**Problem**

- Monotonically improve a parametric gaussian policy \( \pi_\theta \) in a continuous MDP, avoiding unsafe oscillations in the expected performance \( J(\theta) \).
- Episodic Policy Gradient:
  - estimate \( \hat{\nabla}_\theta J(\theta) \) from a batch of \( N \) sample trajectories.
  - \( \theta' \leftarrow \theta + \lambda \hat{\nabla}_\theta J(\theta) \)
- Tune step size \( \alpha \) and batch size \( N \) to limit oscillations. Not trivial:
  - \( \lambda \): trade-off with speed of convergence \( \rightarrow \) adaptive methods.
  - \( N \): trade-off with total learning time \( \leftarrow \) typically tuned by hand.
- Lack of cost sensitive solutions.

**Contributions**

1. We propose a per-component adaptive step size \( \Lambda \) which results in a greedy coordinate descent algorithm, improving over existing safe adaptive step-size methods.
2. We show a duality in the role played by \( \Lambda \) and \( N \) in maximizing the performance improvement \( J(\theta') - J(\theta) \) and how a joint optimization of the two meta-parameters can guarantee monotonic improvement with high probability.
3. We make a first step in the development of practical methods to jointly optimize the step size and the batch size.
4. We offer a preliminary empirical evaluation of the proposed methods on a simple control problem.

**Non-Scalar Adaptive Step Size**

**LOWER bound to policy performance:** [Pirotta et al., 2013]

\[
J(\theta') - J(\theta) \geq \frac{1}{1 - \gamma} \int_s \frac{d^\omega(s)}{d\pi^\omega(s)} \int_s (\pi_\theta(a|s) - \pi_{\theta'}(a|s)) Q^\pi(s, a) d\omega(s) \]  
\[
- \frac{\gamma}{2(1 - \gamma)} \| \pi_\theta - \pi_{\theta'} \|_2^2 \| Q^\pi \|_\infty = B_2(\theta', \theta)
\]

**Solution: coordinate ascent**

**Exact Framework**

Optimal step size:

\[
\alpha_* = \begin{cases} 
\frac{1}{\gamma} & \text{if } k = \min \left\{ \arg \max_i \left( \nabla \omega_i, J_\theta(\theta) \right) \right\} \\
0 & \text{otherwise}
\end{cases}
\]

\[
c = \frac{RM}{1 - \gamma} \left( \frac{L}{2\delta^2} + \frac{\gamma}{2(1 - \gamma)} \right)
\]

Improvement guarantee: \( J(\theta') - J(\theta) \geq (4c)^{-1} \| \nabla J_\theta(\theta) \|_\infty^2 \)

**Approximate Framework**

Given a policy gradient estimate \( \hat{\nabla}_\theta J_\theta(\theta) \) s.t. \( P \left( \| \hat{\nabla}_\theta J_\theta(\theta) - \nabla_\theta J_\theta(\theta) \|_\infty \leq \epsilon_i(N) \right) \leq \delta \)

Optimal step size:

\[
\alpha_* = \begin{cases} 
\frac{\left( \| \hat{\nabla}_\theta J_\theta(\theta) \|_\infty - \epsilon \right)^2}{2c} & \text{if } k = \min \left\{ \arg \max_i \left( \nabla \omega_i, J_\theta(\theta) \right) \right\} \\
0 & \text{otherwise}
\end{cases}
\]

Improvement guarantee: \( J(\theta') - J(\theta) \geq \left( \| \hat{\nabla}_\theta J_\theta(\theta) \|_\infty - \epsilon \right)^2 (4c)^{-1} \| \nabla J_\theta(\theta) \|_\infty^2 \)

**Adaptive Batch Size**

**IDEA:** There are evidences that it is possible to adapt the batch size instead of the step length [Pirotta and Restelli, 2016, Bollapragada et al., 2017, Smith et al., 2017]

- In particular, in RL the cost of collecting new samples may be huge
- Small step size \( \rightarrow \) lot of parameter update \( \rightarrow \) high costs

Cost-sensitive joint optimization

\[
\{ \Lambda^*, N^* \} = \arg \max_{\Lambda, N} \frac{B_1(\Lambda, N)}{N}
\]

**CHERYSHEV-LIKE BOUNDS**

Error bound: \( \epsilon \leq \frac{d_\theta}{\sqrt{N}} \) with probability \( 1 - \delta \)

Optimal meta-parameters:

\[
\alpha_* = \begin{cases} 
\frac{(13 - 3\sqrt{17})d_\theta}{4c} & \text{if } k = \min \left\{ \arg \max_i \left( \nabla \omega_i, J_\theta(\theta) \right) \right\} \\
0 & \text{otherwise}
\end{cases}
\]

\[
N^* = \left[ \frac{(13 + 3\sqrt{17})d_\theta^2}{2 \| \nabla J_\theta(\theta) \|_\infty^2} \right]
\]

**Empirical Results (One-Dimensional LQG)**

Comparing gradient estimation algorithms and values of \( \delta \)

Comparing statistical bounds using G(P)MDP and \( \delta = 0.95 \)

**References**

- S. L. Smith, P. J. Kindermans, and Q. V. Le. Don’t drop the learning rate, increase the batch size. ArXiv pre-print, 2017.