Interpolation-based Q-learning

Csaba Szepesvári
MTA SZTAKI
William D. Smart
WUSTL

Summary

- Introduction
- Algorithm
- Convergence results
- Extensions
- Some experimental results
- Conclusions, future work

Motivation, problem setup

- Q-learning
- Continuous space Q-learning

Problem Setup

- Markovian Decision Problems

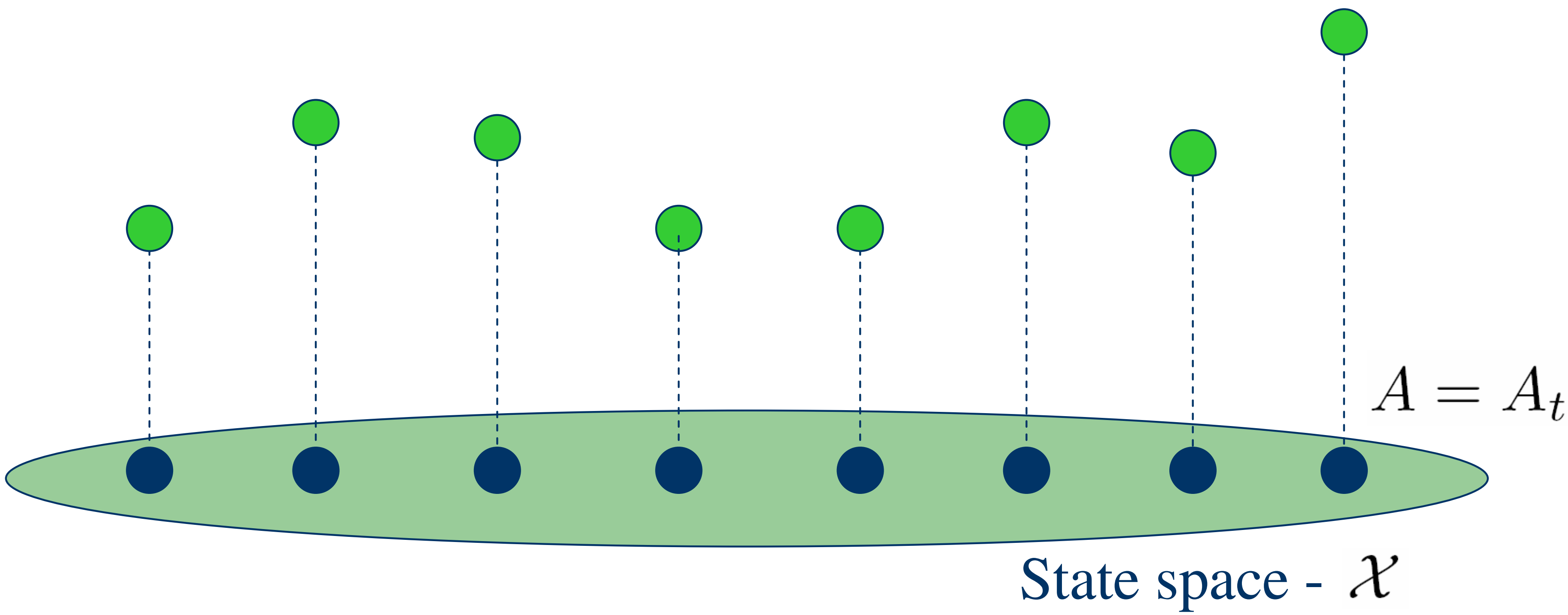
  X – state space

  A – action space

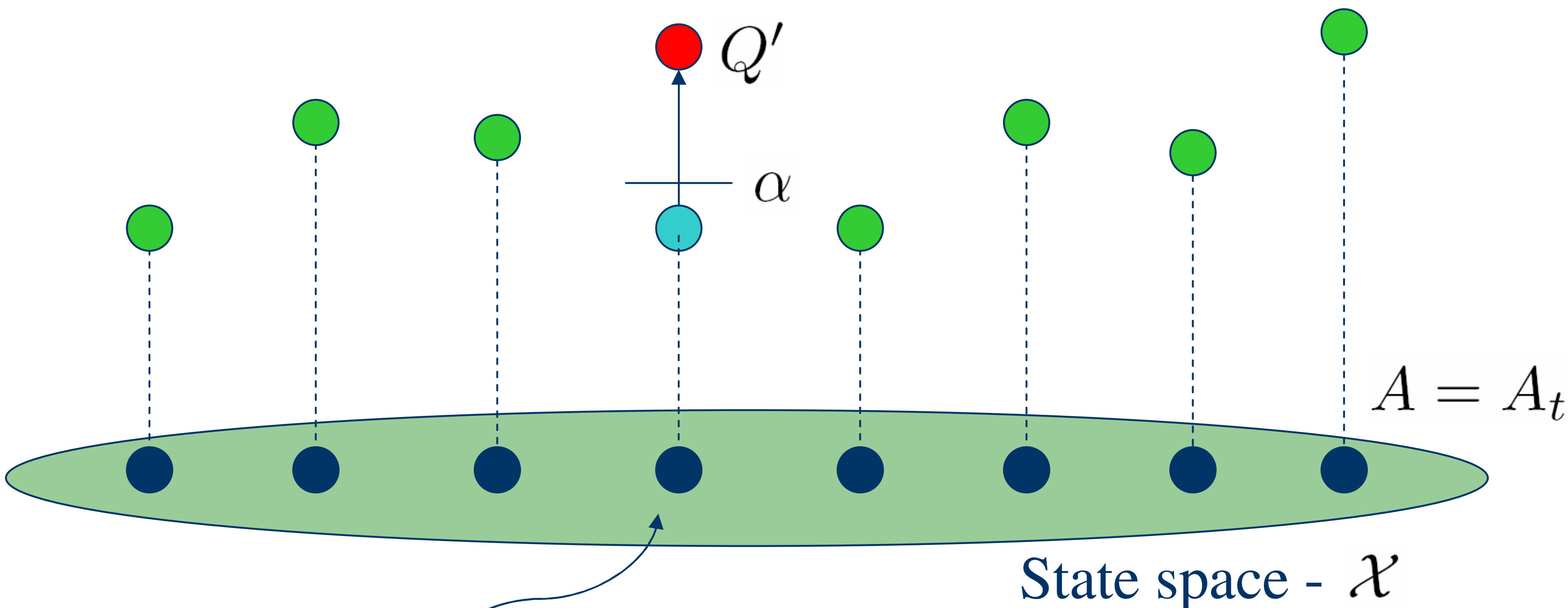
  p – transition kernel (dynamics)

  r – immediate rewards
- Continuous State Space
- Unknown Dynamics

Q-learning in Discrete State Spaces

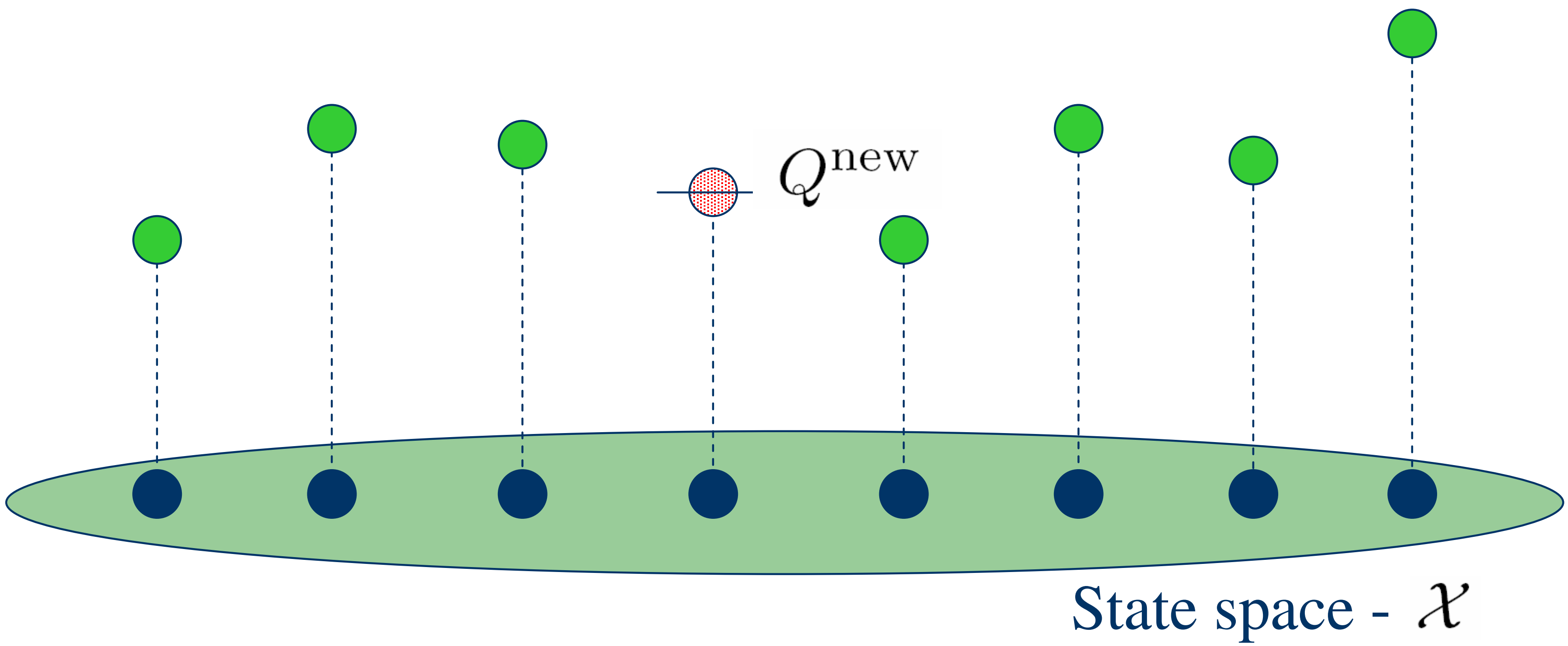


Q-learning in Discrete State Spaces

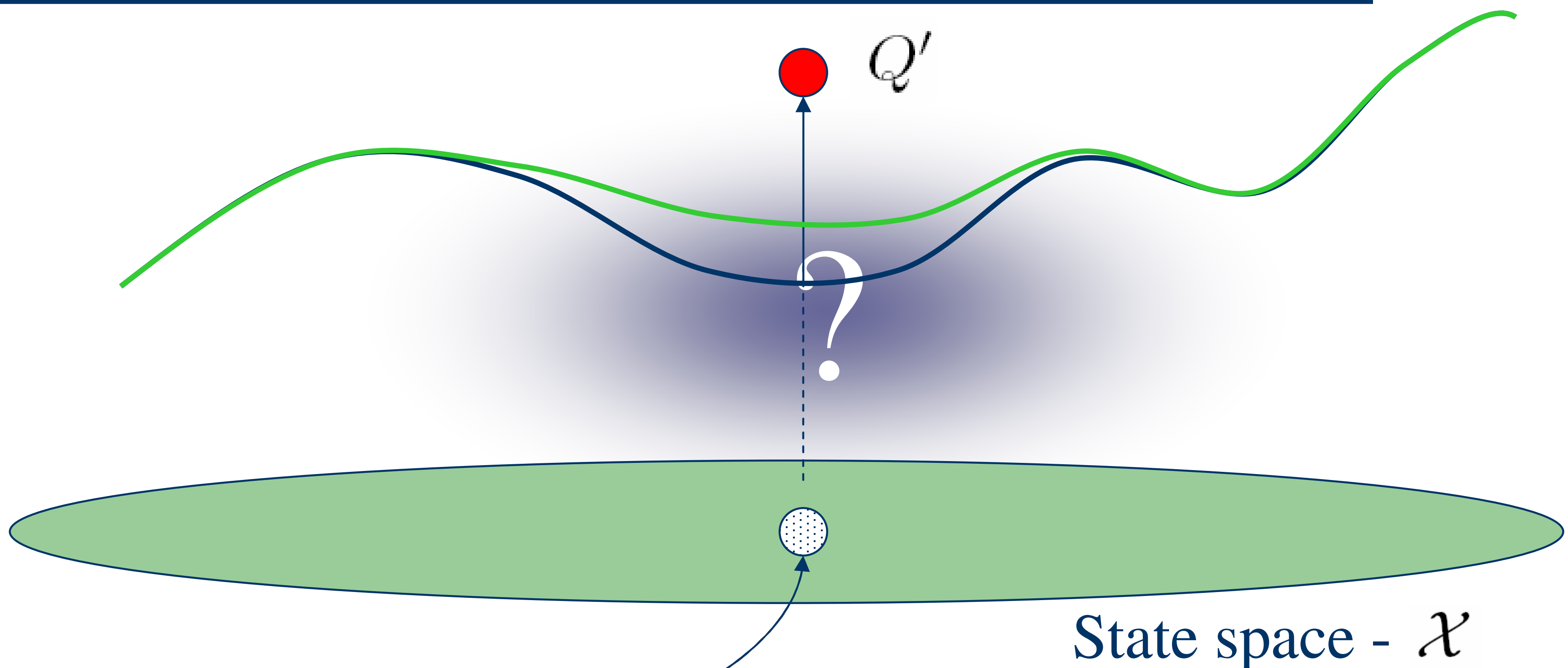


$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

Q-learning in Discrete State Spaces

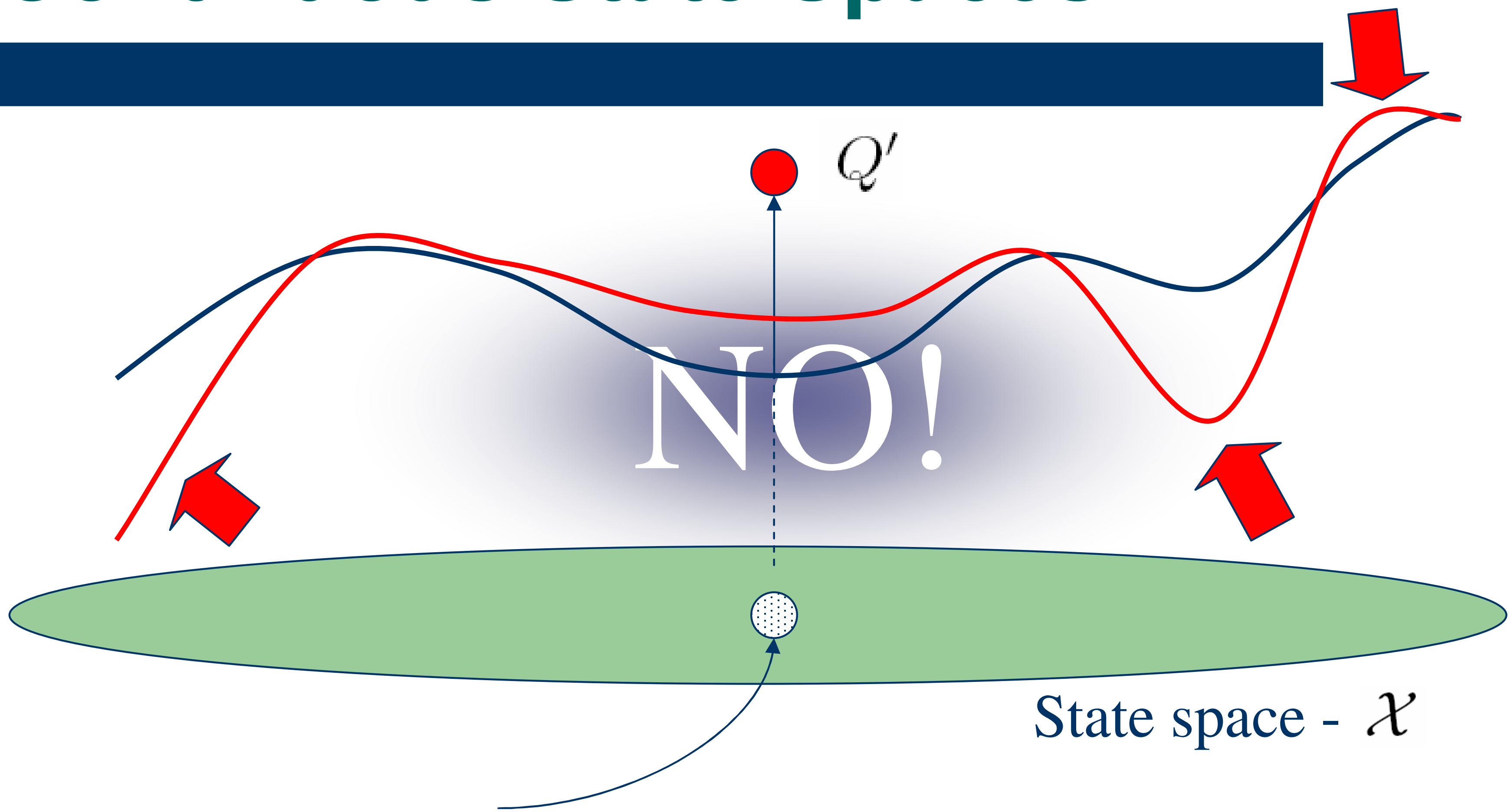


Continuous State Spaces



$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

Continuous State Spaces



$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

Difference to Regression

$$Q' \neq Q^* + \mathrm{Noise}$$

$$Q^* = \mathrm{T}Q^*$$

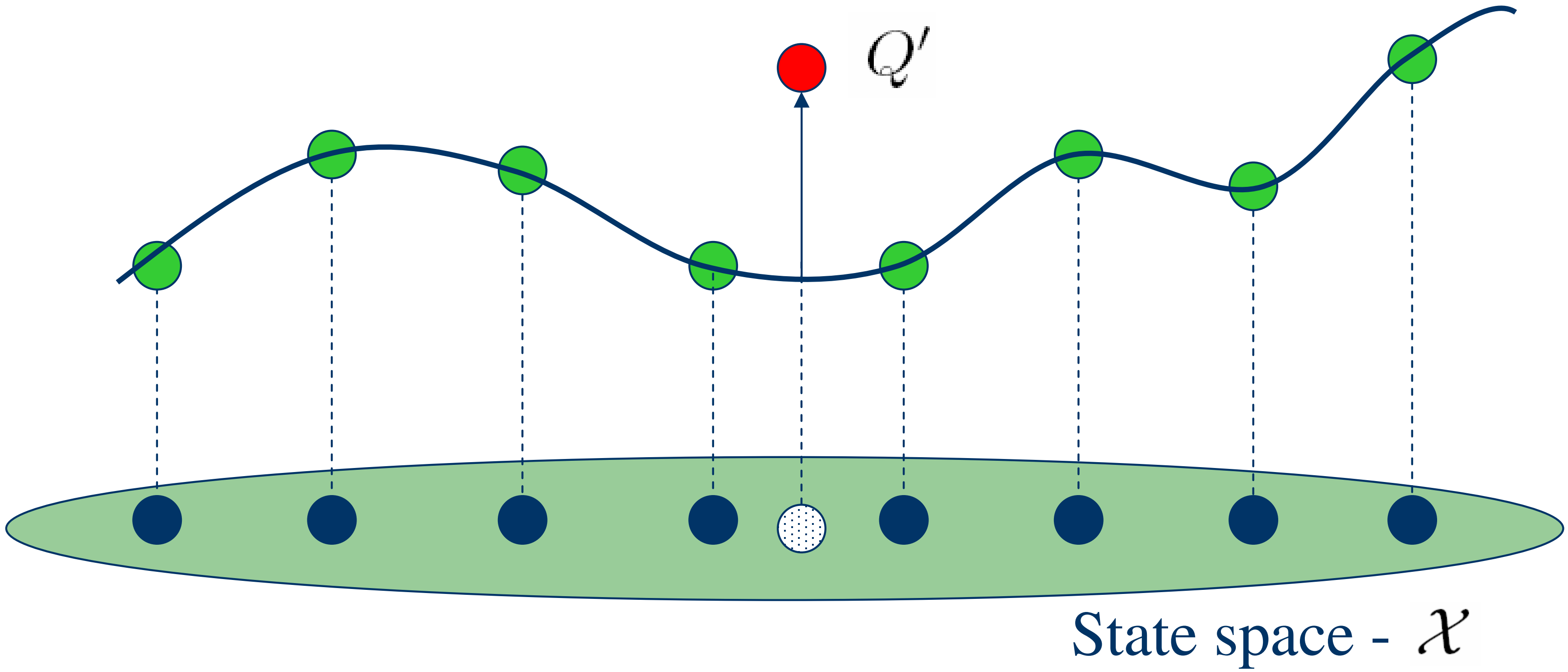
$$Q' = Q^* + (Q - Q^*) + \mathrm{Noise}$$

Goals

- Avoid non-convergence
  - Changes:
    - Local
    - Incremental
- O(1) memory requirements
- O(1) time update

Algorithm

iFAPP-Q Learning

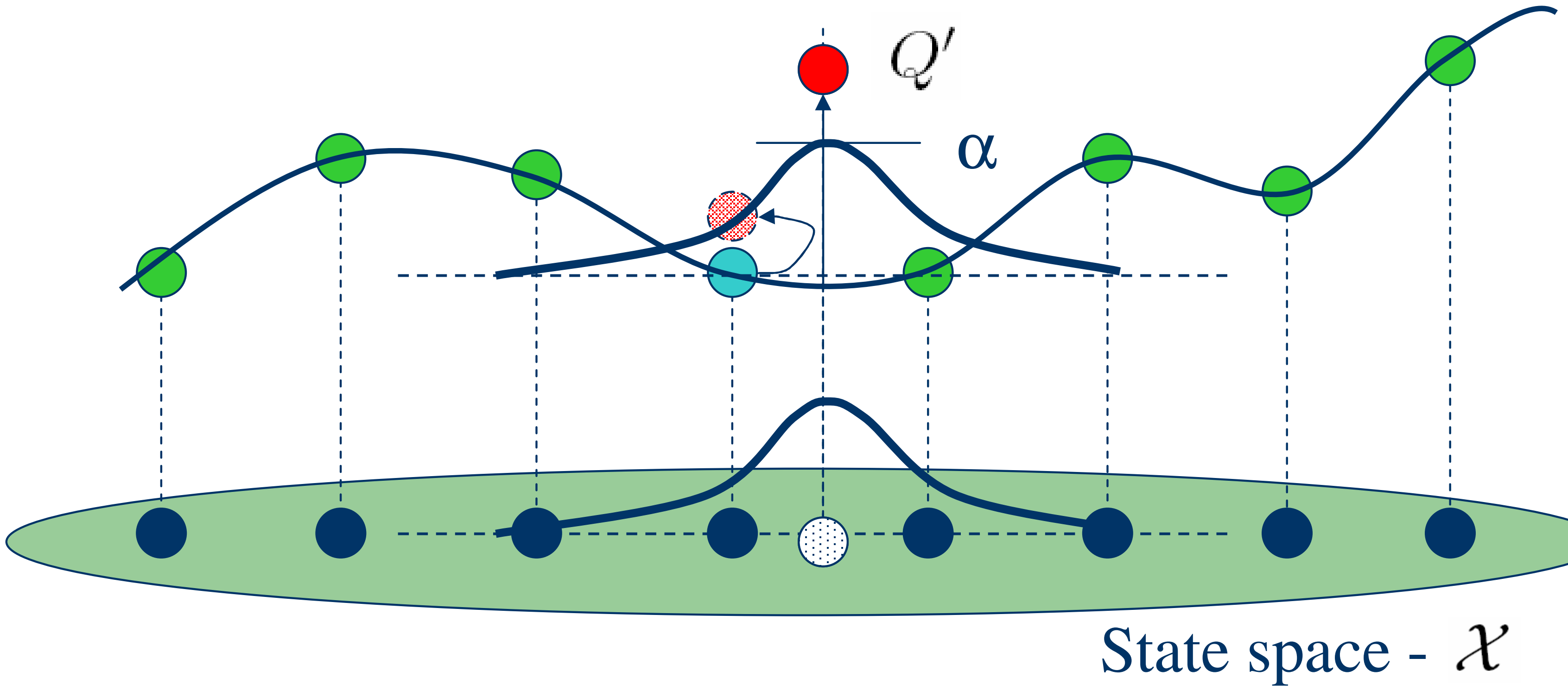


$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

Multi-component updates

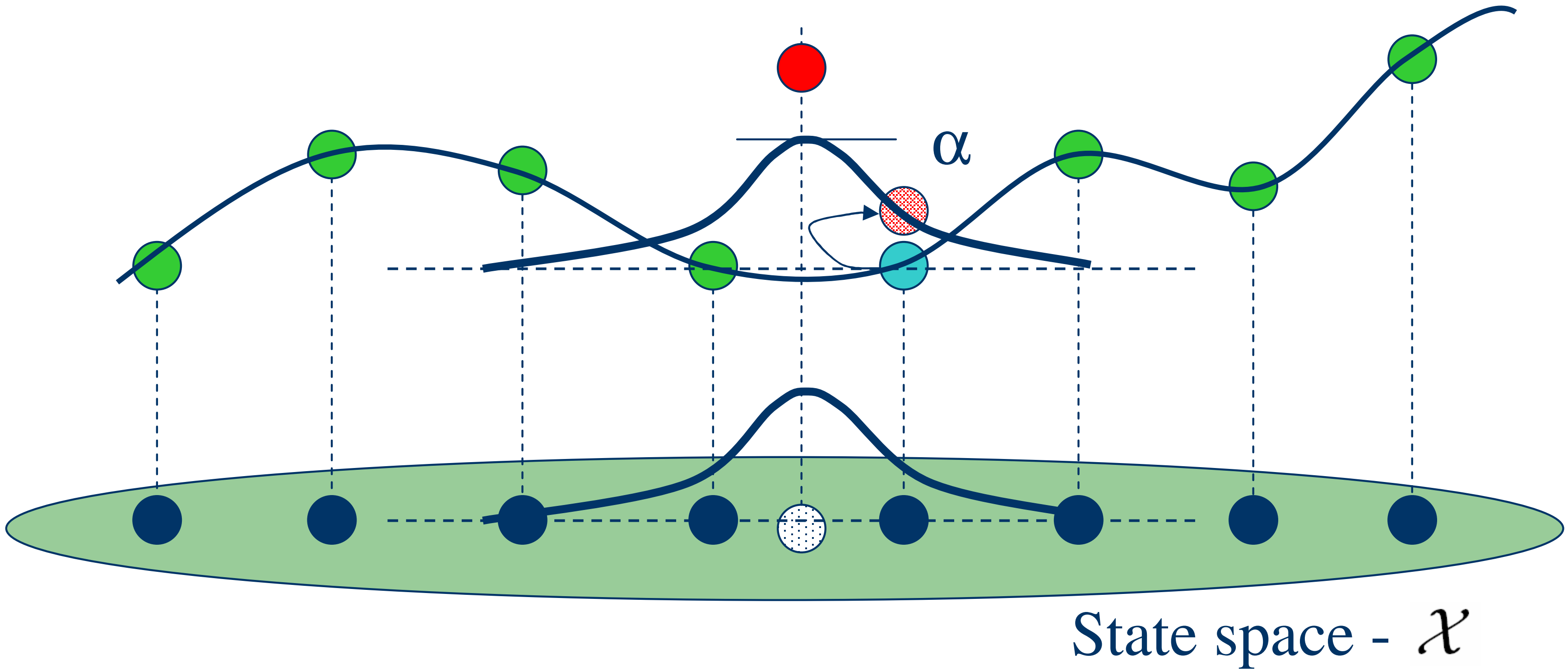
- Ribeiro & Szepesvari, 1996
  "spreading"
- Szepesvari & Littman, 1999
  "multi-state updates"

iFAPP-Q Learning



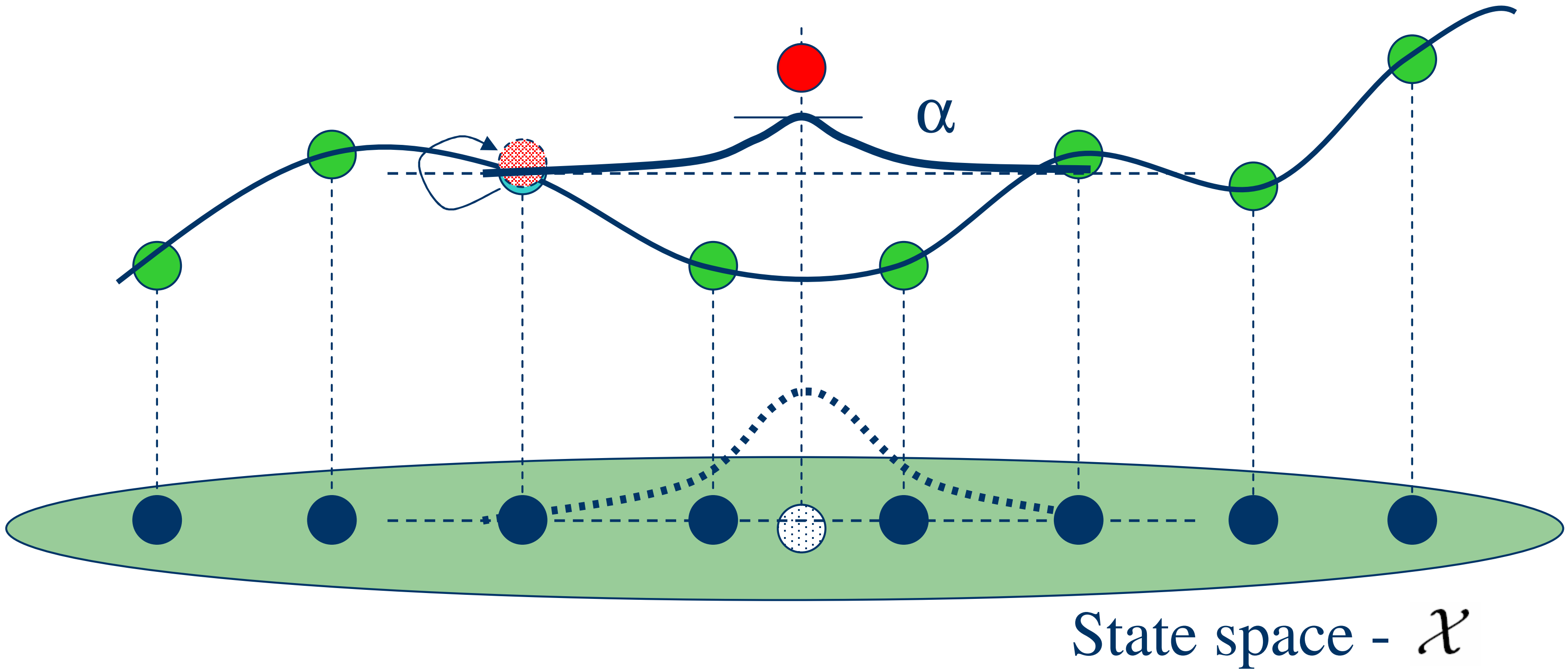
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



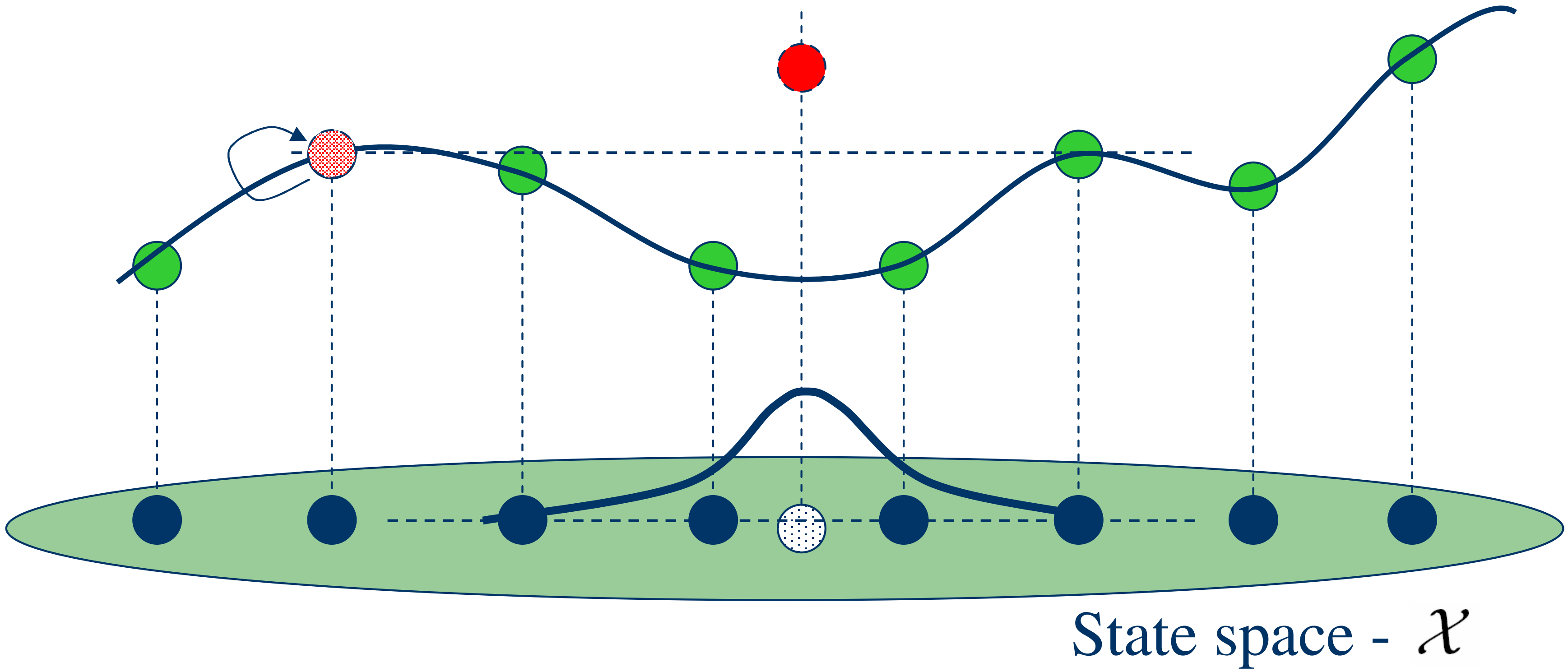
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



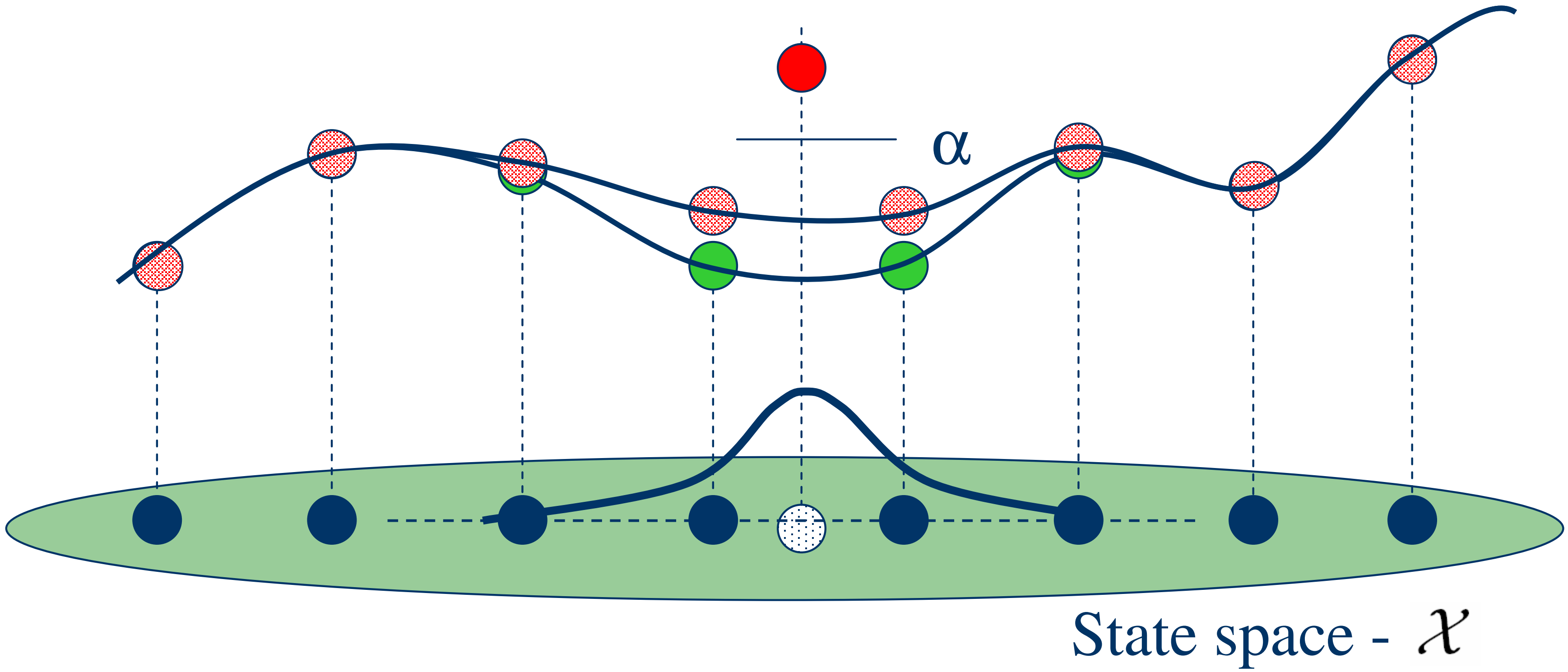
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



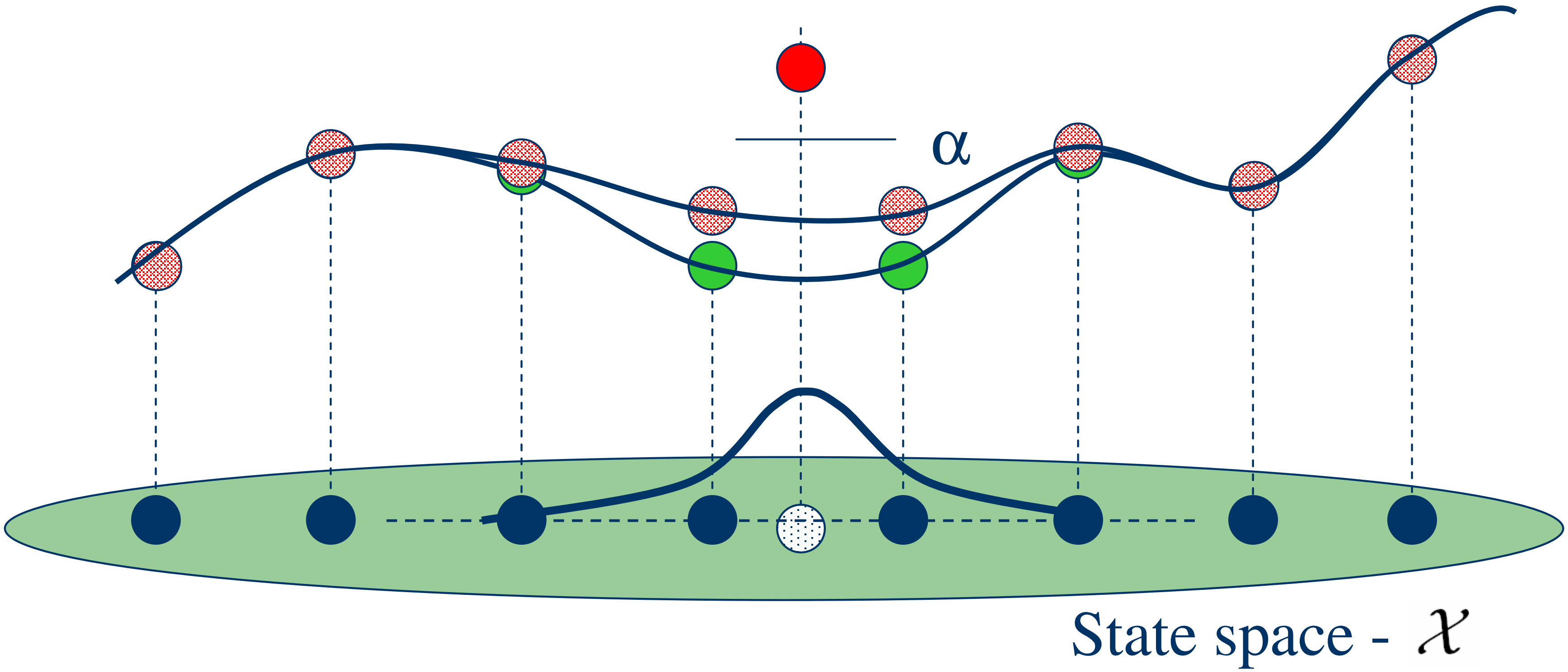
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



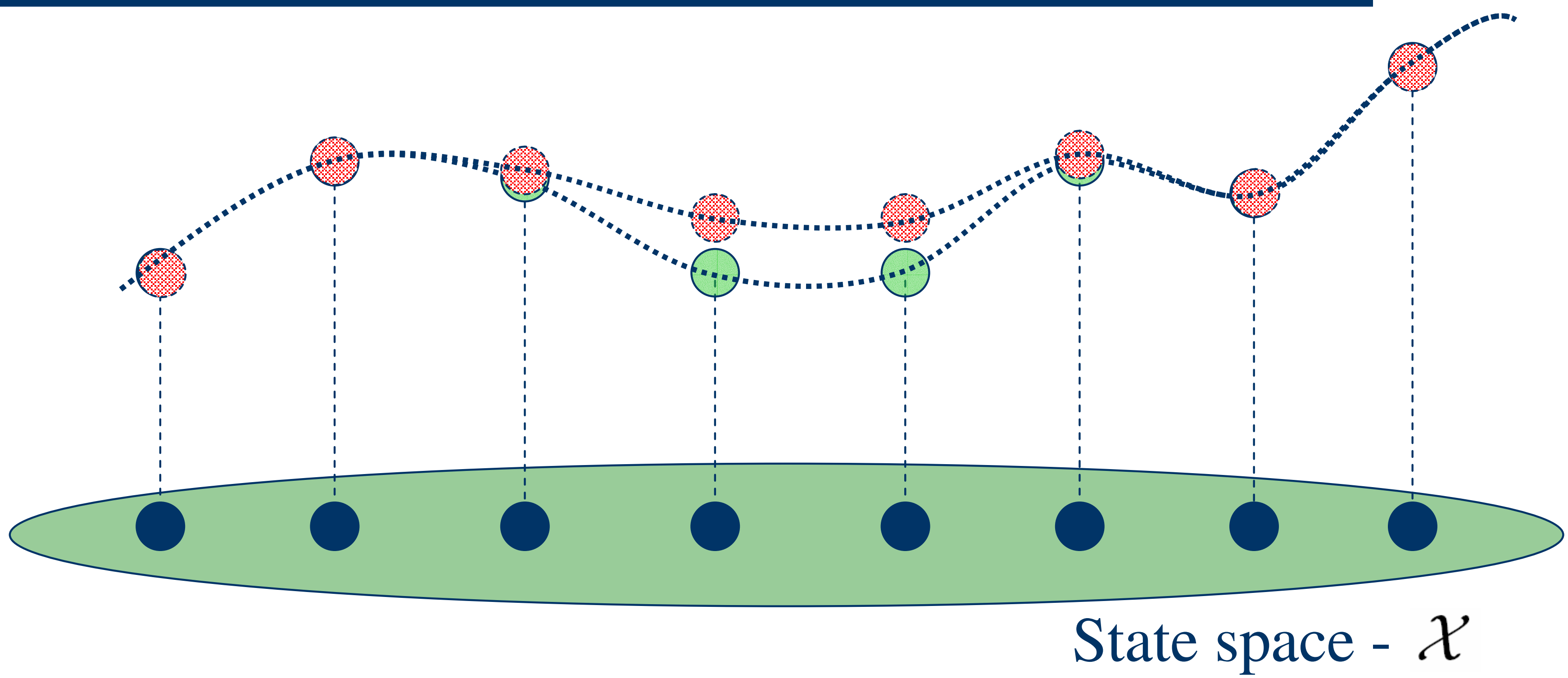
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



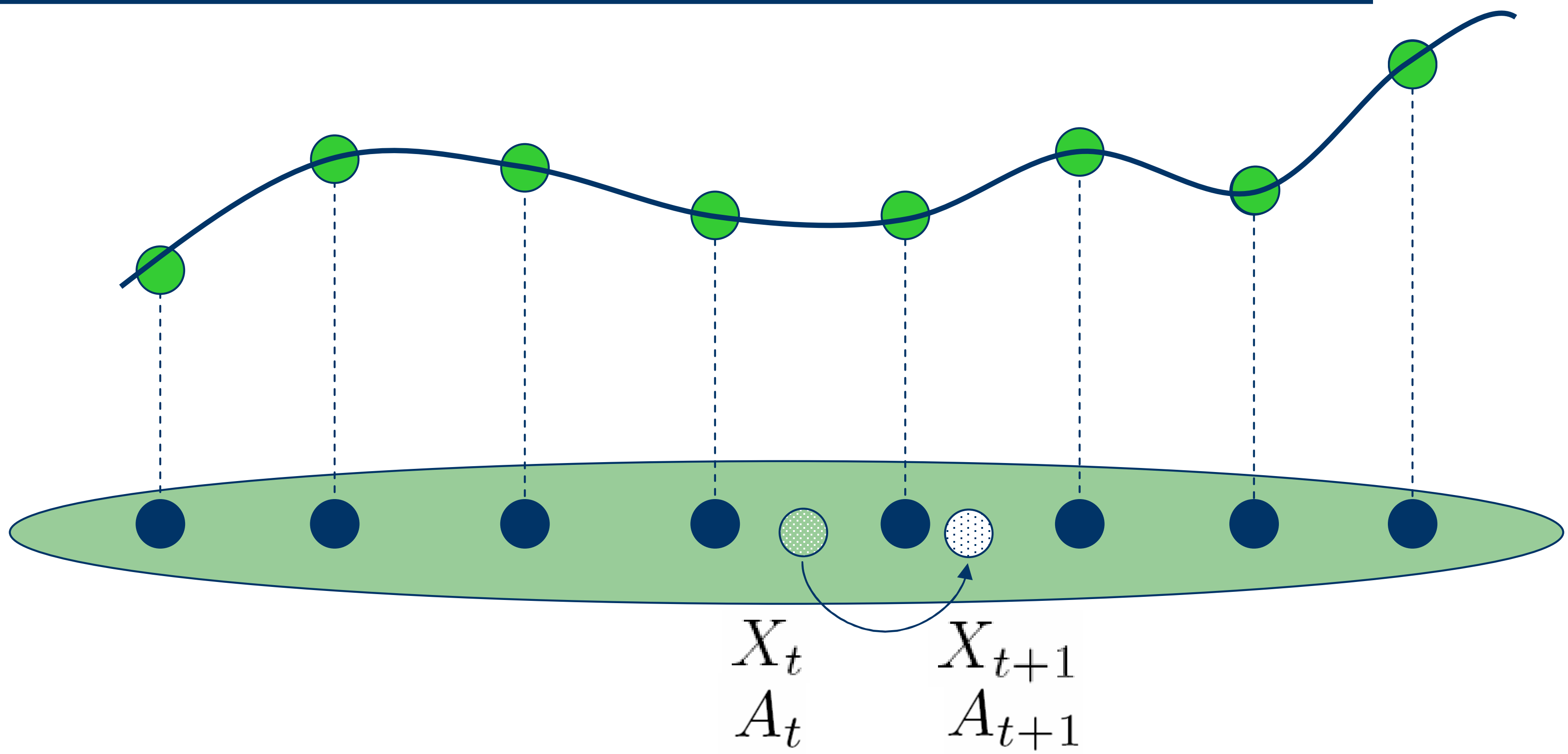
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



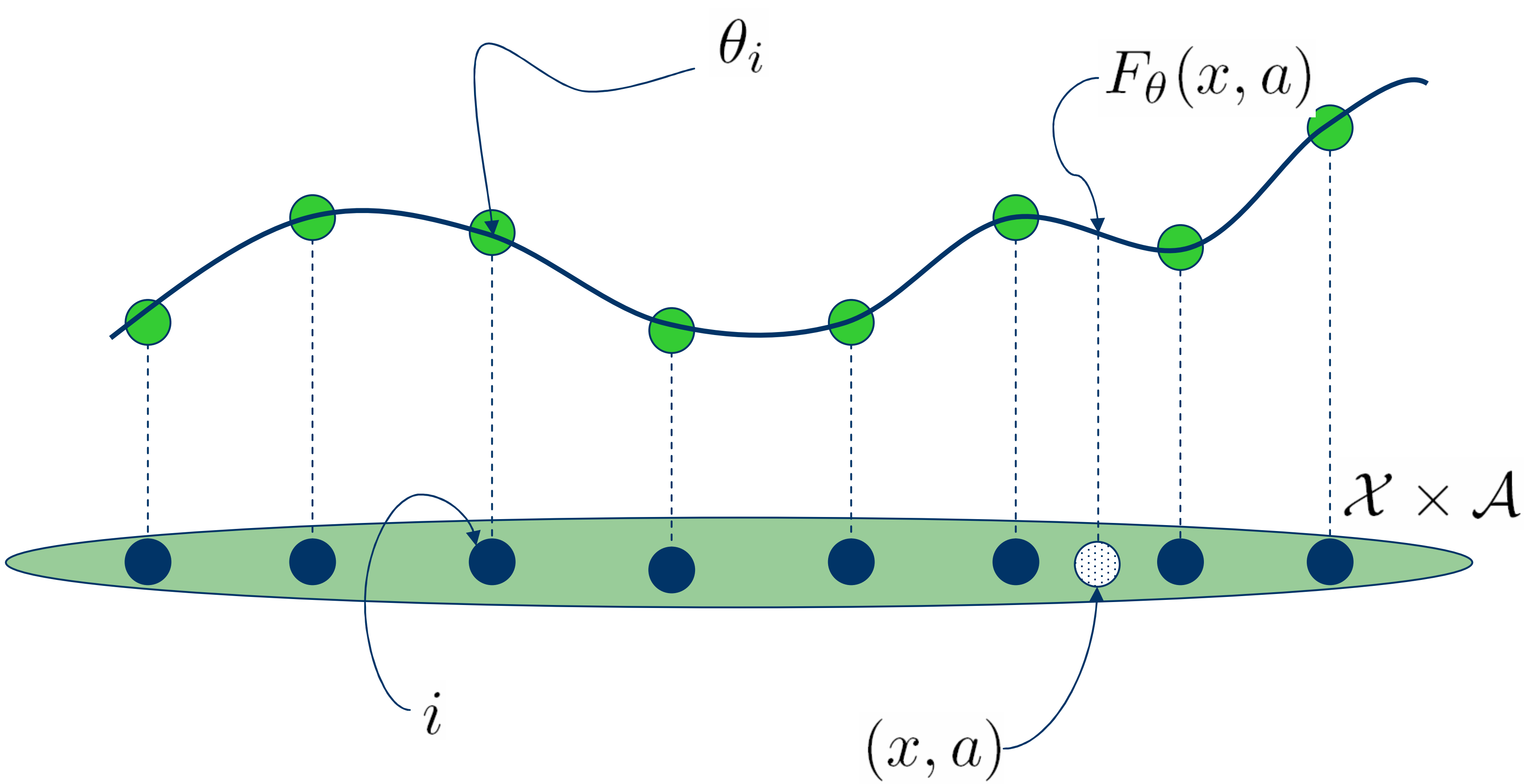
$$Q'(X_t, A_t) = R_t + \gamma \max_b Q(X_{t+1}, b)$$

iFAPP-Q Learning



Equations?

Notation



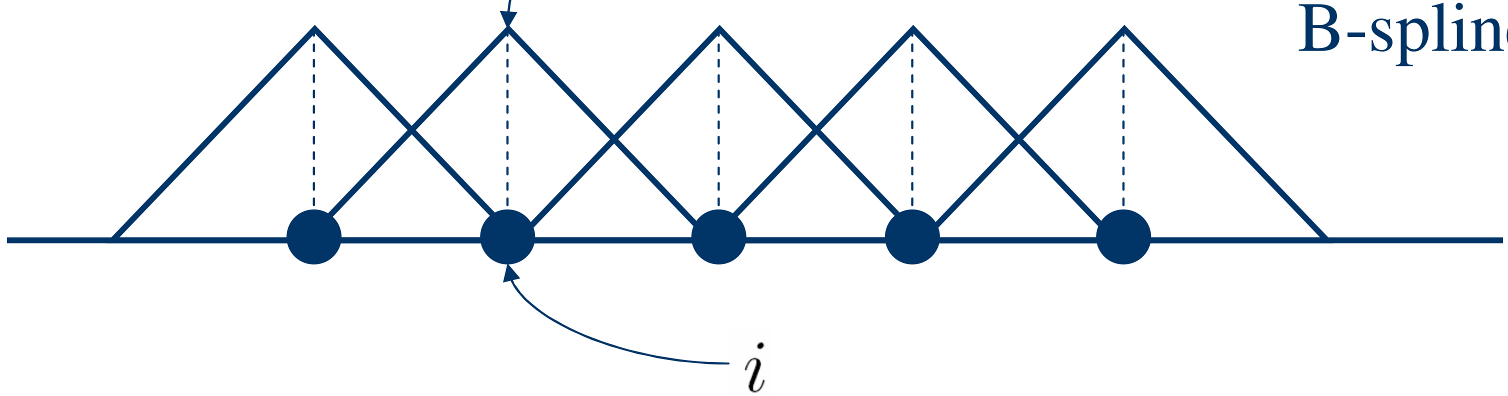
Gordon, 1995

Averagers

$$(F\theta)(x,a) = \sum_i \beta_i(x,a)\theta_i$$

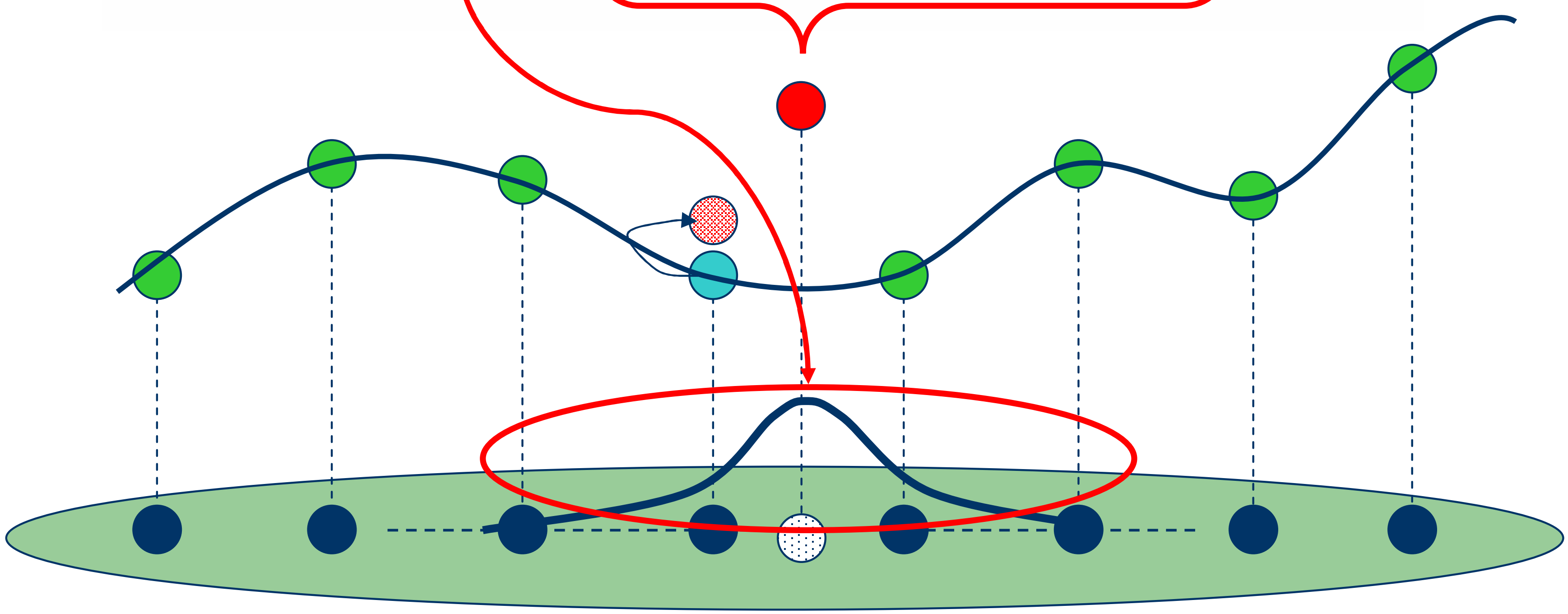
$$\sum_i \beta_i(x,a) = 1, \beta_i(x,a) \geq 0$$

..1st order B-splines



aFAPP-Q Learning Equation

$$\Delta\theta_{ti} = \alpha_{ti}\, s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$



iFAPP-Q Learning vs. Q-Learning

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$

$$\Delta Q_t(X_t, A_t) = \alpha_t(X_t, A_t) \left(R_t + \gamma \max_b Q_t(X_{t+1}, b) - Q_t(X_t, A_t) \right)$$

aFAPP-Q Learning vs. Q-Learning

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$

$$\Delta Q_t(X_t, A_t) = \alpha_t(X_t, A_t) \left(R_t + \gamma \max_b Q_t(X_{t+1}, b) - Q_t(X_t, A_t) \right)$$

aFAPP-Q Learning/Details

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$

$$s : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}^+$$

$$s_{ti} = s(x_i, a_i, X_t), \quad i = 1, \ldots, n.$$

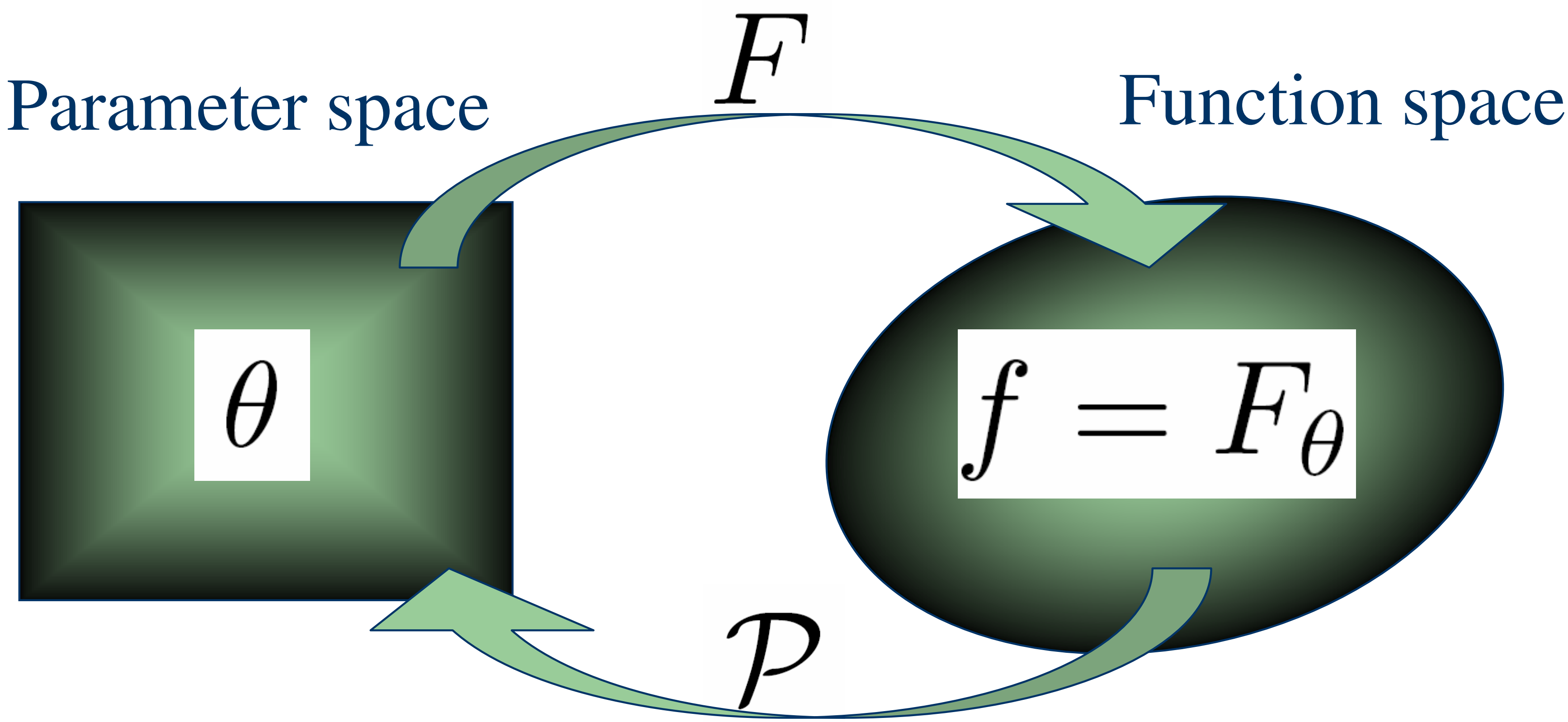
$$\alpha_{ti} = \frac{\chi(s(x_i, a_i, X_t) > \epsilon)}{n_t(x_i, a_i)}$$

$$n_t(x_i, a_i) = 1 + \sum_{s=0}^{t} \chi(s(x_i, a_i, X_s) > \epsilon)$$

Theory

- FAPPs as operators
- Theorem
- Assumptions
- Proof outline

Another View of Function Approximation



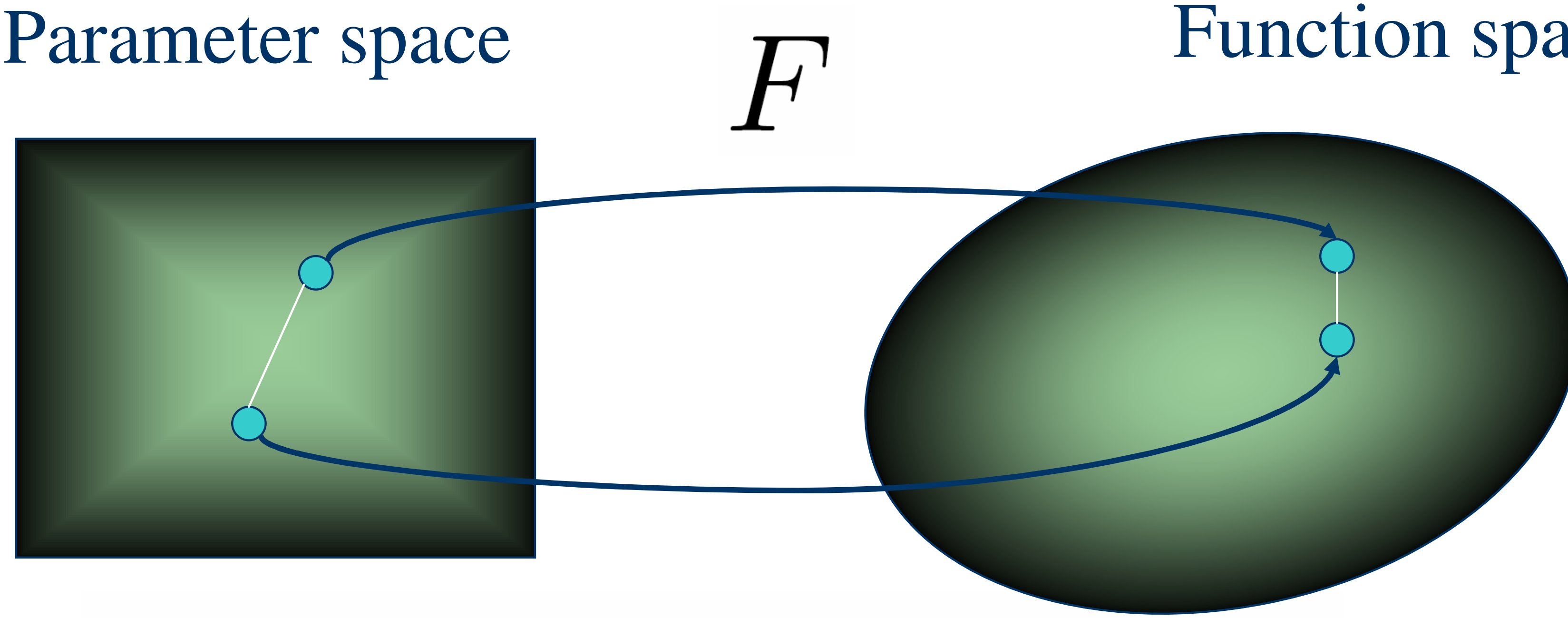
$$(Pf)_i = f(z_i)$$

$$z_i = (x_i, a_i)$$

Non-expansions

Parameter space

Function space



$$d_2(F(\theta_1), F(\theta_2)) \leq d_1(\theta_1, \theta_2)$$

Use sup-norm in both spaces!

Convergence Theorem

Under **Assumptions A1-A4**, and if *F* is a **non-expansion** then

$$\theta_t$$

converges to some vector θ^* w.p.1, such that

$$\hat{Q}^* = F\theta^*$$

is the **fixed point** of the operator $F\mathcal{P}\mathcal{H}$

where $\mathcal{H} : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ and $\mathcal{H}(Q)(z,a)$ is defined by

$$\int\int \hat{s}(z,a,x)\{r(x,a) + \gamma \max_b Q(y,b)\}dP(y|x,a)d\mu_X(x)$$

Assumption A1: MDP

- $(\mathcal{X}, A, p, r, \gamma)$ is a discounted MDP
  - A finite
  - $\mathcal{X}$ is a compact subset of a separable metric space (e.g. an *n*-dimensional Euclidean space)
  - r is continuous

Assumption A2: Sampling

Actions:

$$A_t \sim \pi(a = \cdot | X_t)$$

$$\pi(a|x) > 0$$

~ "positive recurrent"

States:

$$X_0 \sim \pi_0 \qquad X_{t+1} \sim dP(\cdot | X_t, A_t)$$

(X_t) is positive Harris, aperiodic

Rewards:

$$E[R_t | H_t] = r(X_t, A_t)$$

Assumption A3:
Conditions on the Influence Function *s*

$$s : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}^+$$

- bounded, measurable

Positivity condition:

$$\int s(x_i, a_i, z) d\mu_X(z) > 0$$

μ_X - unique invariant measure underlying (X_t)

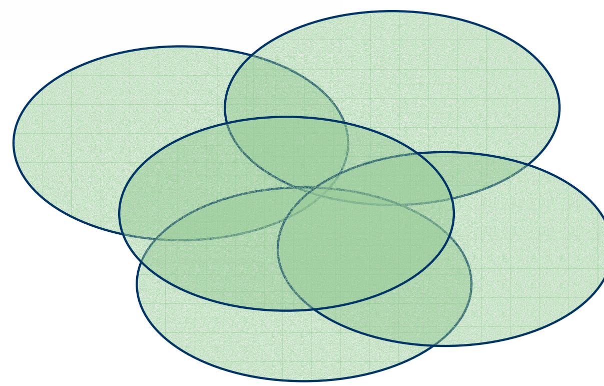
Assumption A4: Learning Rates

Counting visits:

$$n_t(x_i, a_i) = 1 + \sum_{s=0}^{t} \chi(s(x_i, a_i, X_s) > \epsilon)$$

Learning rates:

$$\alpha_{ti} = \frac{\chi(s(x_i, a_i, X_t) > \epsilon)}{n_t(x_i, a_i)}$$



Constraint on ϵ

$$\mu_{\mathcal{X}}(\{\, z \in \mathcal{X} \mid s(x_i, a_i, z) > \epsilon \,\}) > 0$$

N_i

Definition of $\hat{s}$

$$\int\int \hat{s}(z,a,x)\{r(x,a)+\gamma\max_b Q(y,b)\}dP(y|x,a)d\mu_X(x)$$

Truncation:

$$s_\epsilon(x,a,y)=\chi(s(x,a,y)>\epsilon)s(x,a,y)$$

Normalization:

$$\hat{s}(z,a,x)=\frac{s_\epsilon(z,a,x)}{\int s_\epsilon(z,a,x)d\mu_X(x)}$$

Proof (outline)

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F(X_{t+1}, b) - \theta_{ti} \right)$$

allows
..componont-wise analysis
..use of standard stochastic approximation

How?

Proof: Stochastic Approximation

- (X_t) positive Harris, aperiodic (A2)

$t \to \infty$

⇒ Time averages -> expected values

⇒ $$\frac{n_t(x_i, a_i)}{t+1} \to \mu_X(N_i) > 0 \qquad \text{(A4)}$$

⇒ $\sum_{t=1}^{\infty} \alpha_{ti} = \infty$ and $\sum_{t=1}^{\infty} \alpha_{ti}^2 < \infty$

Szepesvari&Littman, 1999

Proof: Relaxation Processes

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F(X_{t+1}, b) - \theta_{ti} \right)$$

$$\theta_{ti} \rightarrow \int \int \hat{s}(x_i, a_i, x)\{r(x, a_i) + \gamma \max_b F(y, b)\} dP(y|x, a_i) d\mu_X(x)$$

$$(\mathcal{J}F)_i = (\mathcal{H}F)(x_i, a_i)$$

..holds for all F

Proof

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F(X_{t+1}, b) - \theta_{ti} \right)$$

$$\theta_{ti} \to (JF)_i$$

Proof

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_\theta(X_{t+1}, b) - \theta_{ti} \right)$$

$$\theta_{ti} \longrightarrow (JF_\theta)_i$$

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$

?

Fixed Point

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right)$$

$$\theta_t \longrightarrow \theta^* = \mathcal{J} F \theta^*$$

..since F is a non-expansion

Proof: Finishing Steps

$$\theta^* = \mathcal{J} F \theta^*$$

$$F\theta^* = F \mathcal{J} F \theta^*$$

$$\mathcal{J} = \mathcal{P}\mathcal{H}$$

$$F\theta^* = F\mathcal{P}\mathcal{H}F\theta^*$$

$$\hat{Q}^* = F\mathcal{P}\mathcal{H}\hat{Q}^*$$

Q.e.d

How to Choose *F*?

Averagers:

$$(F\theta)(x,a) = \sum_i \beta_i(x,a)\theta_i$$

$$\sum_i \beta_i(x,a) = 1, \beta_i(x,a) \geq 0$$

Averagers are non-expansions!

What Does this Theorem Tell Us?

Theorem – Once Again

Under **Assumptions A1-A4**, and if F is a **non-expansion** then

$$\theta_t$$

converges to some vector θ^* w.p.1, such that

$$\hat{Q}^* = F\theta^*$$

is the **fixed point** of the operator $F\mathcal{P}\mathcal{H}$

where $\mathcal{H} : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ and $\mathcal{H}(Q)(z,a)$

is defined by

$$\int\int \hat{s}(z,a,x)\{r(x,a) + \gamma \max_b Q(y,b)\} dP(y|x,a) d\mu_X(x)$$

~~Interpolation~~-based Q-learning

Csaba Szepesvári
MTA SZTAKI
William D. Smart
WUSTL

Averagers-based Q-learning (and other convergent value function approximation algorithms)

Csaba Szepesvári

MTA SZTAKI

William D. Smart

WUSTL

Generalizations

- Other learning algorithms
- Increasing accuracy
- Approximating Q*

Generalizations

$$V_{t+1} = TV_t$$

$$V_{t+1} = F\mathcal{P}TV_t$$

$$V_t = F\theta_t,\ \theta_{t+1} = ?$$

$$\mathcal{P}V_{t+1} = \mathcal{P}F\mathcal{P}TV_t$$

ASSUME $\mathcal{P}V_{t+1} = \theta_{t+1}$

$$\theta_{t+1} = \mathcal{P}F\mathcal{P}TV_t = \cancel{\mathcal{P}F}\mathcal{P}TF\theta_t$$

$$\mathcal{P}F\theta_{t+1} = \theta_{t+1}$$

$$\mathcal{P}F\theta = \theta$$

"interpolation assumption"

Generalizations

Why not start with

$$\theta_{t+1} = \mathcal{P} T F \theta_t \quad ?$$

Generalization to "learning":

$$\theta_{t+1} = \mathcal{P} T_t F \theta_t$$

-learning rates
-observed data

Generalizations

Why not start with

$$\theta_{t+1} = \mathcal{P} T F \theta_t \quad ?$$

Generalization to "learning":

$$\theta_{t+1} = \mathcal{P} T_t (F\theta_t, F\theta_t)$$

"Meta Theorem"

- Let T be a contraction.
  If F is an **interpolative non-expansion**
  and $\mathcal{T}_t$
  defines a "relaxation process" when its second parameter is fixed approximating T w.p.1, and if

$$\begin{aligned} |\mathcal{T}_t(U_1, V) - \mathcal{T}_t(U_2, V)| &\leq G_t |U_1 - U_2|, \\ |\mathcal{T}_t(U, V_1) - \mathcal{T}_t(U, V_2)| &\leq F_t \left(\|V_1 - V_2\| + \lambda_t\right) \end{aligned}$$

  then

$$V_t \rightarrow \hat{V}^* = F\mathcal{P}T\hat{V}^*$$

  w.p.1

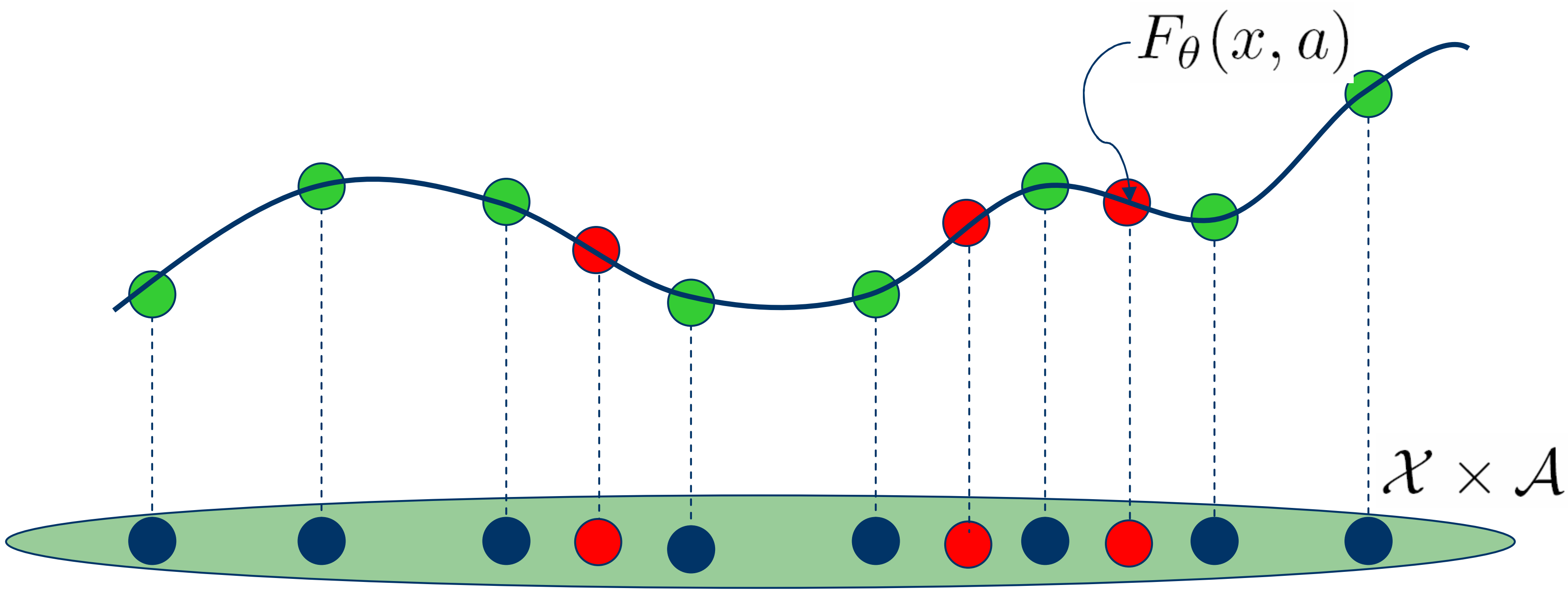
Applications

- FAPP +
  - Real-time dynamic programming (speeding up dynamic programming by not updating irrelevant parts of the state-space)
  - Value iteration with Monte-Carlo updates (when exact updates are too expansive)
  - Other criteria (e.g. Markov games, Risk-sensitive MDPs, ..)

Tsitsiklis&Van Roy, 1997
Gordon, 1995

Extensions

Increasing the Density of Basis Points



Algorithm

While (true)

{

Add basis points dens(S^t)>h_0

Run IFAPP-Q Learning for time T_t

}

Convergence

- Family of FAPPs: $F^{(n)} : \mathbb{R}^n \times Z^n \longrightarrow B(\mathcal{Z})$,
- Non-destructive refinement
- Expansion finishes in finite time with probability one

=> $\theta_t \longrightarrow \theta^*$ $\hat{Q}^* = F^\infty \theta^*$ (random)

$$F^\infty = \lim_{t\to\infty} F^{(n_t)}(\cdot, S^t)$$

$$\hat{Q}^* = F^\infty \mathcal{P}\mathcal{H}\hat{Q}^*$$

$$\|\hat{Q}^* - Q^*\| \leq O(h_0) \quad ?$$

Extension #2

While (true)

{

Add data points

Shrink influence function *s*

Estimate density underlying μ_X with κ_t

and

run modified iFAPP-Q for time T_t

}

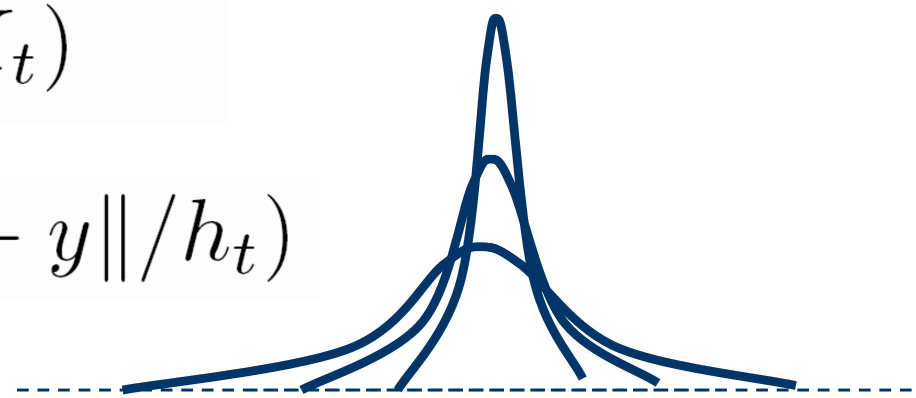
Output learned Q-function

Modified iFAPP-Q Learning

- Use

$$s_{ti} = \frac{s_t(x_{ti}, a_{ti}, X_t)}{\kappa_t(X_t)}$$

$$s_t(x, a, y) = \phi_a(\|x - y\|/h_t)$$



$$\phi_a : \mathbb{R}_0^+ \rightarrow \mathbb{R}$$

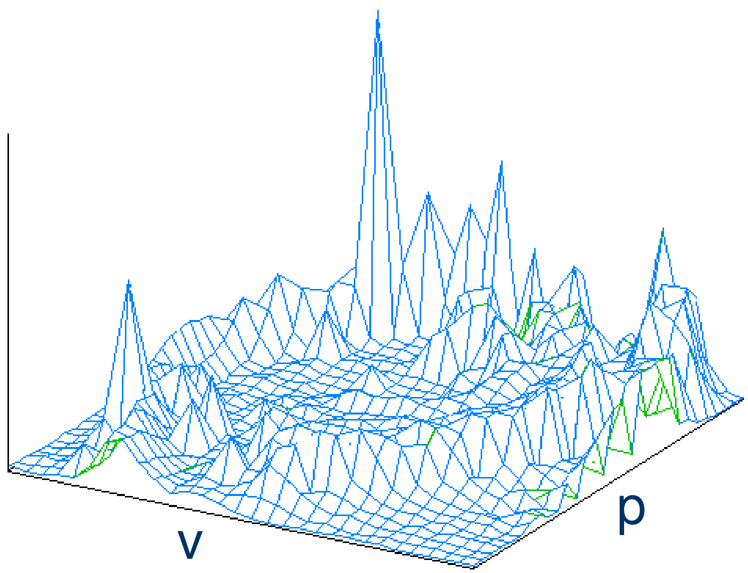
$$h_t \rightarrow 0$$

Some Experimental Results

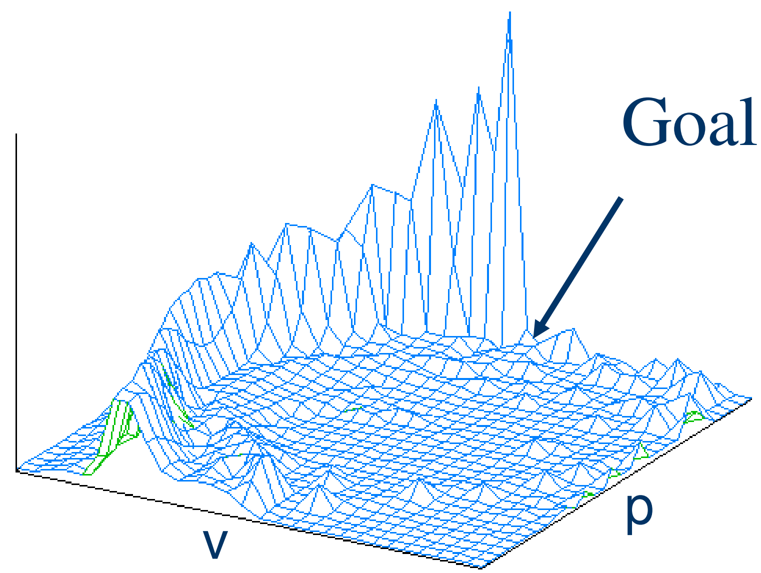
Comparison algorithms

- Soft state aggregation [Singh, Jaakkola & Jordan 95]
- Kernel-based RL [Ormoneit & Sen 02]
- Domain: car on the hill

State-action visit counts

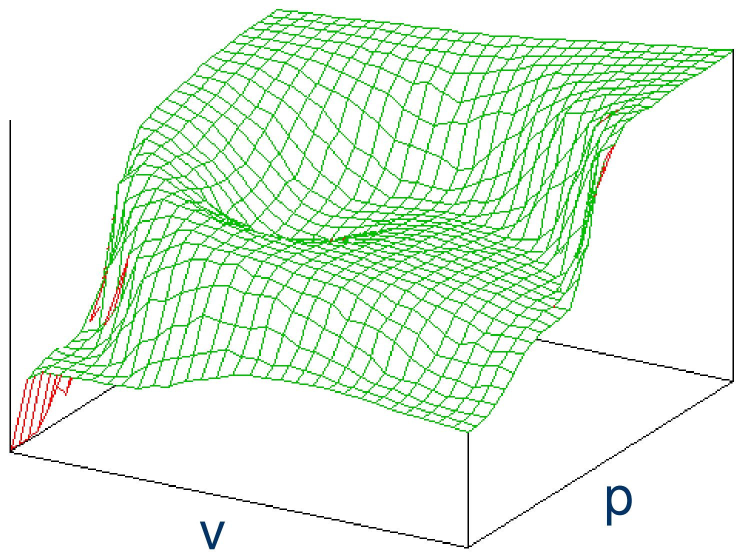


left

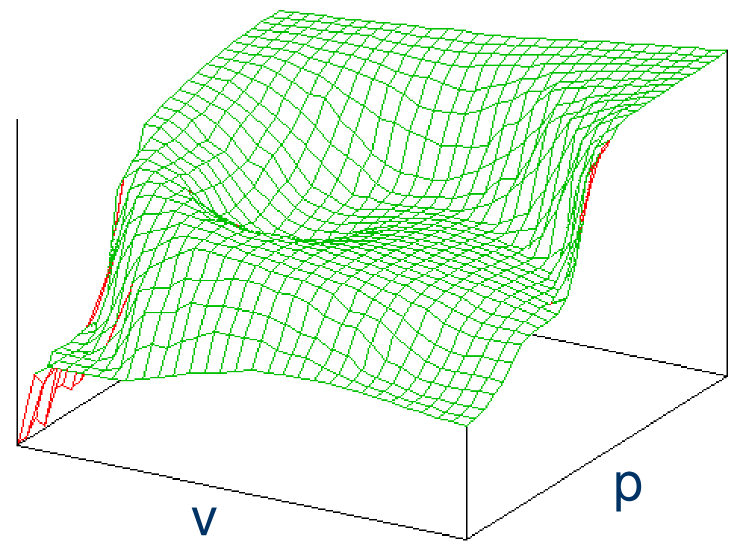


right

Q-values for “optimal” policy

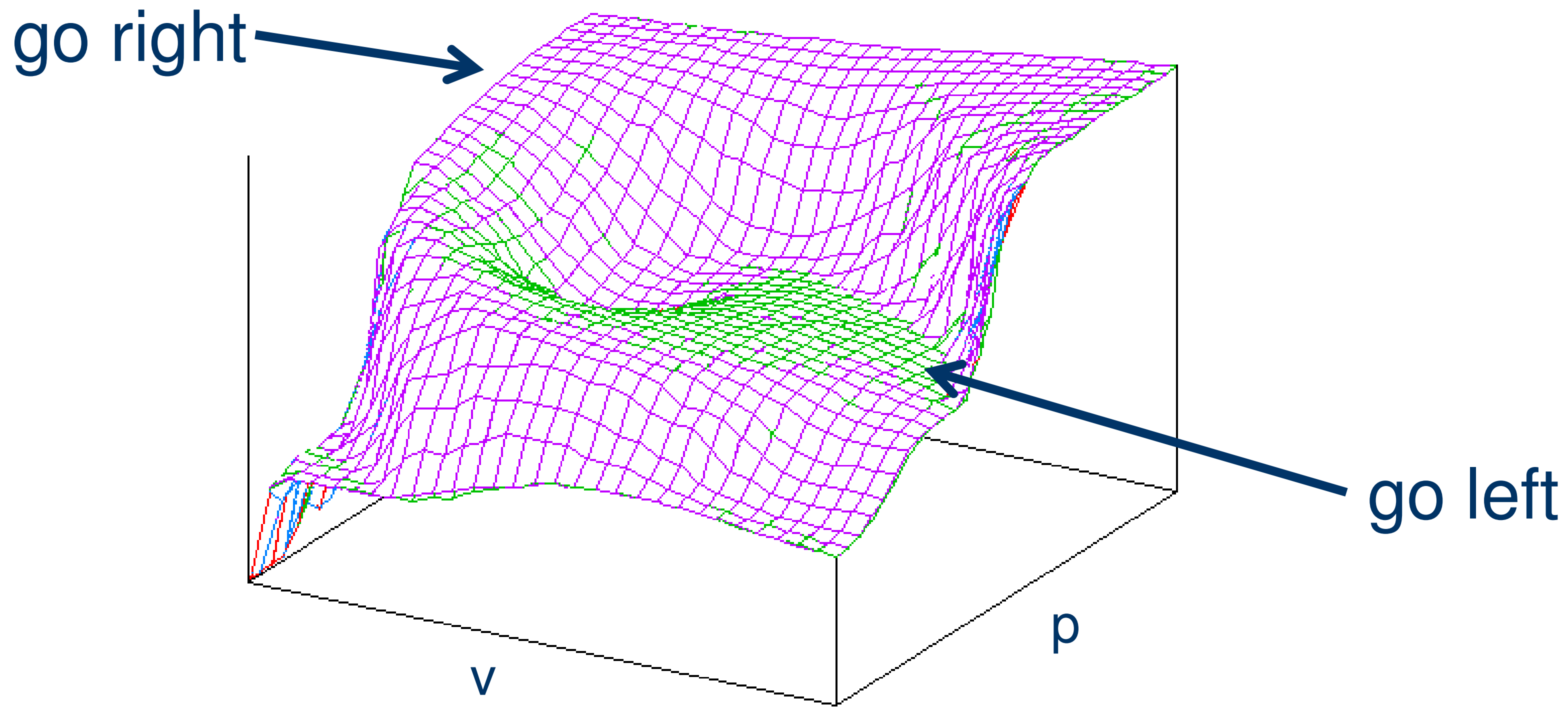


left

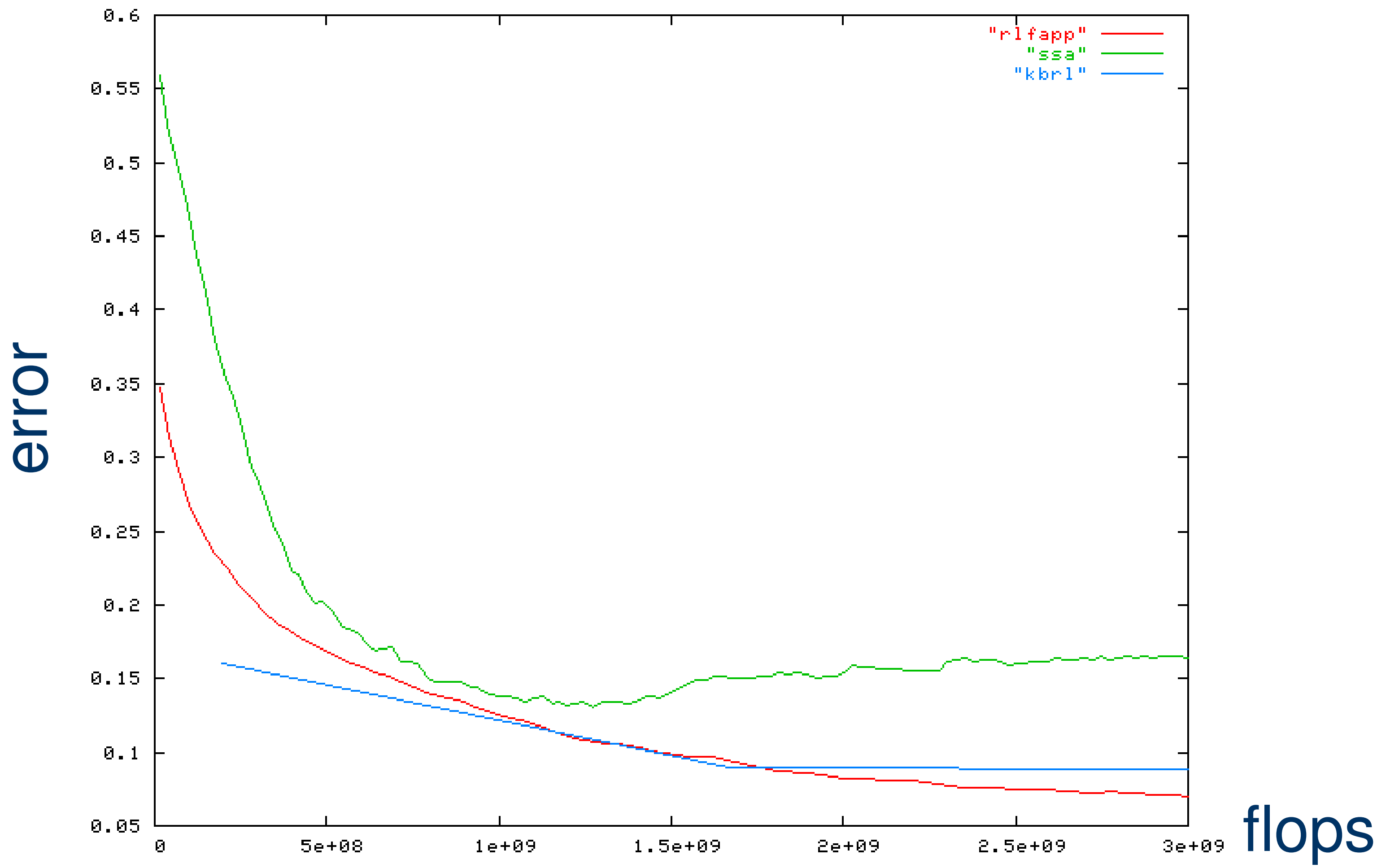


right

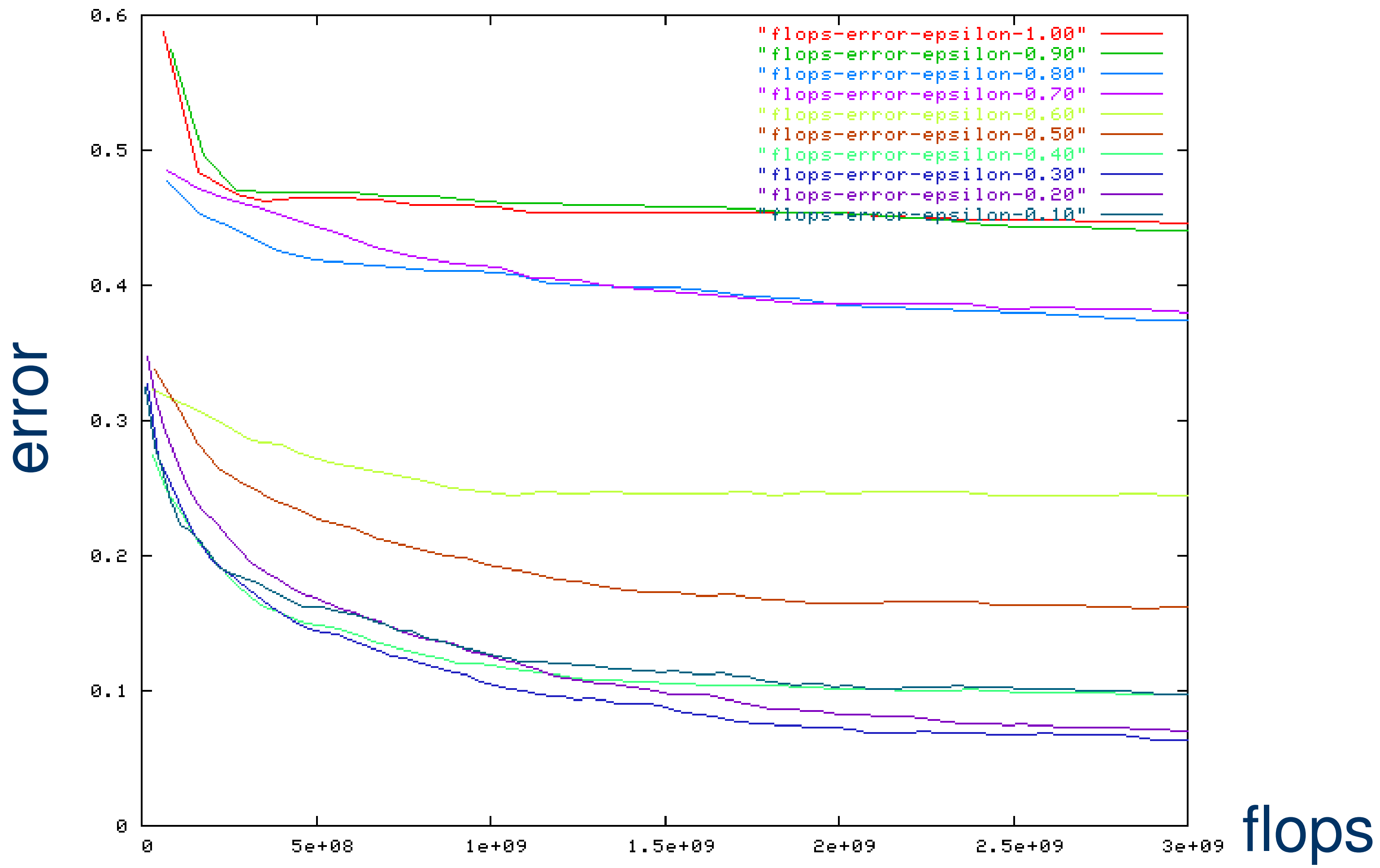
Q-values for "optimal" policy



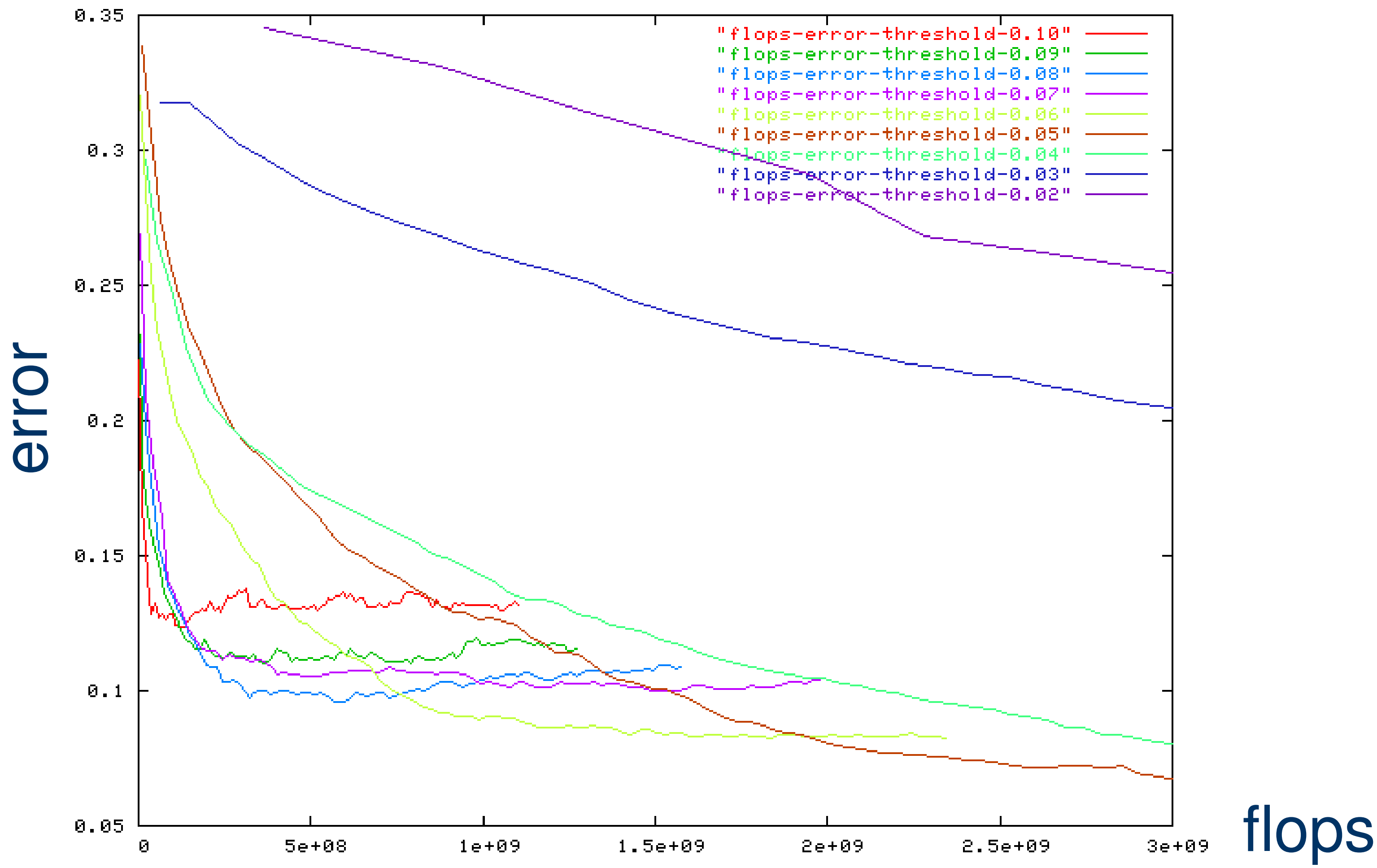
Performance comparison



Varying the exploration policy



Varying the density threshold



Conclusions

Conclusions

- iFAPP-Q: Extends Q-learning to continuous spaces
- Need to update multiple components of the parameter vector => influence function *s*
- Works for non-expansions
- Extensions are possible
- Changing the policy?
- When to add basis points?
- Other FAPPs (LWR?)

What Makes *F* "Interpolative"?

$\theta \in \mathbb{R}^n$ -parameter vector

$F_\theta : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$ -"FAPP"

$S = \{(x_1, a_1), \ldots, (x_n, a_n)\}$ - basis points

$$F_\theta(x_i, a_i) = \theta_i$$