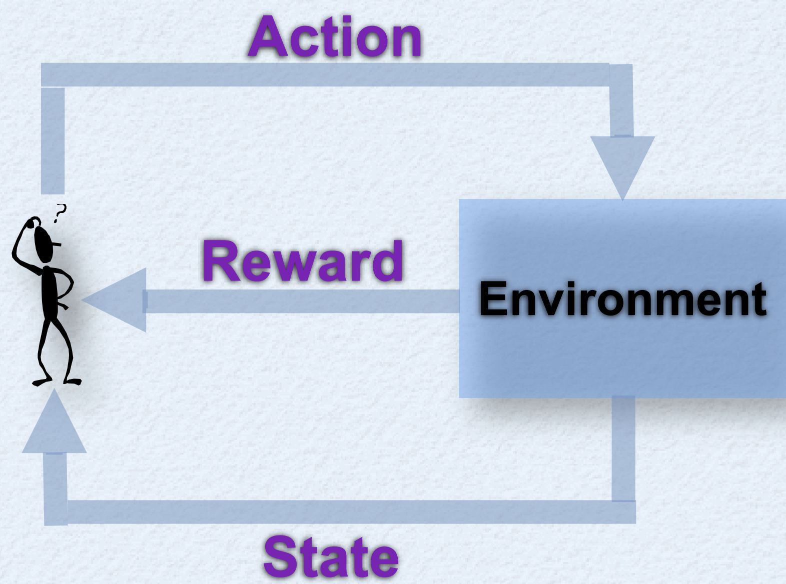
Bayes RL

Bayesian Policy Gradient Algorithms

Reinforcement Learning



- **RL:** A class of learning problems in which an agent interacts with an unfamiliar, dynamic and stochastic environment
- **Goal:** Learn a policy to maximize some measure of long-term reward
- **Interaction:** Modeled as a MDP or a POMDP

Markov Decision Process (MDP)

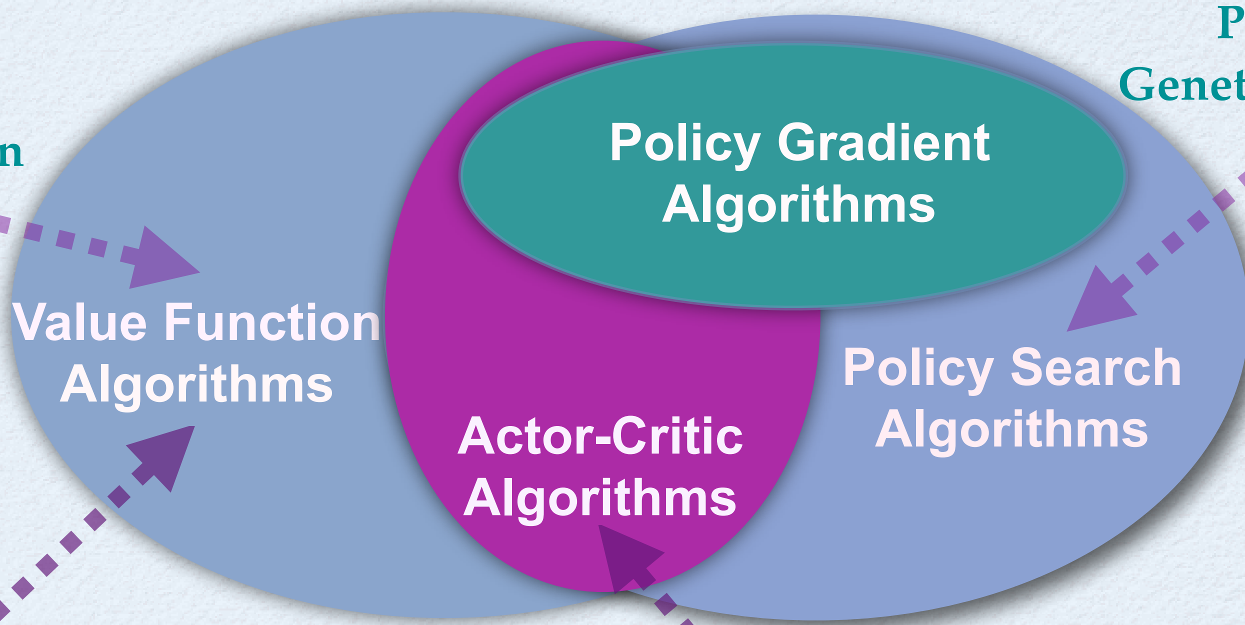
- **An MDP is defined as a 5-tuple** $(\mathcal{X}, \mathcal{A}, p, q, p_0)$
  - $\mathcal{X}$: State space of the process
  - $\mathcal{A}$: Action space of the process
  - $p(\cdot|x, a)$: Probability distribution over next state $x_{t+1} \sim p(\cdot|x_t, a_t)$
  - $q(\cdot|x, a)$: Probability distribution over rewards $R(x_t, a_t) \sim q(\cdot|x_t, a_t)$
  - p_0 : Initial state distribution
- **State-action space:** $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$, $z = (x, a)$
- **Policy:** Mapping from states to actions or distributions over actions

$$\mu(x) \in \mathcal{A} \qquad \text{or} \qquad \mu(\cdot|x) \in \Pr(\mathcal{A})$$

Reinforcement Learning Solutions

SARSA
Q-learning
Value Iteration

PEGASUS
Genetic Algorithms



Sutton, et al. 2000
Konda & Tsitsiklis 2000
Peters, et al. 2005
Bhatnagar, Ghavamzadeh, Sutton 2007

$$V^{\mu}(x) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) | x_0 = x\right]$$

$$Q^{\mu}(x,a) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) | x_0 = x, a_0 = a\right]$$

Policy Performance

- **System Path:** $$\xi = (x_0, a_0, x_1, a_1, \ldots, x_{T-1}, a_{T-1}, x_T)$$
- **Probability of a Path:** $$\Pr(\xi|\mu) = p_0(x_0) \prod_{t=0}^{T-1} \mu(a_t|x_t) p(x_{t+1}|x_t, a_t)$$
- **Return of a Path:** $$D(\xi) = \sum_{t=0}^{T-1} \gamma^t R(x_t, a_t)$$
- **Expected Return:** $$\eta(\mu) = \mathbf{E}[D(\xi)] = \int \bar{D}(\xi) \Pr(\xi|\mu) d\xi$$
- **Expected Return:** $$\eta(\mu) = \int \pi(x, a; \mu) \bar{R}(x, a) dx da$$

Bayes RL

Policy Gradient Methods

- **Policy Gradient (PG) Methods**
  - Define a class of smoothly parameterized stochastic policies

$$\{\mu(.|x;\boldsymbol{\theta}), x \in \mathcal{X}, \boldsymbol{\theta} \in \Theta\}$$

  - Estimate the gradient of the expected return w.r.t. policy parameters

$$\{\xi_1, \xi_2, \ldots, \xi_M\} \longrightarrow \nabla\eta(\boldsymbol{\theta})$$

  - Improve the policy by adjusting its parameters in the gradient direction

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\nabla\eta(\boldsymbol{\theta})$$

Bayes RL

Gradient Estimation

- **Expected Return** $\eta(\boldsymbol{\theta}) = \eta(\mu(\cdot|\cdot;\boldsymbol{\theta})) = \int \bar{D}(\xi)\Pr(\xi;\boldsymbol{\theta})d\xi$

- **Score Function or Likelihood Ratio Method**

$$\nabla\eta(\boldsymbol{\theta}) = \int \bar{D}(\xi)\frac{\nabla\Pr(\xi;\boldsymbol{\theta})}{\Pr(\xi;\boldsymbol{\theta})}\Pr(\xi;\boldsymbol{\theta})d\xi$$

- **Score Function**

$$\boldsymbol{u}(\xi) = \frac{\nabla\Pr(\xi;\boldsymbol{\theta})}{\Pr(\xi;\boldsymbol{\theta})} = \nabla\log\Pr(\xi;\boldsymbol{\theta}) = \sum_{t=0}^{T-1}\nabla\log\mu(a_t|x_t;\boldsymbol{\theta})$$

- **Monte-Carlo (MC) Estimation**

$$\nabla\hat{\eta}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M}D(\xi_i)\sum_{t=0}^{T_i-1}\nabla\log\mu(a_{t,i}|x_{t,i};\boldsymbol{\theta})$$

Shortcomings of Policy Gradient Methods

- **Examples of PG Algorithms**
  - Class of REINFORCE algorithms **(Williams 1992)**
  - Extending to infinite-horizon MDPs and POMDPs **(Kimura et al. 1995, Marbach 1998, Baxter & Bartlett 2001)**
- **Shortcomings of PG Algorithms**
  - MC estimates of the gradient have high variance
    - Require excessive number of samples
    - Slow convergence
  - Inefficient use of data

Improving Policy Gradient Algorithms

- **Speeding up the PG Algorithms**
  - Using discount factor **(Marbach 1998, Baxter & Bartlett 2001)**
  - Using a baseline **(Williams 1992, Sutton et al. 2000)**
  - Natural Gradient **(Kakade 2002, Bagnell & Schneider 2003, Peters et al. 2003)**

- **Contributions of this Work**
  - A Bayesian framework for policy gradient
    - Lower variance - Less samples - Faster convergence
    - Covariance of estimate is provided at little extra cost

Bayesian Quadrature

(O'Hagan 1991)

- **Integral Evaluation** $$\rho = \int F(x)p(x)dx$$
- **MC Estimate** $$\hat{\rho}_{MC} = \frac{1}{M}\sum_{i=1}^{M} F(x_i)$$
- **Bayesian Quadrature**
  - Model F as a Gaussian Process (GP) $$F(\cdot) \sim \mathcal{N}\{f_0(\cdot), k(\cdot,\cdot)\}$$

$$\mathbf{E}[F(x)] = f_0(x) \quad , \quad \mathbf{Cov}[F(x), F(x')] = k(x, x')$$

  - A set of samples is observed $$\mathcal{D}_M = \{(x_i, y_i)\}_{i=1}^{M}$$

Bayes RL

Bayesian Quadrature

(O'Hagan 1991)

- **Bayesian Quadrature**
  - Posterior mean and covariance of f are computed

$$\mathbf{E}[F(x)|\mathcal{D}_M] = f_0(x) + \boldsymbol{k}_M(x)^\top \boldsymbol{C}_M(\boldsymbol{y}_M - \boldsymbol{f}_0)$$

$$\mathbf{Cov}[F(x), F(x')|\mathcal{D}_M] = k(x, x') - \boldsymbol{k}_M(x)^\top \boldsymbol{C}_M \boldsymbol{k}_M(x')$$

  - Posterior mean and variance of ρ are computed as

$$\mathbf{E}[\rho|\mathcal{D}_M] = \int \mathbf{E}[F(x)|\mathcal{D}_M]p(x)dx = \rho_0 + \boldsymbol{z}_M^\top \boldsymbol{C}_M(\boldsymbol{y}_M - \boldsymbol{f}_0)$$

$$\mathbf{Var}[\rho|\mathcal{D}_M] = \int\int \mathbf{Cov}[F(x), F(x')|\mathcal{D}_M]p(x)p(x')dxdx' = z_0 + \boldsymbol{z}_M^\top \boldsymbol{C}_M \boldsymbol{z}_M$$

$$\rho_0 = \int f_0(x)p(x)dx \quad , \quad \boldsymbol{z}_M = \int \boldsymbol{k}_M(x)p(x)dx \quad , \quad z_0 = \int\int k(x, x')p(x)p(x')dxdx'$$

Bayes RL

Bayesian Policy Gradient

(Ghavamzadeh & Engel, NIPS 2006)

- **Gradient of the performance measure**

$$\nabla\eta(\boldsymbol{\theta}) = \int \bar{D}(\xi)\nabla \log \Pr(\xi;\boldsymbol{\theta}) \; \Pr(\xi;\boldsymbol{\theta}) \; d\xi$$

$F(\xi;\boldsymbol{\theta})$ $\qquad$ $p(\xi;\boldsymbol{\theta})$

- **Model 1:**

$$\mathbf{E}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_M) = \boldsymbol{Y}_M \boldsymbol{C}_M \boldsymbol{z}_M \quad , \quad \mathbf{Cov}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_M) = (z_0 - \boldsymbol{z}_M^\top \boldsymbol{C}_M \boldsymbol{z}_M)\boldsymbol{I}$$

$$\boldsymbol{z}_M = \int \boldsymbol{k}_M(\xi) \Pr(\xi;\boldsymbol{\theta}) d\xi \quad , \quad z_0 = \int\int k(\xi,\xi') \Pr(\xi;\boldsymbol{\theta}) \Pr(\xi';\boldsymbol{\theta}) d\xi d\xi'$$

$$k(\xi_i,\xi_j) = (1 + \boldsymbol{u}(\xi_i)^\top \boldsymbol{G}^{-1}\boldsymbol{u}(\xi_j))^2 \longrightarrow \begin{cases} (\boldsymbol{z}_M)_i = 1 + \boldsymbol{u}(\xi_i)^\top \boldsymbol{G}^{-1}\boldsymbol{u}(\xi_j) \\ z_0 = 1 + n \end{cases}$$

Bayes RL

Bayesian Policy Gradient

(Ghavamzadeh & Engel, NIPS 2006)

- **Gradient of the performance measure**

$$\nabla \eta(\boldsymbol{\theta}) = \int \bar{D}(\xi) \, \nabla \log \Pr(\xi; \boldsymbol{\theta}) \Pr(\xi; \boldsymbol{\theta}) \, d\xi$$

$F(\xi; \boldsymbol{\theta})$ → $\bar{D}(\xi)$; $p(\xi; \boldsymbol{\theta})$ → $\nabla \log \Pr(\xi; \boldsymbol{\theta}) \Pr(\xi; \boldsymbol{\theta})$

- **Model 2:**

$$\mathbf{E}(\nabla \eta(\boldsymbol{\theta}) | \mathcal{D}_M) = \boldsymbol{Z}_M \boldsymbol{C}_M \boldsymbol{y}_M \quad , \quad \mathbf{Cov}(\nabla \eta(\boldsymbol{\theta}) | \mathcal{D}_M) = \boldsymbol{Z}_0 - \boldsymbol{Z}_M \boldsymbol{C}_M \boldsymbol{Z}_M^\top$$

$$\boldsymbol{Z}_M = \int \boldsymbol{k}_M(\xi)^\top \nabla \Pr(\xi; \boldsymbol{\theta}) d\xi \quad , \quad \boldsymbol{Z}_0 = \int\int k(\xi, \xi') \nabla \Pr(\xi; \boldsymbol{\theta}) \nabla \Pr(\xi'; \boldsymbol{\theta})^\top d\xi d\xi'$$

$$k(\xi_i, \xi_j) = \boldsymbol{u}(\xi_i)^\top \boldsymbol{G}^{-1} \boldsymbol{u}(\xi_j) \longrightarrow \begin{cases} \boldsymbol{Z}_M = \boldsymbol{U}_M = [\boldsymbol{u}(\xi_1), \ldots, \boldsymbol{u}(\xi_M)] \\ \boldsymbol{Z}_0 = \boldsymbol{G} - \boldsymbol{U}_M \boldsymbol{C}_M \boldsymbol{U}_M^\top \end{cases}$$

Bayesian Policy Gradient

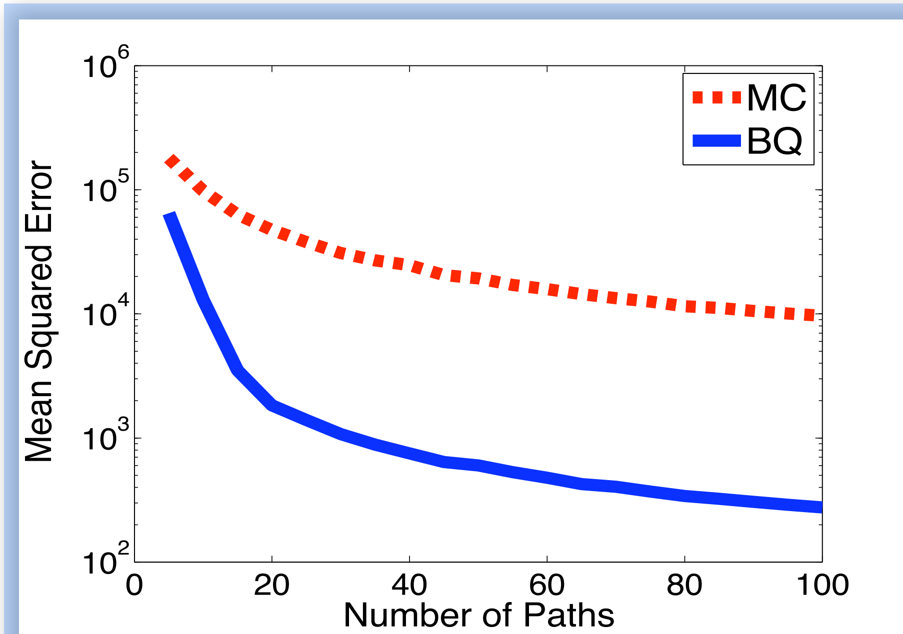
(Ghavamzadeh & Engel, NIPS 2006)

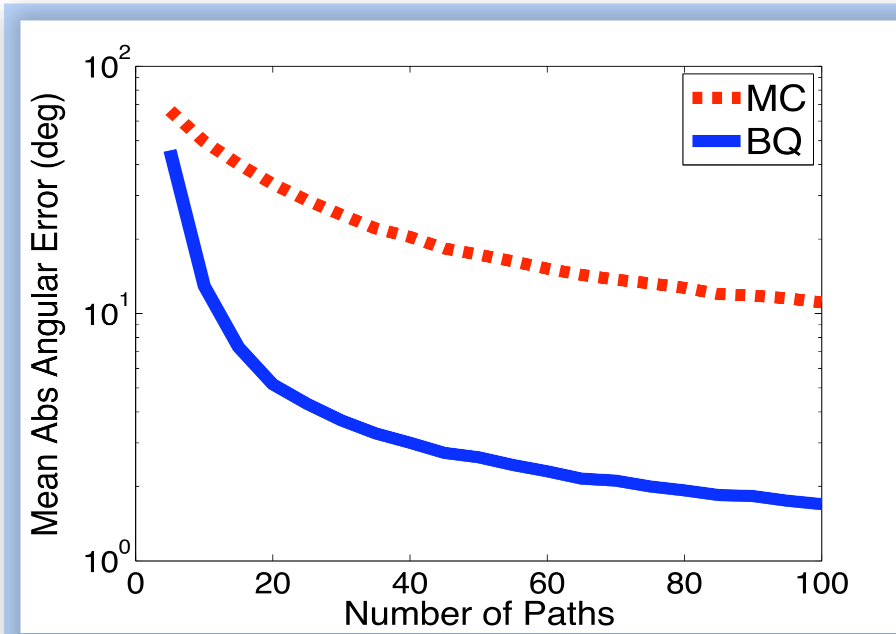
- **Online Sparsification** **(Engel et al. 2002)**
  - Selectively add a new observed path to the set of dictionary paths

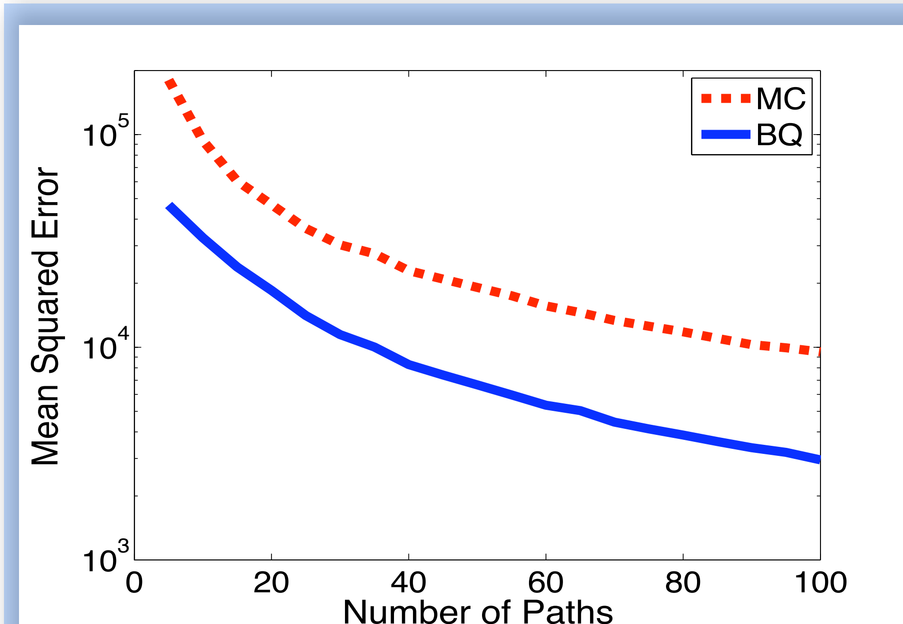
- **Fisher Information Matrix Estimation**
  - MC estimation $\hat{\boldsymbol{G}}_{MC}(\boldsymbol{\theta}) = \frac{1}{\sum_{i=1}^{M} T_i} \sum_{i=1}^{M} \sum_{t=0}^{T_i - 1} \nabla \log \mu(a_{t,i}|x_{t,i};\boldsymbol{\theta}) \nabla \log \mu(a_{t,i}|x_{t,i};\boldsymbol{\theta})^\top$
  - Model-based policy gradient
    - Parameterize the transition probability function
    - Estimate its parameters **(ML estimation)**

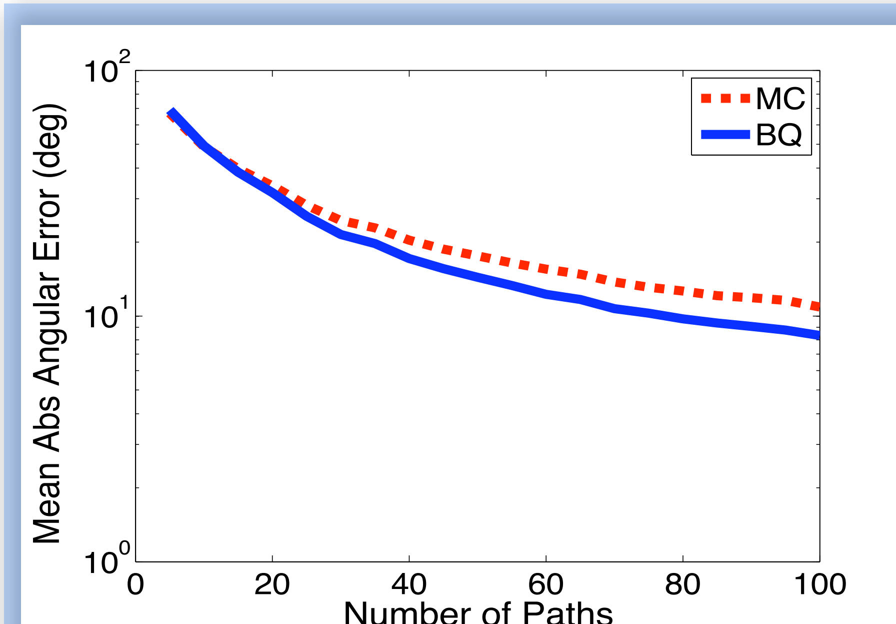
Linear Quadratic Regulator

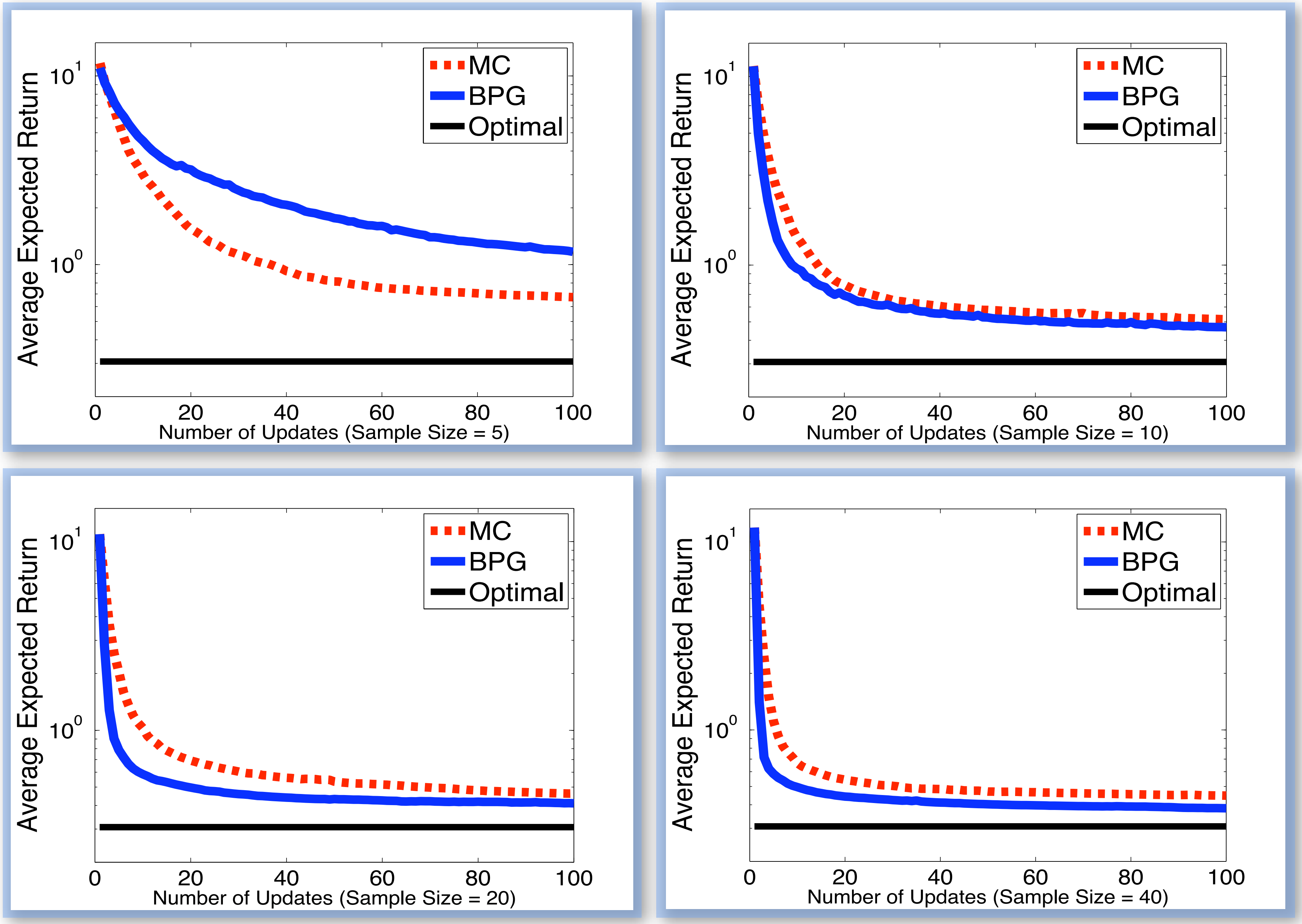
- **System**
  - Initial State $$x_0 \sim \mathcal{N}(0.3, 0.001)$$
  - State Transition $$x_{t+1} = x_t + a_t + n_x \quad , \quad n_x \sim \mathcal{N}(0, 0.01)$$
  - Reward $$R_t = x_t^2 + 0.1 a_t^2$$
- **Policy**
  - Actions $$a_t \sim \mu(.|x_t; \boldsymbol{\theta}) = \mathcal{N}(\lambda x_t, \sigma^2)$$
  - Parameters $$\boldsymbol{\theta} = (\lambda, \sigma)^\top$$

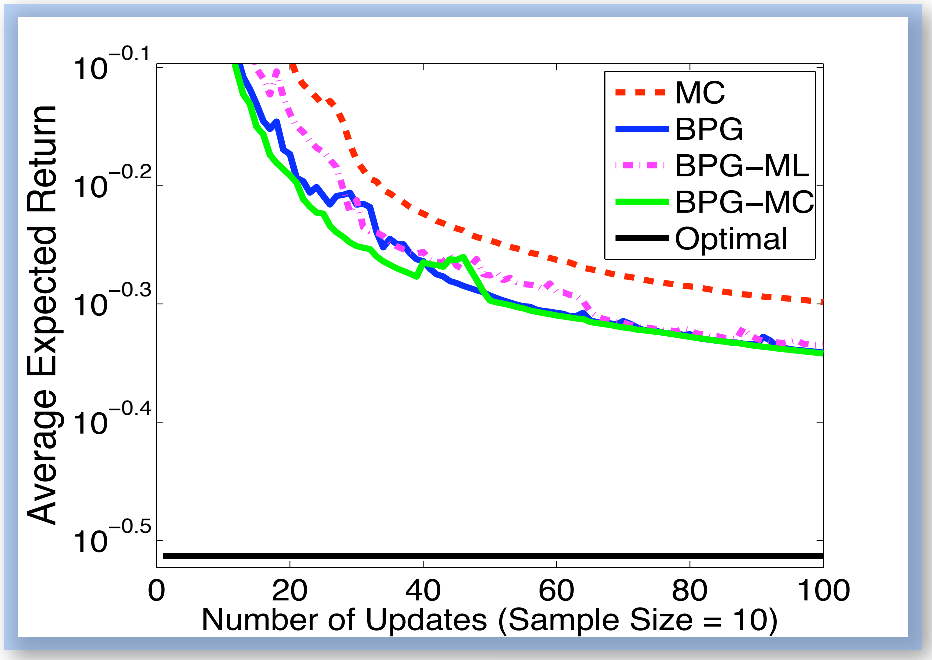

Mean Squared Error
Number of Paths
MC
BQ

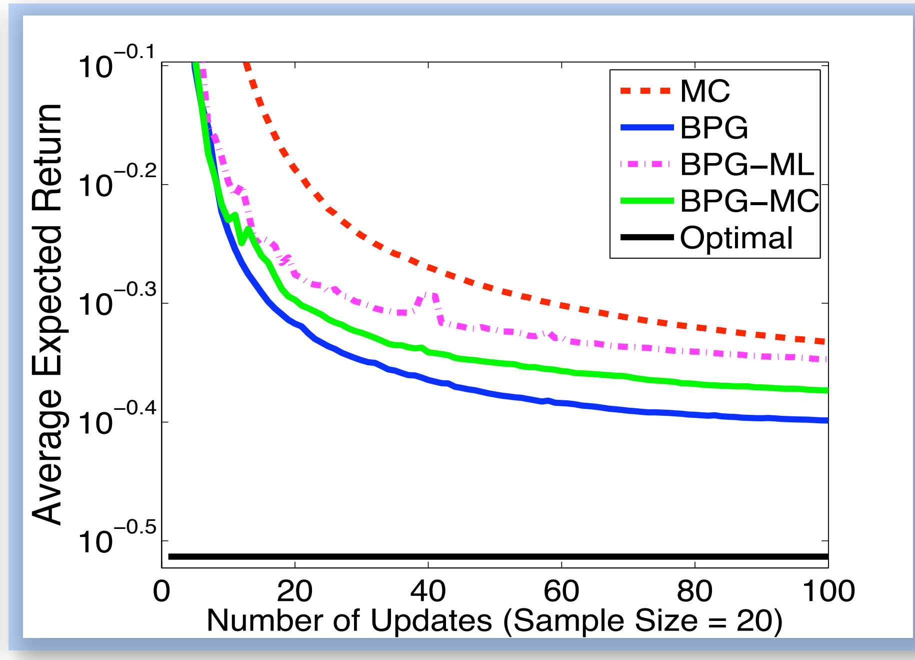

Mean Abs Angular Error (deg)
Number of Paths
MC
BQ

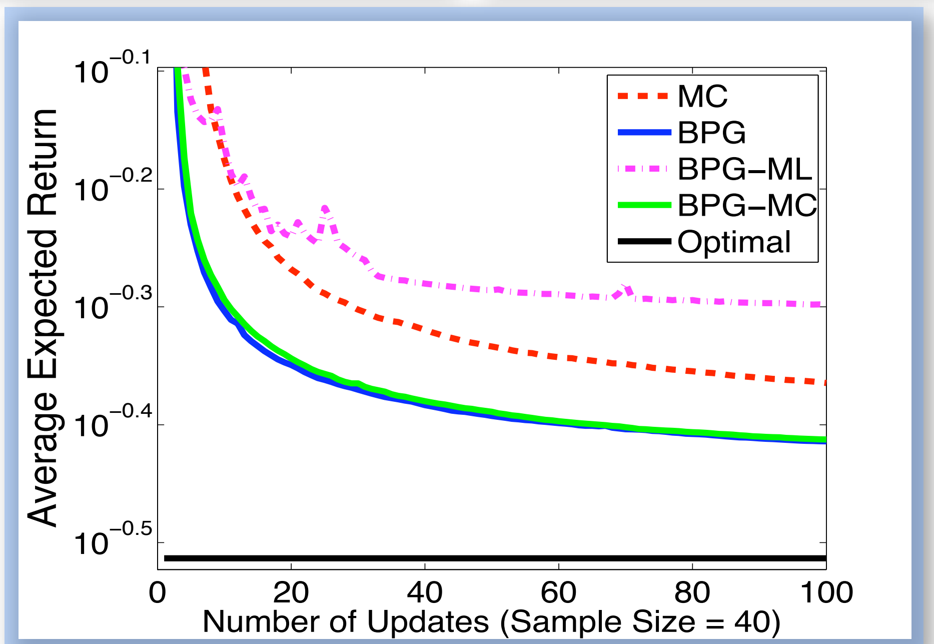

Mean Squared Error
Number of Paths
MC
BQ


Mean Abs Angular Error (deg)
Number of Paths
MC
BQ


MC
BPG
Optimal
Average Expected Return
10^1
10^0
0 20 40 60 80 100
Number of Updates (Sample Size = 5)
Number of Updates (Sample Size = 10)
Number of Updates (Sample Size = 20)
Number of Updates (Sample Size = 40)

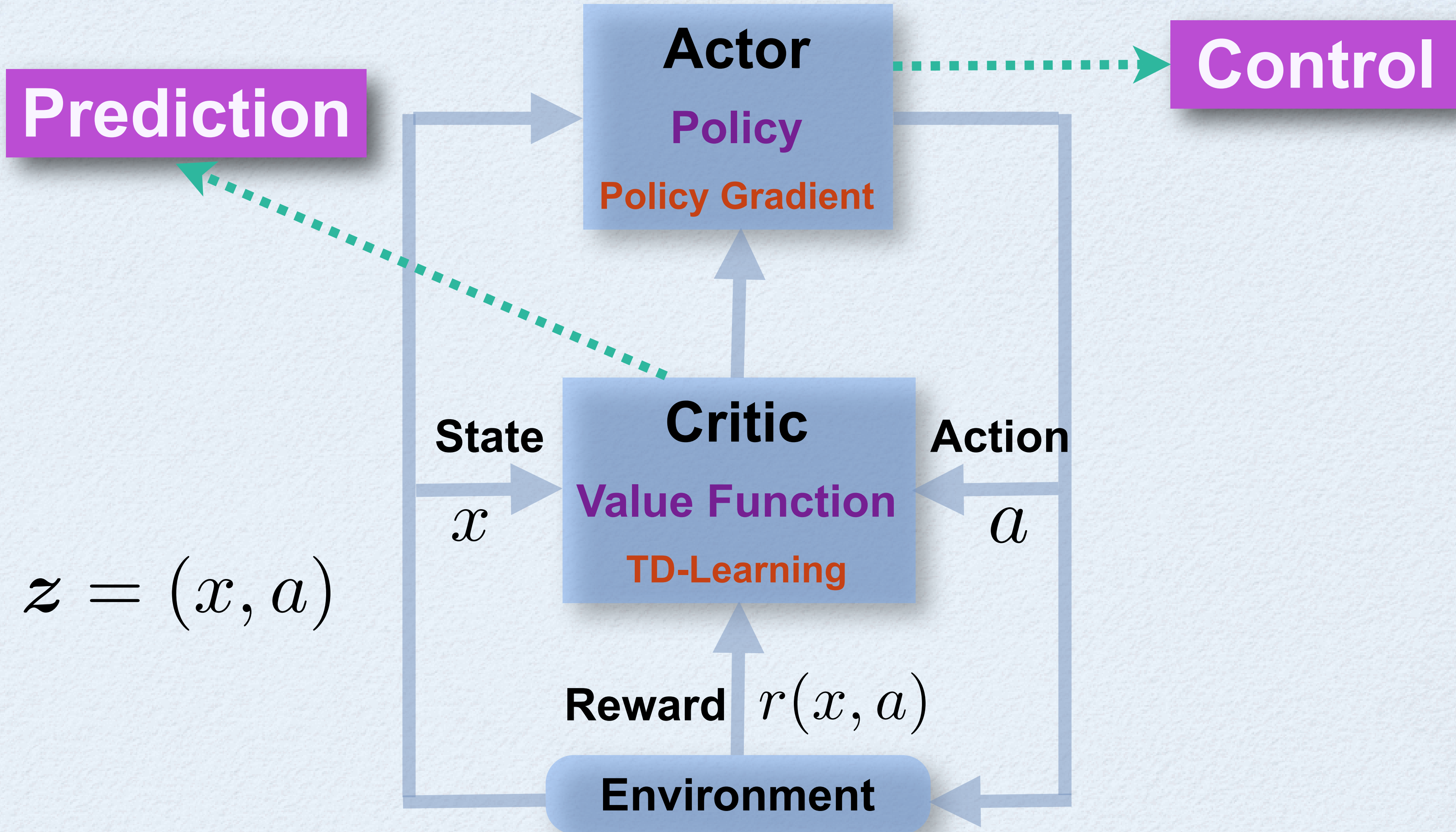

Average Expected Return
$10^{-0.1}$
$10^{-0.2}$
$10^{-0.3}$
$10^{-0.4}$
$10^{-0.5}$
0
20
40
60
80
100
Number of Updates (Sample Size = 10)
MC
BPG
BPG–ML
BPG–MC
Optimal


Average Expected Return
$10^{-0.1}$
$10^{-0.2}$
$10^{-0.3}$
$10^{-0.4}$
$10^{-0.5}$
0
20
40
60
80
100
Number of Updates (Sample Size = 20)
MC
BPG
BPG–ML
BPG–MC
Optimal


Average Expected Return
$10^{-0.1}$
$10^{-0.2}$
$10^{-0.3}$
$10^{-0.4}$
$10^{-0.5}$
0
20
40
60
80
100
Number of Updates (Sample Size = 40)
MC
BPG
BPG–ML
BPG–MC
Optimal

Bayesian Actor-Critic Algorithms

ACTOR-CRITIC



$$z = (x, a)$$

Bayesian Actor-Critic

(Ghavamzadeh & Engel, ICML 2007)

Actor

- Performance measure

$$\eta(\boldsymbol{\theta}) = \int \pi(\boldsymbol{z};\boldsymbol{\theta})\bar{R}(\boldsymbol{z})d\boldsymbol{z}$$

- Gradient of the performance measure

$$\nabla\eta(\boldsymbol{\theta}) = \int \pi(\boldsymbol{z};\boldsymbol{\theta})\ \nabla \log \mu(a|x;\boldsymbol{\theta})\ Q(\boldsymbol{z};\boldsymbol{\theta})\ d\boldsymbol{z}\ = \int \boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\ \underbrace{Q(\boldsymbol{z};\boldsymbol{\theta})}_{\text{GP}}\ d\boldsymbol{z}$$

- Posterior moments of gradient

$$\mathbf{E}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_t) = \int \boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\ \mathbf{E}(Q(\boldsymbol{z};\boldsymbol{\theta})|\mathcal{D}_t)\ d\boldsymbol{z}$$

$$\mathbf{Cov}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_t) = \int\int \boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\ \mathbf{Cov}(Q(\boldsymbol{z};\boldsymbol{\theta}), Q(\boldsymbol{z}';\boldsymbol{\theta})|\mathcal{D}_t)\ \boldsymbol{g}(\boldsymbol{z}';\boldsymbol{\theta})^\top\ d\boldsymbol{z}\ d\boldsymbol{z}'$$

Bayesian Actor-Critic

(Ghavamzadeh & Engel, ICML 2007)

Critic

- GPTD **(Engel, Mannor, & Meir 2003, 2005)**

$$\mathbf{E}(Q(\boldsymbol{z};\boldsymbol{\theta})|\mathcal{D}_t) = \boldsymbol{k}_t(\boldsymbol{z})^\top \boldsymbol{\alpha}_t$$

$$\mathbf{Cov}(Q(\boldsymbol{z};\boldsymbol{\theta}), Q(\boldsymbol{z}';\boldsymbol{\theta})|\mathcal{D}_t) = k(\boldsymbol{z},\boldsymbol{z}') - \boldsymbol{k}_t(\boldsymbol{z})^\top \boldsymbol{C}_t \boldsymbol{k}_t(\boldsymbol{z}')^\top$$

Actor

$$\mathbf{E}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_\mathbf{t}) = \boldsymbol{U}_\mathbf{t}\boldsymbol{\alpha}_\mathbf{t} \quad , \quad \mathbf{Cov}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_\mathbf{t}) = \boldsymbol{V} - \boldsymbol{U}_\mathbf{t}\boldsymbol{C}_\mathbf{t}\boldsymbol{U}_\mathbf{t}^\top$$

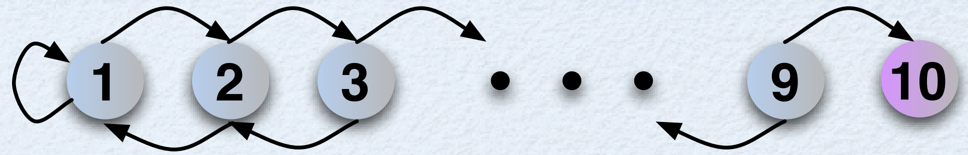
$$\boldsymbol{U}_t = \int \boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\boldsymbol{k}_t(\boldsymbol{z})^\top d\boldsymbol{z} \quad , \quad \boldsymbol{V} = \int\int \boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})k(\boldsymbol{z},\boldsymbol{z}')\boldsymbol{g}(\boldsymbol{z}';\boldsymbol{\theta})^\top d\boldsymbol{z}d\boldsymbol{z}'$$

$$k(\boldsymbol{z},\boldsymbol{z}') = k_x(x,x') + k_F(\boldsymbol{z},\boldsymbol{z}') \longrightarrow \begin{cases} \boldsymbol{U}_t = [\boldsymbol{u}(\boldsymbol{z}_0),\ldots,\boldsymbol{u}(\boldsymbol{z}_t)] \\ \boldsymbol{V} = \boldsymbol{G} \end{cases}$$

$$k_F(\boldsymbol{z},\boldsymbol{z}') = \boldsymbol{u}(\boldsymbol{z})^\top \boldsymbol{G}^{-1}\boldsymbol{u}(\boldsymbol{z}')$$

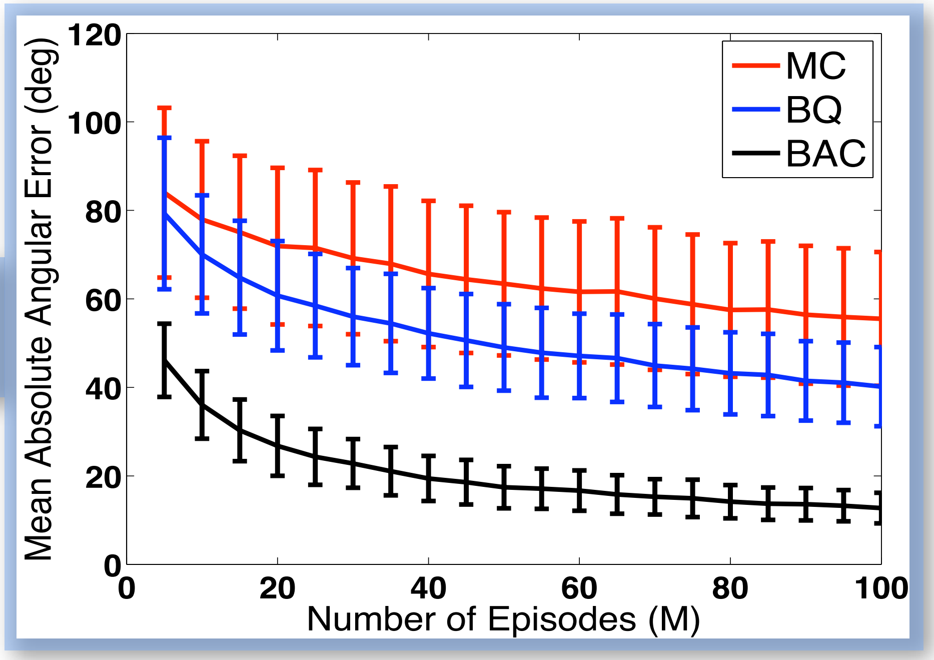
Random Walk Problem

- **System**

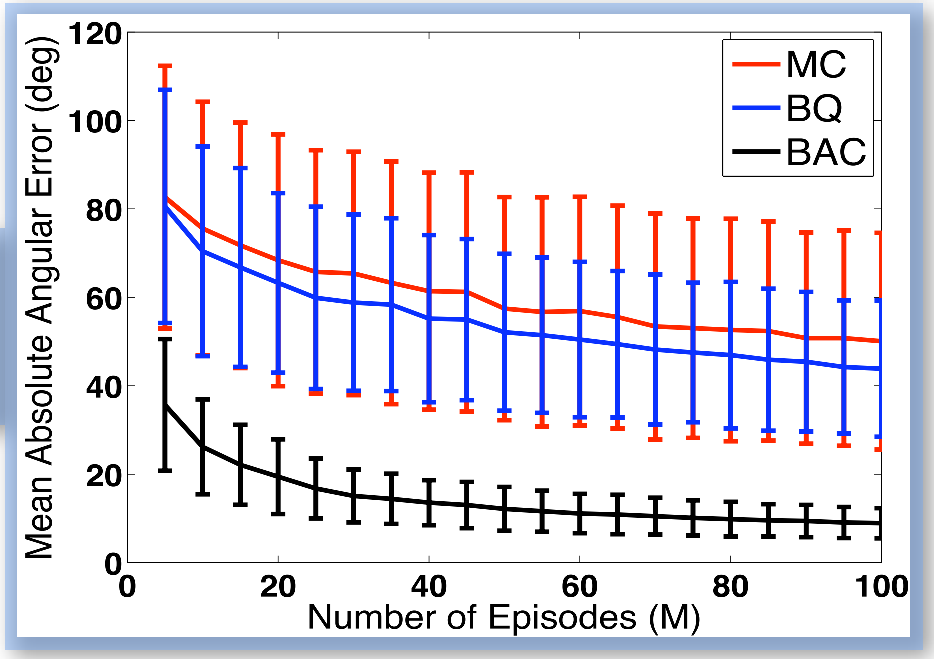


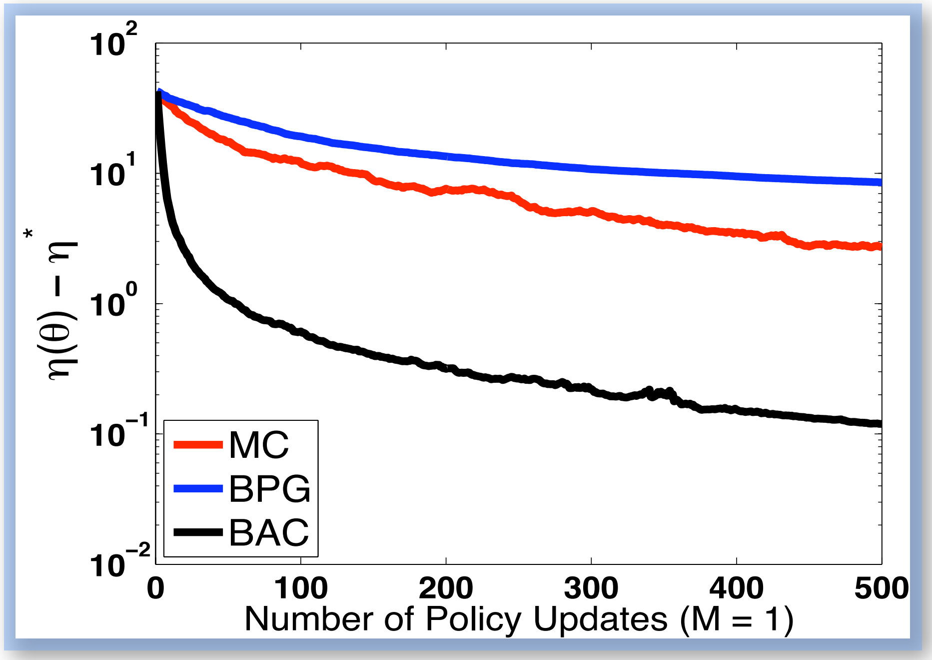
  - State Space $$\mathcal{X} = \{1, \ldots, 10\}$$
  - Action Space $$\mathcal{A} = \{\text{right}, \text{left}\}$$
  - Initial State - Terminal State $$x_0 = 1 \quad , \quad x_T = 10$$
  - Cost $$R(x) = \begin{cases} \mathcal{N}(1, 0.01) & x = 1, \ldots, 9 \\ 0 & x = 10 \end{cases}$$
- **Policy**
  - Actions $$\mu(\text{right}|x) = \frac{1}{1 + \exp(-\theta_x)}$$
  - Parameters $$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{10})^\top$$

One Policy

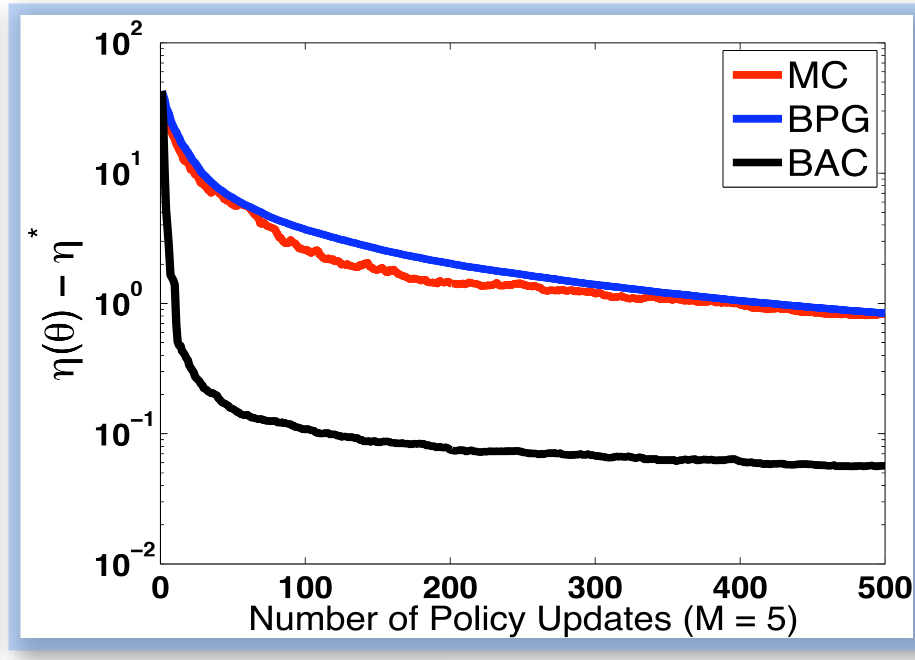


Multiple Policies

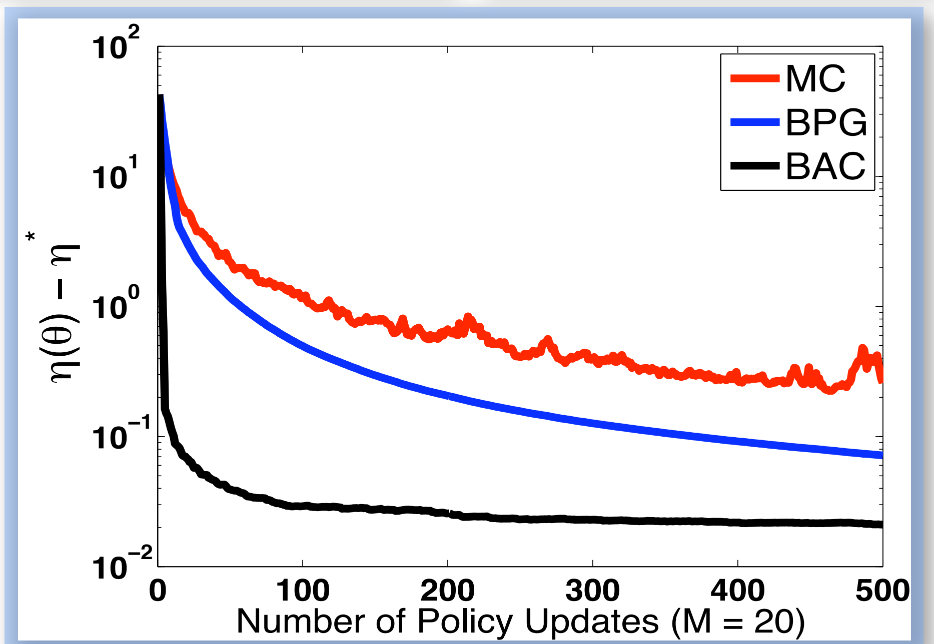




10^2
10^1
10^0
10^{-1}
10^{-2}
$\eta(\theta) - \eta^*$
0
100
200
300
400
500
Number of Policy Updates (M = 1)
MC
BPG
BAC



10^2
10^1
10^0
10^{-1}
10^{-2}
$\eta(\theta) - \eta^*$
0
100
200
300
400
500
Number of Policy Updates (M = 5)
MC
BPG
BAC



10^2
10^1
10^0
10^{-1}
10^{-2}
$\eta(\theta) - \eta^*$
0
100
200
300
400
500
Number of Policy Updates (M = 20)
MC
BPG
BAC

BPG & BAC Comparison

- **Bayesian Policy Gradient (BPG)** (Ghavamzadeh & Engel, 2006)
  - **Basic observable unit:** complete system trajectory
    - Allow handling non-Markovian systems **(e.g. partial observability, Markov games)**
- **Bayesian Actor-Critic (BAC)**
  - **Basic observable unit:** individual state-action-reward transitions **(Markov property)**
    - Reduce the variance of the gradient estimates
    - Allow handling systems with long and/or variable-length trajectories

SUMMARY

- An alternative approach **(Bayesian)** to conventional MC-based **(frequentist)** policy gradient estimation procedure
  - Less variance
  - Less number of samples
  - Faster convergence
  - Natural gradient and gradient covariance are provided at little extra cost
- GP to define a prior distribution over the gradient of the expected return
- Compute its posterior conditioned on the observed data

FUTURE WORK

- **Using gradient covariance**
  - Risk-aware selection of the update step-size and direction
  - Termination condition
- **Combining with MDP model estimation** **(Model-Based BAC Algorithms)**
  - Transfer of learning between different policies
  - More data efficient PG algorithms
  - More flexibility in kernel function selection
- **Non-parametric policies**
- **Second order updates - how to estimate the Hessian?**
- **More challenging problems** **(e.g. control of an octopus arm)**