stochastic optimization with variance reduction for infinite datasets with finite sum structure. In *Advances in Neural Information Processing Systems*, pages 1622–1632.

Du, S. S., Chen, J., Li, L., Xiao, L., and Zhou, D. (2017). Stochastic variance reduction methods for policy evaluation. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1049–1058. PMLR.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338.

Harikandeh, R., Ahmed, M. O., Virani, A., Schmidt, M., Konečný, J., and Sallinen, S. (2015). Stopwasting my gradients: Practical svrg. In *Advances in Neural Information Processing Systems*, pages 2251–2259.

Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M., et al. (2017). Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323.

References (cont.)

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

OpenAI (2018). Openai five.

Peters, J. and Schaal, S. (2008). Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. (2016). Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Xu, T., Liu, Q., and Peng, J. (2017). Stochastic variance reduction for policy gradient estimation. *CoRR*, abs/1710.06034.

For $s = 1, \dots$

Sample N trajectories using $\widetilde{\theta}$

Compute FG $= \widehat{\nabla}_N J(\widetilde{\theta})$

For $t = 1, \dots, m$

Sample B trajectories using θ

Compute SG $= \widehat{\nabla}_B J(\theta)$

Compute correction $= \omega(\theta, \widetilde{\theta}) \widehat{\nabla}_B J(\widetilde{\theta})$

Update $\theta \leftarrow \theta + \alpha \blacktriangledown J(\theta)$

iteration

Update $\widetilde{\theta} \leftarrow \theta$

epoch

Adaptive Step Size

ADAM (Kingma and Ba, 2014):

- adapts to gradient variance
- can manage different batch sizes
- **has memory of past gradients (momentum)**

Problem: FG and SG have very different variance magnitudes
$\implies$ spurious momentum

We use two *separate* annealing schedules:

$$\widetilde{\boldsymbol{\theta}} \leftarrow \widetilde{\boldsymbol{\theta}} + \alpha_{FG} \widehat{\nabla}_N J(\widetilde{\boldsymbol{\theta}}) \quad \text{at the snapshot}$$
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{SG} \blacktriangledown J(\boldsymbol{\theta}) \quad \text{otherwise}$$

Note that $\widehat{\nabla}_N J(\widetilde{\boldsymbol{\theta}}) \equiv \blacktriangledown J(\boldsymbol{\theta})$ at the snapshot

Adaptive Epoch Size

Epoch size (m) trade-off:

- Large $m \implies$ large importance-weighting variance $\implies$ unstable
- Small $m \implies$ frequent snapshots $\implies$ data-inefficient

Idea: ADAM already relates gradient variance and efficiency

Our stopping criterion:

$$\frac{\alpha_{SG}}{B} < \frac{\alpha_{FG}}{N} \implies \textbf{snapshot}$$

When going on is not *convenient*, take new snapshot

Normalized Importance Weighting

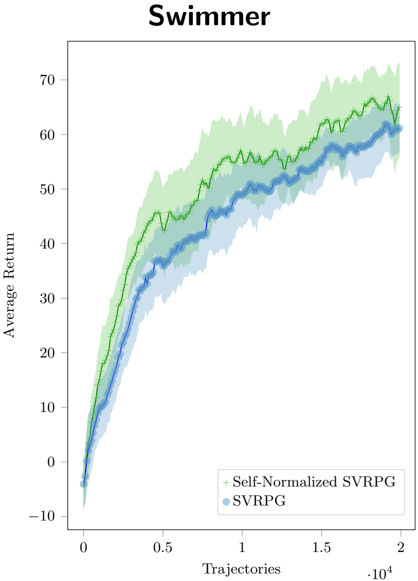
Regular importance weighting (unbiased):

$$\omega(\boldsymbol{\theta},\widetilde{\boldsymbol{\theta}})\widehat{\nabla}_B J(\widetilde{\boldsymbol{\theta}}) = \frac{1}{B}\sum_{i=1}^{B}\frac{p(\tau_i|\widetilde{\boldsymbol{\theta}})}{p(\tau_i|\boldsymbol{\theta})}\nabla\log p(\tau_i|\widetilde{\boldsymbol{\theta}})R(\tau_i)$$

Normalized importance weighting:

$$\omega(\boldsymbol{\theta},\widetilde{\boldsymbol{\theta}})\widehat{\nabla}_B J(\widetilde{\boldsymbol{\theta}}) = \frac{\sum_{i=1}^{B}\frac{p(\tau_i|\widetilde{\boldsymbol{\theta}})}{p(\tau_i|\boldsymbol{\theta})}\nabla\log p(\tau_i|\widetilde{\boldsymbol{\theta}})R(\tau_i)}{\sum_{i=1}^{B}\frac{p(\tau_i|\widetilde{\boldsymbol{\theta}})}{p(\tau_i|\boldsymbol{\theta})}}$$

- Less variance at the price of small bias
- Only affects the correction term
- Benefits are task-dependent



Actor-Critic SVRPG

Preliminary results

Critic (or *baseline*): an orthogonal variance-reduction technique

Gradient sample: $\sum_{t=1}^{H}\left(\sum_{k=1}^{t}\nabla\log\pi_{\boldsymbol{\theta}}(a_t|s_t)\right)(\gamma^t r_t - \underbrace{\mathbf{b}}_{\text{baseline}})$ (Peters and Schaal, 2008)

Not trivial to combine SVRG with critic: variance reduction is not additive

We combine SVRG with a simple critic from Duan et al. (2016)

Future work: ad hoc critic

The Full Story

- For **Swimmer**, we employ normalized weights in our final result
- For **Half-Cheetah**, we employ normaized weights *and* critic in our final result
- We compare **SVRPG** with GPOMDP (Baxter and Bartlett, 2001) with batch size $B = 10$
- This shows the *advantage* of correcting SG with more data
- However, GPOMDP with batch size $N = 100$ is even worse

Half-Cheetah