Sequential decisions under uncertainty

KTH/EES PhD course

Lecture 4

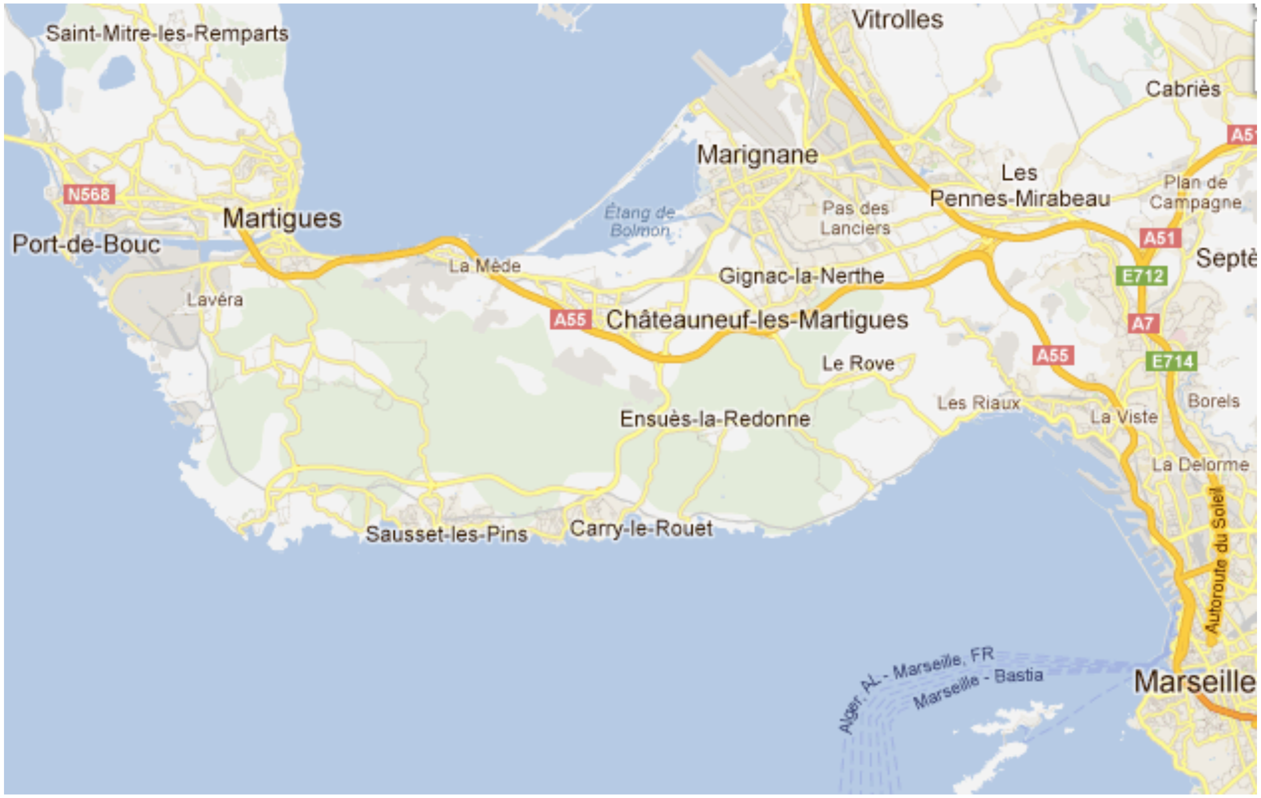
Lecture 4

- Stochastic approximation
  - Martingales
  - Convergence of stochastic approximation algorithms
- Q-learning

Stochastic approximation

Martingale?

- “Jouga a la martegalo” (Play like inhabitants of Martigues, i.e., naively)



Martingale?

- "Jouga a la martegalo" (Play like inhabitants of Martigues, i.e., naively)
- 18th century: gambling. "To play the martingale is to always bet all that was lost".
- Formalism: Paul Pierre Levy (1886-1971), Joseph Doob (1910-2004)

Martingales

- Probability space: $(\Omega, \mathcal{F}, P)$
- Martingale
  - A sequence of random variables $(X_n, n \in \mathbb{N})$
  - Natural filtration: $\mathcal{F}_n = \sigma(X_k, k \leq n)$
  - Definition:

$$E[\|X_n\|] < \infty, \quad \forall n \geq 0,$$

$$E[X_n | \mathcal{F}_{n-1}] = X_{n-1}, \quad a.s. \quad (n \geq 1)$$

- Examples: random walks, gambling processes, …

Doob convergence results

Theorem If $\sup_n E[\|X_n\|] < \infty,$

then $X_\infty = \lim_{n\to\infty} X_n,$ almost surely, and X_∞ is finite.

Theorem If $E[\|X_n\|^2] < \infty, \forall n,$

and if $\sum_n E[\|X_n - X_{n-1}\|^2] < \infty,$

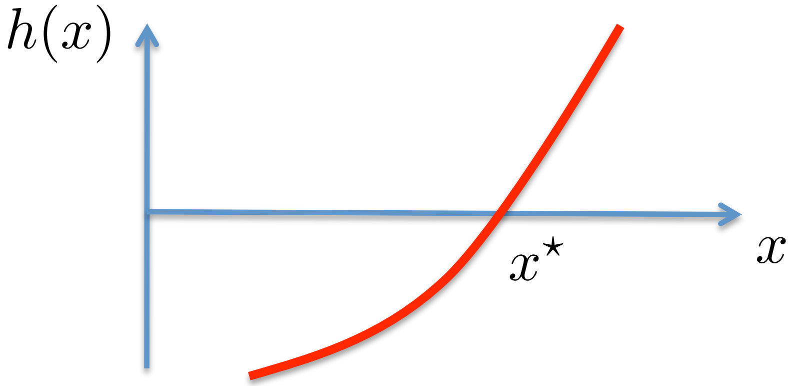
then $X_\infty = \lim_{n\to\infty} X_n,$ almost surely.

More on Martingales

- See e.g., Probability with Martingales, by Williams

Robbins-Monro algorithm

- Find the root of a function from noisy measurements



- Assume that at the *n* iteration, you select x_n

  You get a noisy measurement $y_n = h(x_n) + \xi_{n+1}$

  The zero-mean noise may depend on the selected x_n

$$E[\xi_{n+1}|\mathcal{F}_n] = 0$$

$$\mathcal{F}_n = \sigma(\xi_1, \ldots, \xi_n)$$

Robbins-Monro algorithm

- Algorithm (1951): $x_{n+1} = x_n - a_n \times y_n, \quad \forall n.$

- Assume that

$$\sum_n a_n = \infty, \quad \sum_n a_n^2 < \infty,$$

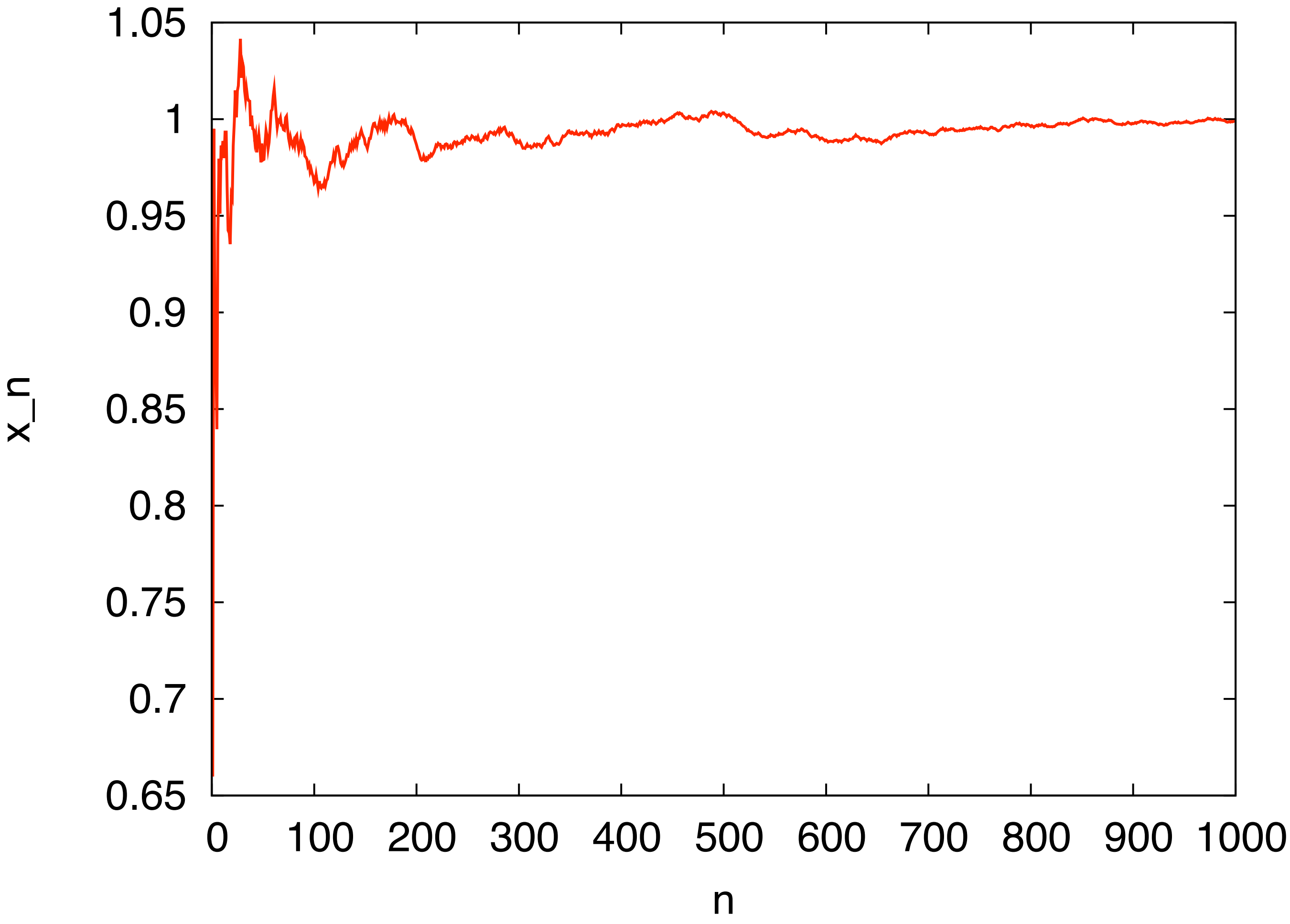
$$E[\xi_{n+1}^2 | \mathcal{F}_n] \leq K(1 + x_n^2), \quad a.s., \forall n$$

$$\sup_n |x_n| < \infty, a.s.$$

then $\lim x_n = x^\star, a.s.$

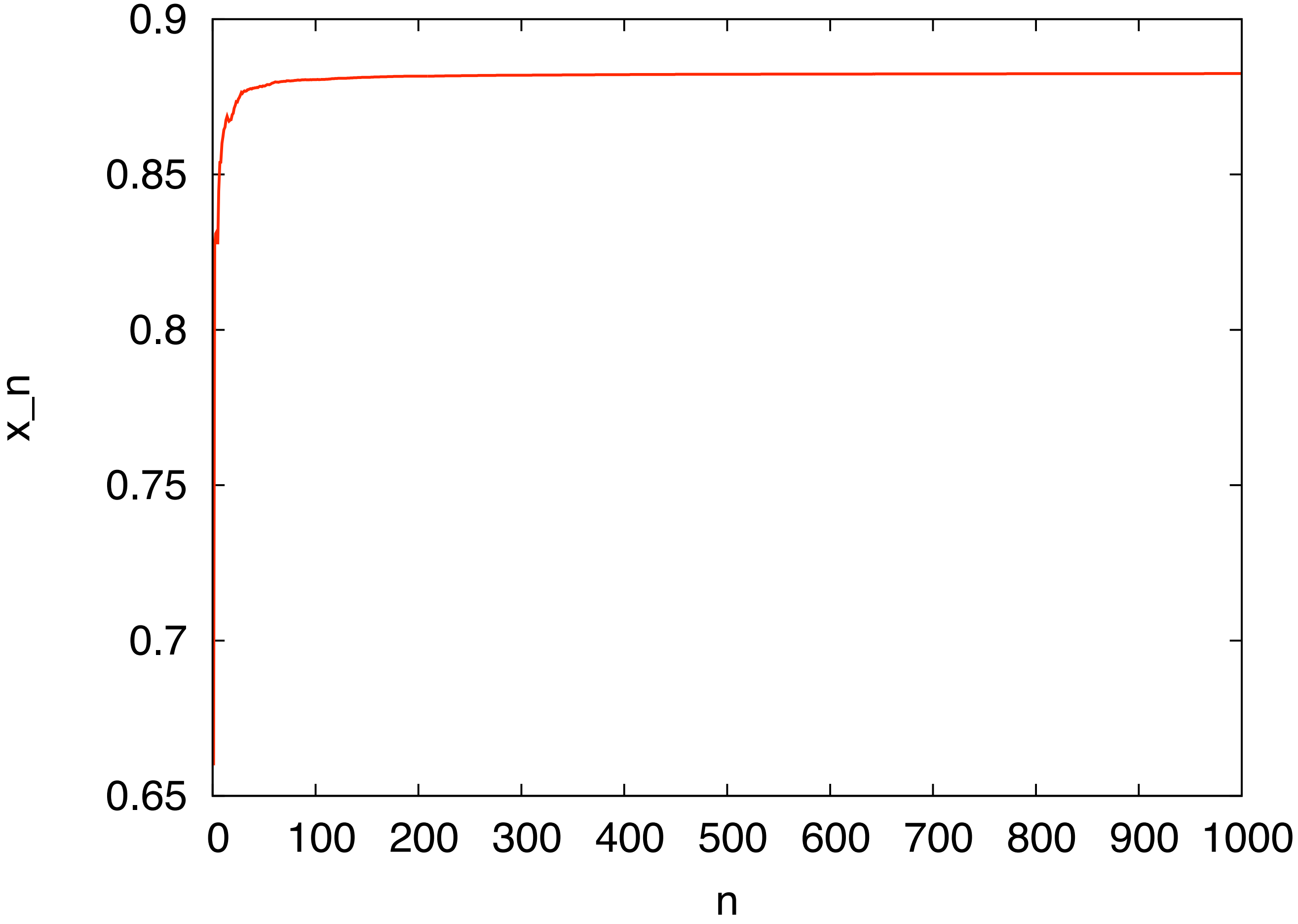
Example

$$h(x) = x^2 - 1, \quad a_n = 1/n, x_0 = 0$$



Example

$$h(x) = x^2 - 1, \quad a_n = 1/n^2, x_0 = 0$$



SA algorithm

- Algorithm: $x_{n+1} = x_n + a_n \times (h(x_n) + \xi_{n+1}), \quad \forall n.$
- Assumptions: $E[\xi_{n+1} | \mathcal{F}_n] = 0, \quad a.s., \forall n$

h *L*-Lipschitz

$$\sum_n a_n = \infty, \quad \sum_n a_n^2 < \infty,$$

$$E[\|\xi_{n+1}\|^2 | \mathcal{F}_n] \leq K(1 + \|x_n\|^2), \quad a.s., \forall n$$

$$\sup_n \|x_n\| < \infty, a.s.$$

ODE method

- Time: $t(0) = 0, \quad t(n) = \sum_{k=0}^{n-1} a_k, \forall n \geq 1$

$$\lim_{n \to \infty} t(n) = \infty$$

- Continuous piece-wise linear interpolation: $\bar{x}(t)$

$$\bar{x}(0) = 0$$

$$\bar{x}(t) = x_n + (x_{n+1} - x_n) \times \frac{t - t(n)}{t(n+1) - t(n)},$$

$$\forall t \in [t(n), t(n+1))$$

ODE method

- Approximate ODE: $x^s(s) = \bar{x}(s)$

  $$\dot{x}^s(t) = h(x^s(t)), \quad \forall t \geq s$$

- The interpolated algorithm trajectory is well approximated by the ODE:

Theorem For any $T > 0$,

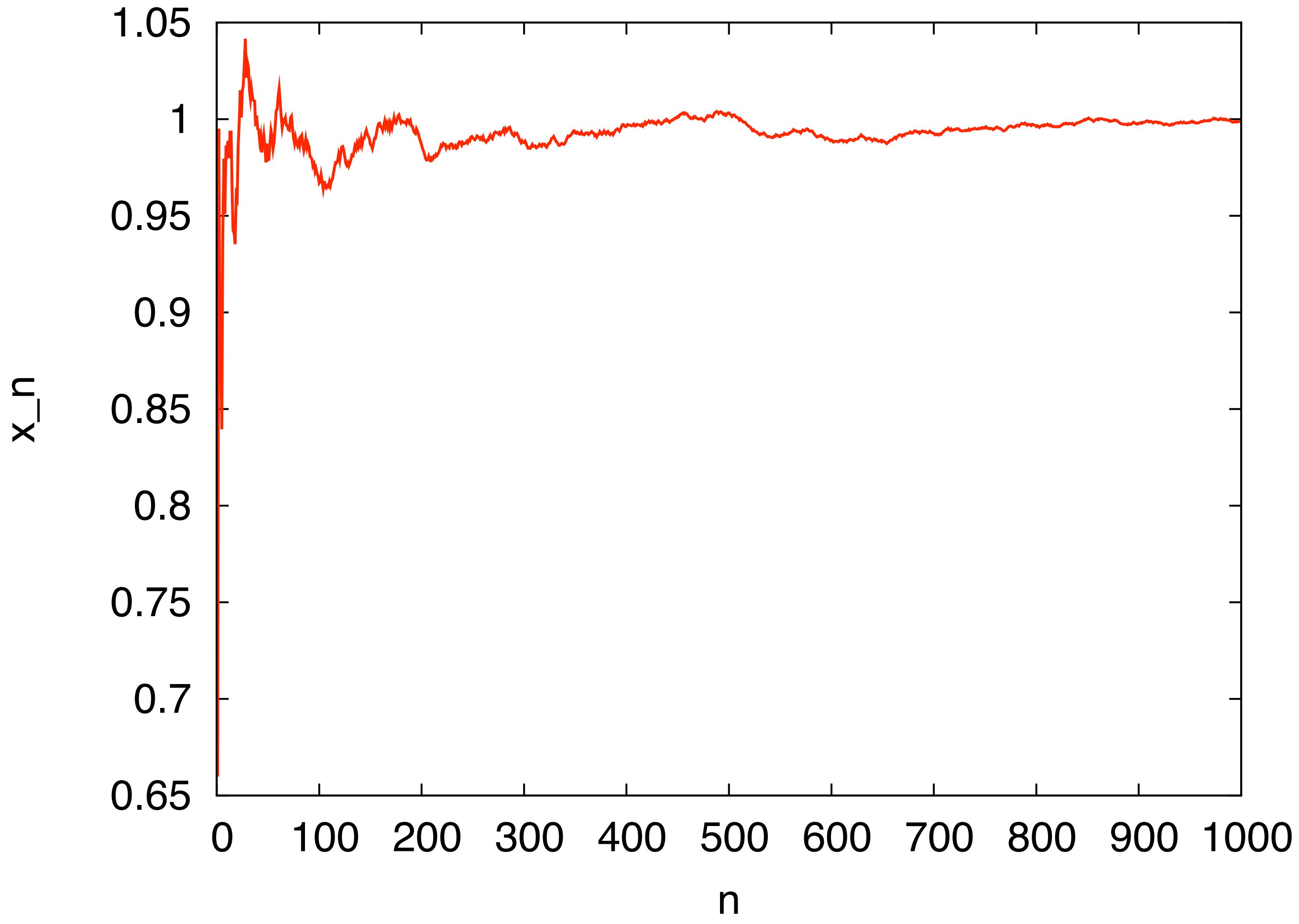
$$\lim_{s \to \infty} \sup_{t \in [s, s+T]} \|\bar{x}(t) - x^s(t)\| = 0, a.s.$$

ODE method

Corollary If *h* has a unique globally asymptotically stable point $x^\star$ then $\lim_{n\to\infty} x_n = x^\star$.

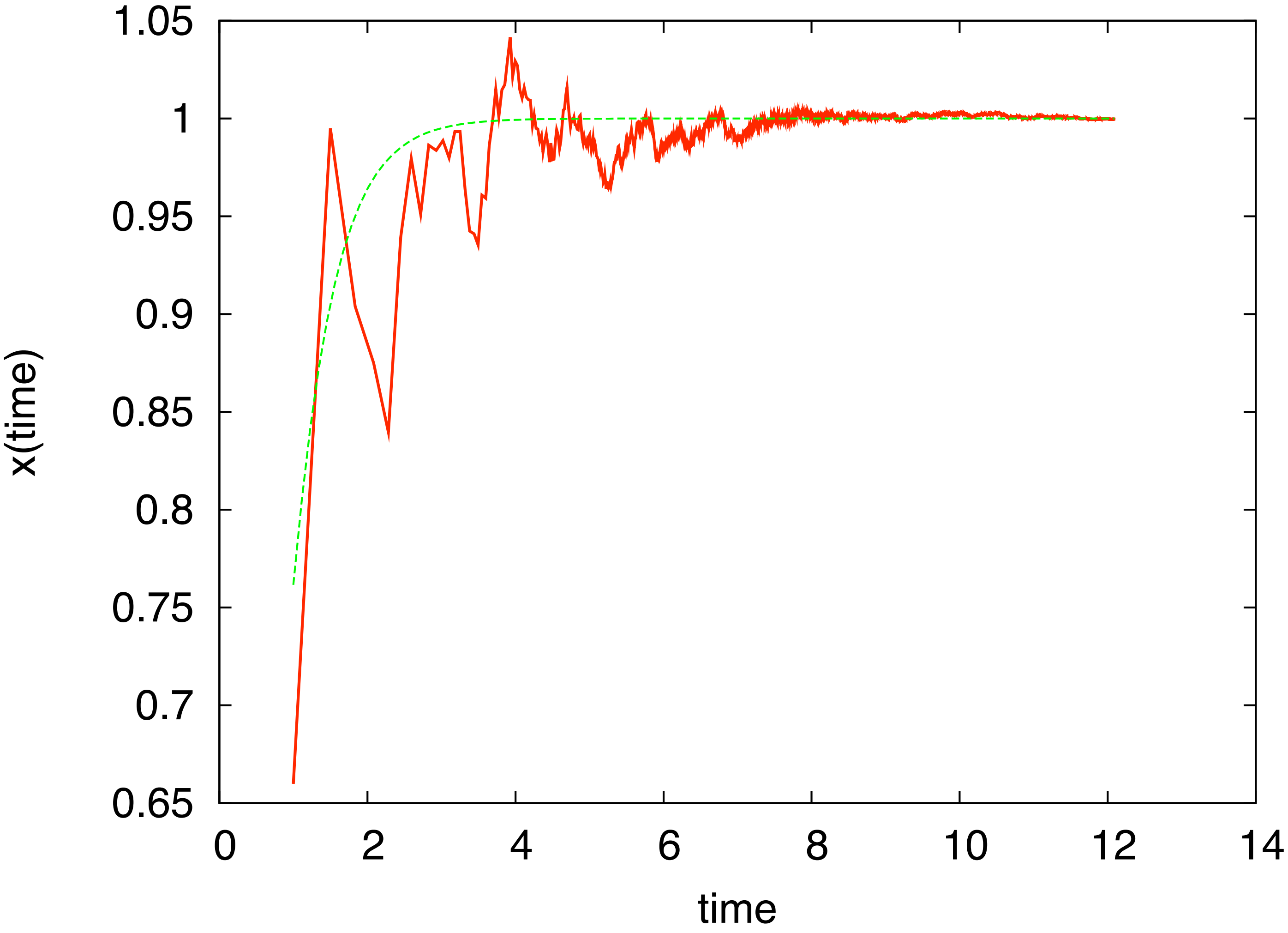
Example

$$h(x) = x^2 - 1, \quad a_n = 1/n, x_0 = 0$$



Example

$$h(x) = x^2 - 1, \quad a_n = 1/n, x_0 = 0$$



Proof – Gronwall lemmas

Lemma (Continuous) u, v positive continuous functions

$$u(t) \leq C + K \int_0^t u(s)v(s)ds, \quad \forall t \in [0,T]$$

$$\Longrightarrow \quad u(t) \leq C \exp(K \int_0^t v(s)ds), \quad \forall t \in [0,T]$$

Lemma (Discrete) x_n, a_n positive sequences

$$x_{n+1} \leq C + L \sum_{m=0}^{n} a_m x_m$$

$$\Longrightarrow \quad x_{n+1} \leq C \exp(L \sum_{m=0}^{n} a_m)$$

Refs on Stochastic Approximation

- A Stochastic Approximation method. H. Robbins, S. Monro. 1951, Annals of Math. Stat., 22.
- Random Iterative Models. M. Duflo. 1997, Springer.
- Stochastic Approximation and Recursive Algorithms. H. Kushner, H. Yin. 2003, Springer.
- Stochastic Approximation: A Dynamical Systems viewpoint. V. Borkar. 2008, Cambridge Univ.

Q-learning

C. Watkins, P. Dayan.
Machine Learning, 8 (1992), pp. 279-292

Model

- Policies: $\pi = (\pi_1, \pi_2, \ldots) \in HR$

  $\pi_t : H_t \to \mathcal{P}(A)$

- Assumptions:
  - Stationary rewards and transitions: $r(s,a), \quad p(j|s,a)$
  - Bounded rewards
  - Finite state space, finite action space

- Discounted reward:

$$\forall \pi \in HR, \quad v_\lambda^\pi(s) = \lim_{N\to\infty} E^\pi[\sum_{t=1}^{N} \lambda^{t-1} r(X_t, Y_t)]$$

Objective

- Value function:

$$v_\lambda^\star(s) = \sup_{\pi \in HR} v_\lambda^\pi(s)$$

Bellman's equations

- The value function *should* satisfy:

$$\forall s \in S, \quad v(s) = \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j)\}$$

- Bellman's equations provide a characterization of the value function

Theorem $v^\star = v^\star_\lambda$

Summary $v_\lambda^\star(s) = \sup_{\pi \in HR} v_\lambda^\pi(s)$

- $v_\lambda^\star(s)$ is the unique solution of Bellman's equations

$$\forall s \in S, \quad v(s) = \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j)\}$$

- Optimal stationary policies: $\pi = (\pi_1, \pi_1, ...) \in MD$

$$\forall s \in S, \quad \pi_1(s) \in \arg\max_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v_\lambda^\star(j)\}$$

- ε-optimal stationary policies: $\pi = (\pi_1, \pi_1, ...) \in MD$

$$\forall s \in S, \quad r(s, \pi_1(s)) + \lambda \sum_{j \in S} p(j|s, \pi_1(s)) v_\lambda^\star(j)$$

$$\geq \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v_\lambda^\star(j)\} - \epsilon$$

Solving Bellman's equations

- A way of solving Bellman's equations without the knowledge of the model (transition probabilities)
- In practice, when selecting an action a in state s , we move to a single random state s' where

$$s' \sim p(\cdot|s, a)$$

- Is there an ***online algorithm*** able to converge to the value function? (online: we just observe the sequences of states and actions)

Value iteration

- Algorithm

1. Fix $v_0 \in V$ $(V = \{v : S \to \mathbb{R}\})$. Fix $\epsilon > 0$.
2. Do until $\|v_{n+1} - v_n\| \leq \epsilon(1-\lambda)/2\lambda$: $v_{n+1} = \mathcal{L}v_n$

$$v_{n+1}(s) = \sup_{a \in A_s} \left(r(s,a) + \sum_{j \in S} p(j|s,a)\lambda v_n(j)\right)$$

Value iteration

- Algorithm

1. Fix $v_0 \in V$ $(V = \{v : S \to \mathbb{R}\})$. Fix $\epsilon > 0$.
2. Do until $\|v_{n+1} - v_n\| \leq \epsilon(1-\lambda)/2\lambda$: $v_{n+1} = \mathcal{L}v_n$

$$v_{n+1}(s) = \sup_{a \in A_s} \left(r(s,a) + \boxed{\sum_{j \in S} p(j|s,a)\lambda v_n(j)}\right)$$

Cannot be evaluated!

Q-values

- The Q-value: the maximum expected rewards starting from a given state and selecting a given action

$$q(s,a) = r(s,a) + \lambda \sum_j p(j|s,a) v(j)$$

- Q-values vs. value function: $v(s) = \max_{a \in A_s} q(s,a)$

$$q(s,a) = r(s,a) + \lambda \sum_j p(j|s,a) \max_{b \in A_s} q(j,b)$$

- Q-value iteration:

$$q_{n+1}(s,a) = r(s,a) + \lambda \sum_j p(j|s,a) \max_{b \in A_j} q_n(j,b)$$

Q-values

- The Q-values: the unique fixed point of *F*

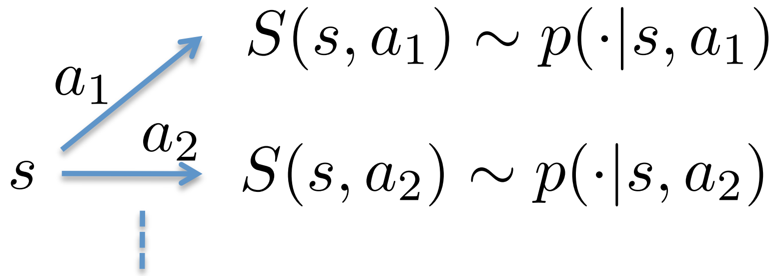
$$q(s,a) = r(s,a) + \lambda \sum_j p(j|s,a) \max_{b \in A_s} q(j,b)$$

$$F : \mathbb{R}^{S \times A} \to \mathbb{R}^{S \times A}$$

$$F(q)_{sa} = r(s,a) + \lambda E_{p(.|s,a)} \left[\max_{b \in A_J} q(J,b) \right]$$

Q-learning, expert version

- Expert: in a given state, you are able to sample the next state for all possible actions



$$S(s,a_1) \sim p(\cdot|s,a_1)$$

$$S(s,a_2) \sim p(\cdot|s,a_2)$$

- Algorithm: Initialize $q_0 \in \mathbb{R}^{S\times A}$

  Step *n*: $\forall s, a$

$$\begin{aligned} q_{n+1}(s,a) &= q_n(s,a) \\ &\quad + \alpha_n \left[r(s,a) + \lambda \max_b q_n(S_{n+1}(s,a),b) - q_n(s,a) \right] \end{aligned}$$

Q-learning, expert version

- Algorithm: Initialize $q_0 \in \mathbb{R}^{S \times A}$

  Step *n*: $\forall s, a$

$$
\begin{aligned}
q_{n+1}(s,a) &= q_n(s,a) \\
&\quad + \alpha_n \left[r(s,a) + \lambda \max_b q_n(S_{n+1}(s,a), b) - q_n(s,a) \right]
\end{aligned}
$$

- Convergence: $\sum_n \alpha_n = \infty, \quad \sum_n \alpha_n^2 < \infty$

  The algorithm approximates ODE: $\dot{q} = F(q) - q$

  Globally stable dynamical system (*F* is contractive)

Q-learning, bandit version

- Bandit: in a given state, you shave to select an action and you may observe the corresponding reward and new state

$$s \xrightarrow{a} S(s,a) \sim p(\cdot|s,a)$$

- Choose a stationary randomized policy arbitrarily such that:

$$P^{\pi}(Y_t = a | X_t = s) > 0, \quad \forall s, a$$

- In a given state, each action is explored an infinite number of times

Q-learning, bandit version

- Algorithm: Initialize $q_0 \in \mathbb{R}^{S\times A}, \quad s_0$

  Step *n*:

$$q_{n+1}(s_n,a_n) = q_n(s_n,a_n) \\ + \alpha_n(s_n,a_n)\left[r(s_n,a_n) + \lambda \max_b q_n(S_{n+1}(s,a),b) - q_n(s_n,a_n)\right]$$

- Convergence: $\forall s, a,$

$$\sum_n \alpha_n(s,a) = \infty, \quad \sum_n \alpha_n^2(s,a) < \infty$$

  The algorithm approximates ODE: $\dot{q} = F(q) - q$

  Convergence is slower than the expert version

Remark

- The Q-values: the unique fixed point of

$$F : \mathbb{R}^{S\times A} \to \mathbb{R}^{S\times A}$$

$$F(q)_{sa} = r(s,a) + \lambda E_{p(.|s,a)}\left[\max_{b\in A_J} q(J,b)\right]$$

- The value function: the unique fixed point of

$$G : \mathbb{R}^{S} \to \mathbb{R}^{S}$$

$$G(q)_s = \max_{a\in A_s}\left[r(s,a) + \lambda E_{p(\cdot|s,a)}[v(J)]\right]$$

Stochastic approximation is not directly applicable to the value function …

Remark

- The Q-values: the unique fixed point of

$$F : \mathbb{R}^{S\times A} \to \mathbb{R}^{S\times A}$$

$$F(q)_{sa} = r(s,a) + \lambda E_{p(.|s,a)}\left[\max_{b\in A_J} q(J,b)\right]$$

- The value function: the unique fixed point of

$$G : \mathbb{R}^{S} \to \mathbb{R}^{S}$$

$$G(q)_s = \max_{a\in A_s}\left[r(s,a) + \lambda E_{p(\cdot|s,a)}[v(J)]\right]$$

Stochastic approximation is not directly applicable to the value function …

Remark

- Algorithm (value iteration): a naïve tentative

$$v_{n+1}(s) = v_n(s) + \alpha_n \max_{a \in A_s} \left[r(s,a) + \lambda v(S_{n+1}(s,a) \right]$$

… it would approximate the behavior of ODE: $\dot{v} = H(v)$

$$H(v)_s = E\left[\max_{a \in A_s} \{ r(s,a) + \lambda v(S(s,a)) \} \right] - v(s)$$

$$H \neq G$$

Refs on Q-learning

- On the convergence of Stochastic Iterative Dynamical Programming Algorithms. T. Jaakkola, M. Jordan, S. Singh. MIT tech report, 1993
- The ODE method for convergence of stochastic approximation and reinforcement learning. V. Borkar, S. Meyn. SIAM J. Control Opt. 38-2, 2000
- Reinforcement learning. R. Sutton, A. Barto. MIT press, 1998.
- Reinforcement learning: a survey. L. Kaelbling, M. Littman, A. Moore. J. of Artificial Intelligence Research, 4, 1996.