RL 11: RL with Function Approximation ctd.

Michael Herrmann

University of Edinburgh, School of Informatics

23/02/2016

RL with function approximation: Points to remember

- $V_\theta(x) = \theta^\top \varphi(x)$, $\mathcal{Q}_\theta(x,a) = \theta^\top \varphi(x,a)$
- $\theta \in \mathbb{R}^N$, $\varphi(x) : \mathcal{X} \to \mathbb{R}^N$, $\varphi(x,a) : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^N$
- e.g. $V_\theta(x) = \sum_{i=1}^N \theta_i \frac{G\left(\|x - x^{(i)}\|\right)}{\sum_{m=1}^N G\left(\|x - x^{(m)}\|\right)}$
- TD(λ) with function approximation

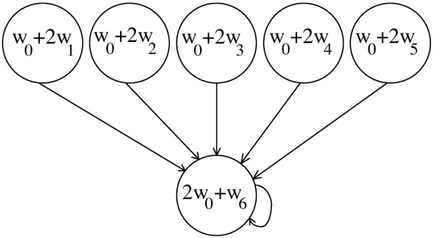
$$
\begin{aligned}
\delta_{t+1} &= r_{t+1} + \gamma \theta_t^\top \varphi(x_{t+1}) - \theta_t^\top \varphi(x_t) \\
z_{t+1} &= \varphi(x_t) + \lambda z_t \\
\theta_{t+1} &= \theta_t + \alpha_t \delta_{t+1} z_{t+1}
\end{aligned}
$$

- $\mathcal{Q}$-learning with function approximation

$$
\begin{aligned}
a_{t+1} &= \arg\max_a \theta_t^\top \varphi(x_t, a) \\
\delta_{t+1} &= r_{t+1} + \gamma \max_a \theta_t^\top \varphi(x_{t+1}, a) - \theta_t^\top \varphi(x_t, a_t) \\
\theta_{t+1} &= \theta_t + \alpha_t \delta_{t+1} \varphi(x_t, a_t)
\end{aligned}
$$

Function approximation: Baird's counter example

- Value of 6 states represented by 7 functions with weights w_i:



$$\Delta w_i = \eta \left(r + \gamma V_{\text{new}} - V_{\text{old}} \right) \frac{\partial V_{old}}{\partial w_i}$$

- Update every transition equally often
- If state 6 starts high, it climbs more often than falls.
- All states/weights diverge to $\pm\infty$

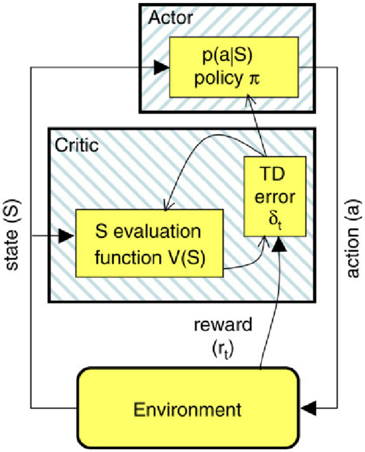
Baird, L. (1995) Residual algorithms: Reinforcement learning with function approximation. In Proc. 12. Int. Conf. on Machine Learning, pp. 30-37.

Today's topics

- Actor-Critic Methods (1981, see Barto, Sutton & Anderson, 1983)
- Parametrisation of the policy function: Policy gradient
- Compatible function approximation
- Natural actor-critic (NAC)

Actor-Critic Methods

- Actor aims at improving policy (adaptive search element)
- Critic evaluates the current policy (adaptive critic element)
- Learning is based on the TD error δ_t (usually on-policy)
- Reward only known to the critic
- Critic should improve as well



Actor-Critic Methods

- Policy (actor) is represented independently of the (state) value function (critic)
- Usually on-policy
- A number of variants exist, in particular among the early reinforcement learning algorithms, but also more recent ones

Advantages[1]

- AC methods require minimal computation in order to select actions which is beneficial in continuous cases, where search becomes a problem.
- They can learn an explicitly stochastic policy, i.e. learn the optimal action probabilities. Useful in competitive and non-Markov cases[2].

[1] Mark Lee following Sutton&Barto

[2] see, e.g., Singh, Jaakkola, and Jordan, 1994

Example: Policies for the inverted pendulum

- Exploitation (**actor**): Escape from low-reward regions as fast as possible
- aim at max. r
- e.g. Inverted pendulum task: Wants to stay near the upright position
- preferentially greedy and deterministic

- Exploration (**critic**): Find examples where learning is optimal
- aim at max. δ
- e.g. Inverted pendulum task: Wants to move away from the upright position
- preferentially non-deterministic

Critic-only methods and Actor-only methods

- Critic-only methods: Value function approximation and learning an approximate solution to the Bellman equation. Do not try to optimize directly over a policy space. May succeed in constructing a "good" approximation of the value function, yet lack reliable guarantees in terms of near-optimality of the resulting policy.
- Actor-only methods work with a parametrised family of policies. The gradient of the performance, with respect to the actor parameters, is directly estimated by simulation, and the parameters are updated in a direction of improvement.
  A possible drawback of such methods is that the gradient estimators may have a large variance. Furthermore, as the policy changes, a new gradient is estimated independently of past estimates. Hence, there is no "learning" in the sense of accumulation and consolidation of older information.

Konda, V. R., & Tsitsiklis, J. N. (1999). Actor-Critic Algorithms. In *NIPS* 13, 1008-1014.

See also Refs. 8, 10, 16, 23 therein.

Parametric policy

Approximation of the value function or action-value function using parametric function

$$\begin{aligned}\hat{V}_\theta(x) &\approx V(x)\\ \hat{Q}_\theta(x;a) &\approx Q(x;a)\end{aligned}$$

Policy can be generated directly from the value function e.g. using ε-greedy exploration

Today we will directly use a parametric function also to represent the policy

$$\pi_\omega(a|x) = \mathrm{Prob}[a|x]$$

Benefits: no worries about value function, uncertain state information or complexity arising from continuous states and actions

Problems: Needs good parametrisation. How to do exploration?

http://www.scholarpedia.org/article/Policy_gradient_methods

Reformulation of the goal of reinforcement learning

Maximise *global average of expected return* (cummulative reward)

$$\begin{aligned}\rho_{\mathcal{Q},\pi} &= \int_{\mathcal{X}}\int_{\mathcal{A}} \mathcal{Q}(x,a)\,\pi(a|x)\,\mu(x)\,\mathrm{d}a\,\mathrm{d}x \\ &= \int_{\mathcal{X}} \mu(x)\int_{\mathcal{A}} \mathcal{Q}(x,a)\,\pi(a|x)\,\mathrm{d}a\,\mathrm{d}x\end{aligned}$$

- ρ is equivalent to long-run average expected reward (if ergodic)
- μ is the (stationary) density of states, π is a stochastic policy

Function approximation for the value function and for the policy:

Maximisation over a restricted class of policies to prevent overfitting e.g. using policies π_ω parametrised by parameter vector $\omega \in \mathbb{R}^{d_\omega}$.

Perform stochastic gradient ascent on $\rho_{\mathcal{Q},\pi_\omega} =: \rho_\omega$ in order to find

$$\arg\max_\omega \rho_\omega \qquad \text{locally, using:} \qquad \omega_{t+1} = \omega_t + \beta_t \nabla_\omega \rho_\omega$$

where $\omega = (\omega_1, \ldots, \omega_M)^\top$ and ∇_ω is the gradient $\left(\frac{\partial}{\partial\omega_1}, \ldots, \frac{\partial}{\partial\omega_M}\right)^\top$

Reformulation of the goal of reinforcement learning

Another form for the *global average* of the expected reward:

$$
\begin{aligned}
\rho_{\pi_\omega} &= \sum_x \mu^{\pi_\omega}(x)\, V^{\pi_\omega}(x) \\
\rho_{\mathcal{Q},\pi_\omega} &= \sum_{x,a} \mu^{\pi_\omega}(x)\, \pi_\omega(a|x)\, \mathcal{Q}^{\pi_\omega}(x,a)
\end{aligned}
$$

In order to realise the policy gradient

$$\omega_{t+1} = \omega_t + \beta_t \nabla_\omega \rho_\omega$$

we will assume the dependency of μ and $\mathcal{Q}$ on ω to be "weak", i.e. use a simplifying assumption for the dependency of μ and $\mathcal{Q}$ on ω, namely

$$\nabla_\omega \rho(\omega) = \sum_{x,a} \mu^\pi(x) \left\{\nabla_\omega \pi_\omega(a|x)\right\} \mathcal{Q}^\pi(x,a)$$

A simplified example (to start with)

Consider only immediate reward (bandits with several "casinos")

$$\begin{aligned}
\rho_\omega &= \langle r(x,a) \rangle_{a,x} \\
&= \sum_x \mu(x) \sum_a \pi_\omega(a|x)\, r(x,a) \\
\nabla_\omega \rho_\omega &= \sum_x \mu(x) \sum_a \pi_\omega(a|x)\, \nabla_\omega \log \pi_\omega(a|x)\, r(x,a) \\
&= \sum_x \sum_a \left(\pi_\omega(a|x)\, \mu(x)\right) \nabla_\omega \log \pi_\omega(a|x)\, r(x,a) \\
&= \langle \nabla_\omega \log \pi_\omega(a|x)\;\; r(x,a) \rangle_{a,x}
\end{aligned}$$

The score function ($\nabla \log \pi$) comes into play by expressing the gradient as an average.

N.B.: $\frac{df(t)}{dt} = f(t)\,\frac{d \log f(t)}{dt}$

Score function

Let $\Psi_\omega : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{d_\omega}$ be the *score function* for π_ω, i.e.

$$\Psi_\omega (x, a) = \nabla_\omega \log \pi_\omega (a|x)$$

Score functions are also used in statistics (remember that $\pi (a|x)$ is a probability)

Example: Non-deterministic Gibbs- Boltzmann policies (for finite action space)

$$\pi_\omega (a|x) = \frac{\exp \left(\omega^\top \xi (x, a)\right)}{\sum_{a' \in \mathcal{A}} \exp \left(\omega^\top \xi (x, a)\right)}$$

ω are parameters and ξ are features (similar to θ and ψ for the approximation of the value function, but now it's for the policy)

$$\Psi_\omega (x, a) = \xi (x, a) - \sum_{a' \in \mathcal{A}} \pi_\omega \left(a'|x\right) \xi \left(x, a'\right)$$

Score function

Let $\Psi_\omega : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{d_\omega}$ be the *score function* for π_ω, i.e.

$$\Psi_\omega(x, a) = \frac{\partial}{\partial \omega} \log \pi(a|x)$$

Example: For infinite action space, Gaussian policies

$$\pi_\omega(a|x) = \frac{(2 \cdot 3.141..)^{-d_\omega/2}}{\sqrt{\det \Xi_\omega}} \exp\left(-\left(a - \omega \cdot g(x)\right)^\top \Xi_\omega^{-1} \left(a - \omega \cdot g(x)\right)\right)$$

The positive matrix $\Xi > 0$ is often simply a scaled version of the unit matrix, i.e. $\Xi = c\mathbf{I}$. Then, for $\omega = (\omega_1, \ldots, \omega_M)$,

$$\Psi_{\omega_i}(x, a) = -\left(c^{-1}\right)^\top \mathbf{I}\left(a - \omega \cdot g_\omega(x)\right) g_i(x)$$

... seems to provide us with reasonable gradients for typical policies

Does it work? The policy gradient theorem

Assume: Markov chain resulting from policy π_ω is ergodic for any ω

Estimate the gradient of ρ_ω

Policy gradient theorem (Bhatnagar et al., 2009)

$$\nabla_\omega \rho_\omega = \mathbb{E}_{x,a}\left[B\left(\omega\right)\right]$$

where

$$B\left(\omega\right) = \left(\mathcal{Q}^{\pi_\omega}\left(x,a\right) - h\left(x\right)\right)\Psi_\omega\left(x,a\right)$$

h an arbitrary bounded function (will be used later) that depends only on x and $\Psi_\omega\left(x,a\right)$ is the *score function* of the policy.

Instead of the expectation we will use a sample average $\langle\cdot\rangle$, i.e. a stochastic gradient version (i.e. following estimated gradient of ρ_ω)

$$\hat{\nabla}_\omega \rho_\omega = \left\langle B\left(\omega\right)\right\rangle$$

Adding a baseline

The introduction of a free function $h(x)$ is justified because

$$\sum_x \mu^\pi(x) \sum_a \nabla \pi(x,a)\, h(x) = \sum_x \mu^\pi(x)\, h(x)\, \nabla \sum_a \pi(x,a)$$
$$= \sum_x \mu^\pi(x)\, h(x)\, \nabla 1 = 0$$

so it does not affect the calculation of the gradient:

$$\nabla_\omega \rho(\omega) = \sum_x \mu^\pi(x) \sum_a \nabla_\omega \pi_\omega(a|x)\left(\mathcal{Q}^\pi(x,a) - h(x)\right)$$

How is the baseline h useful?

h may, e.g., represent a baseline for the value or express other constraints (see next slide)

Function approximation: Decoupling state value and policy

Features φ [used in the state-action value function] are to some extent arbitrary. Introduce orthogonality condition as additional constraint:

$$\sum_{a \in \mathcal{A}} \pi\left(a|x\right) \varphi\left(x, a\right) = 0$$

Using state features $\psi : \mathcal{X} \to \mathbb{R}^d$, perform a change of basis functions:

$$\mathcal{Q}_\theta\left(x, a\right) = \theta^\top \left(\psi\left(x\right) - \varphi\left(x, a\right)\right)$$

Then $V_\theta\left(x\right) = \sum_{a \in \mathcal{A}} \pi\left(a|x\right) \mathcal{Q}_\theta\left(x, a\right) = \theta^\top \psi\left(x\right)$

In the learning rule, set $V_{t+1} = V_\theta\left(x_{t+1}\right)$ which is now independent on the randomness of (non-deterministic) action choice

$\to$ lower variance

$\to$ better estimation of V

Back to the policy gradient theorem: Update rule for ω

Stochastic gradient of global reward average

$$\hat{\nabla}_\omega \rho_\omega = B(\omega)$$

where

$$B(\omega) = \left\langle \left(\mathcal{Q}^{\pi_\omega}(x, a) - h(x) \right) \Psi_\omega(x, a) \right\rangle$$

Typical (but not optimal) choice for h: $h = V^{\pi_{\omega_t}}$

$A(x, a) = \mathcal{Q}(x, a) - V(x)$ is sometimes called "advantage".

Now, form a stochastic gradient ascent on ρ

$$\omega_{t+1} = \omega_t + \beta_t B_t$$

β_t: decreasing learning rate (Robbins-Monro conditions!)

Depends on estimates of $\mathcal{Q}$. There are several ways to approximate.

REINFORCE (Williams, 1987)

Required are good estimates of Q and stationary samples of x and a

For episodic problems: Gradient ascent on the expected reward (MC!)

Update parameters at the end of each episode

$\rightarrow$ REINFORCE algorithms

In this way a direct policy search (without value functions) is possible

In non-episodic problems: two time-scales $\alpha \gg \beta$: make sure that the estimate $\hat{Q}$ is faster, i.e. can be assumed to have no bias, policy is changing slowly such that this is actually possible

Action-value Actor-Critic

Actor-critic algorithms maintain two sets of parameters (θ, ω), one (θ) for the representation of the value function and one (ω) for the representation of the policy.

Algorithm:

- Initialise x and ω, sample $a \sim \pi_\omega(\cdot|x)$
- Iterate:
  - obtain reward r, transition to new state x'
  - new action $a' \sim \pi_\omega(\cdot|x')$
  - $\delta = r + \gamma \mathcal{Q}_\theta(x', a') - \mathcal{Q}_\theta(x, a)$
  - $\omega = \omega + \beta \nabla_\omega \log \pi_\omega(a|x)\, \mathcal{Q}_\theta(x, a)$
  - $\theta = \theta + \alpha\delta \frac{\partial \mathcal{Q}}{\partial \theta}$
  - $a \leftarrow a'$, $x \leftarrow x'$
- Until termination criterion.

Variants of policy gradient

The policy gradient has many similar forms which are different realisations of the stochastic gradient w.r.t. to ρ

$$\begin{array}{rcll}
\nabla_\omega \rho_\omega^{(a)} & = & \langle \nabla_\omega \log \pi_\omega (a|x) \, \Sigma r_t \rangle & \text{REINFORCE} \\
\nabla_\omega \rho_\omega^{(b)} & = & \langle \nabla_\omega \log \pi_\omega (a|x) \, \mathcal{Q}_\theta (x,a) \rangle & \mathcal{Q} \text{ AC} \\
\nabla_\omega \rho_\omega^{(c)} & = & \langle \nabla_\omega \log \pi_\omega (a|x) \, A_\theta (x,a) \rangle & \text{advantage AC} \\
\nabla_\omega \rho_\omega^{(d)} & = & \langle \nabla_\omega \log \pi_\omega (a|x) \, \delta \rangle & \text{TD AC} \\
\nabla_\omega \rho_\omega^{(e)} & = & \langle \nabla_\omega \log \pi_\omega (a|x) \, \delta e \rangle & \text{TD}(\lambda) \text{ AC} \\
\tilde{\nabla}_\omega \rho_\omega^{(f)} & = & \theta & \text{natural AC}
\end{array}$$

AC: actor-critic

Bias and Variance in the Actor-Critic Algorithm

The approximation of the policy gradient introduces bias and variance. We need to be careful with the choice of the function approximation for $\mathcal{Q}$.

We will see in a moment that a compatibility of the representations of value function and policy is acheived, if we require

$$\nabla_\theta \mathcal{Q}_\theta = \nabla_\omega \log \pi_\omega$$

Consider minimal squared error when calculating ρ based on an approximation $\hat{\mathcal{Q}}^\pi(x, a; \theta)$ instead of the true $\mathcal{Q}^\pi(x, a)$

$$\epsilon^\pi(\theta) = \sum_{x,a} \mu^\pi(x) \left(\hat{\mathcal{Q}}^\pi(x, a; \theta) - \mathcal{Q}^\pi(x, a) \right)^2 \pi_\omega(a|x)$$

We want to show now that using the best (w.r.t. θ) approximation $\hat{\mathcal{Q}}^\pi(x, a; \theta)$ leaves the gradient of ρ (w.r.t to ω) unchanged.

S. Kakade (2001) A natural policy gradient. NIPS 14, 1531-1538.

Compatible function approximation

Use score function

$$\Psi_i(x,a)^\pi = \frac{\partial}{\partial \omega_i} \log \pi_\omega(a|x)$$

as basis functions, i.e. approximate of the state-action value function in terms of Ψ

$$\hat{\mathcal{Q}}^\pi(x,a;\theta) = \sum_i \theta_i \Psi_i^\pi(x,a) =: \theta \Psi^\pi(x,a)$$

This implies $\nabla_\theta \mathcal{Q}_\theta = \nabla_\omega \log \pi_\omega$. It is usually possible, but may not always be a good choice (consider e.g. Gaussian π_ω which give linear Ψ)

Consequences of the compatible function approximation

Minimisation of ϵ, i.e. $\frac{\partial \epsilon}{\partial \omega_i} = 0$, implies

$$\sum_{x,a} \mu^{\pi}(x)\,\Psi_i(x,a)^{\pi}\left(\hat{\mathcal{Q}}^{\pi}(x,a;\theta) - \mathcal{Q}^{\pi}(x,a)\right)\pi_{\omega}(a|x) = 0$$

or equivalently (this is what we wanted to show!)

$$\sum_{x,a}\mu^{\pi}(x)\Psi_i(x,a)^{\pi}\hat{\mathcal{Q}}^{\pi}(x,a;\theta)\pi_{\omega}(a|x)=\sum_{x,a}\mu^{\pi}(x)\Psi_i(x,a)^{\pi}\mathcal{Q}^{\pi}(x,a)\pi_{\omega}(a|x)$$

and in vector form using basis functions for $\hat{\mathcal{Q}}^{\pi} = \theta\Psi(x,a)^{\pi}$

$$\sum_{x,a}\mu^{\pi}(x)\Psi(x,a)^{\pi}\theta\Psi(x,a)^{\pi}\pi_{\omega}(a|x)=\sum_{x,a}\mu^{\pi}(x)\Psi(x,a)^{\pi}\mathcal{Q}^{\pi}(x,a)\pi_{\omega}(a|x)$$

Consequences of the compatible function approximation

$$\sum_{x,a}\mu^\pi(x)\Psi(x,a)^\pi\theta\Psi(x,a)^\pi\pi_\omega(a|x)=\sum_{x,a}\mu^\pi(x)\Psi(x,a)^\pi\mathcal{Q}^\pi(x,a)\pi_\omega(a|x)$$

By definition $\nabla_\omega\pi=\pi\Psi_i\left(x,a\right)^\pi$ because $\Psi_i(x,a)^\pi=\frac{\partial}{\partial\omega_i}\log\pi_\omega(a|x)$

$$\begin{aligned}\sum_{x,a}\mu^\pi(x)\Psi(x,a)^\pi\theta\Psi(x,a)^\pi\pi_\omega(a|x) &= \sum_{x,a}\mu^\pi(x)\mathcal{Q}^\pi(x,a)\nabla_\omega\pi_\omega(a|x)\\ &= \nabla_\omega\rho\left(\omega\right)\end{aligned}$$

Compare left hand side and

$$\begin{aligned}F\left(\omega\right)&=\mathbb{E}_{\mu^\pi(x)}\left[\mathbb{E}_{\pi_\omega(a|x)}\left[\frac{\partial\log\pi_\omega\left(a|x\right)}{\partial\omega_i}\frac{\partial\log\pi_\omega\left(a|x\right)}{\partial\omega_j}\right]\right]\\ &=\sum_{x,a}\mu^\pi(x)\pi_\omega(a|x)\,\frac{\partial\log\pi_\omega\left(a|x\right)}{\partial\omega_i}\frac{\partial\log\pi_\omega\left(a|x\right)}{\partial\omega_j}\\ &\sum_{x,a}\mu^\pi(x)\Psi(x,a)^\pi\Psi(x,a)^\pi\pi_\omega(a|x)\;\Rightarrow\;F\left(\omega\right)\theta=\nabla_\omega\rho\left(\omega\right)\end{aligned}$$

Gradient descent/ascent

Given an objective function, e.g. average undiscounted reward,

$$\rho_{\mathcal{Q},\pi,\mu} = \sum_{x\in\mathcal{X}} \sum_{a\in\mathcal{A}} \mu(x)\,\mathcal{Q}(x,a)\,\pi(a|x)\,,$$

depends (via π as well as $\mathcal{Q}$ and μ) on a vector of parameters ω.

Maximisation

$$\rho(\omega + d\omega) - \rho(\omega) \rightarrow \max \quad \text{for fixed } |d\omega|$$

$|d\omega|$ is the length of the $d\omega$, defined by $|d\omega|^2 = \sum_{ij} J_{ij}\omega_i\omega_j$

If $J = \{J_{ij}\}$ is the unit matrix, the length is given by the standard Pythagorean theorem $|d\omega|^2 = \sum_i \omega_i^2 \Rightarrow$ the geometry is Euclidean. The question: *Where on a small circle of radius $|d\omega|$ around ω the value of ρ is largest?* implies standard gradient ascent.

Idea: Use $J > 0$ to take shape of objective ρ into account.

Fisher Information

How take the shape of the objective into account?

$$\rho_{\mathcal{Q},\pi,\mu} = \sum_{x,a} \mu^{\pi_\omega}(x)\, \mathcal{Q}^{\pi_\omega}(x,a)\, \pi_\omega(a|x)$$

Assume the dependency of μ and $\mathcal{Q}$ on ω to be "weak", i.e.

$$\nabla_\omega \rho(\omega) = \sum_{x,a} \mu^{\pi}(x)\, \mathcal{Q}^{\pi}(x,a)\, \nabla_\omega \pi_\omega(a|x)$$

It can be shown that the solution is to choose J_{ij} as the inverse of

$$F_{ij}(x;\omega) = \mathbb{E}_{\pi_\omega(a|x)}\left[\frac{\partial \log \pi_\omega(a|x)}{\partial \omega_i} \frac{\partial \log \pi_\omega(a|x)}{\partial \omega_j}\right]$$

Remove state dependency by fixing ω and averaging over state distribution that are produced on the long run by the policy π_ω

$$F(\omega) = \mathbb{E}_{\mu^{\pi}(x)}\left[F_{ij}(x;\omega)\right]$$

Assuming this was correct we have now the natural gradient on ρ

$$d\omega \sim F(\omega)^{-1} \nabla\rho(\omega) = \eta \tilde{\nabla}\rho(\omega)$$

Pros and Cons of the Fisher information

+ "Natural" (*covariant*): uses the geometry of the goal function rather than the geometry of the parameter space (Choice of parameters used to be critical, but isn't any more so).
+ Related to Kullback-Leibler divergence and to Hessian
+ Describes efficiency in statistical estimation
+ Many applications in machine learning, statistics and physics
– Depends on parameters and is computationally complex
– Requires sampling of high-dimensional probability distribution
+ May still work if some approximation is used here: Integrate over a generic data distribution (e.g. Gaussian)

- Applying the natural gradient can be interpreted as a removal of any adverse effects of the particular architecture
- Another interpretation: Modified geometry: If $J > 0$ then all eigenvalues λ_k of this matrix are positive and $|d\omega|^2 = \sum_{ij} J_{ij}\omega_i\omega_j$ describes an ellipsoid with semi-axes λ_k

Natural actor-critic (NAC)

$$F\left(\omega\right)\theta = \nabla_{\omega}\rho\left(\omega\right) \Leftrightarrow \theta = F\left(\omega\right)^{-1}\nabla_{\omega}\rho\left(\omega\right) = \tilde{\nabla}_{\omega}\rho\left(\omega\right)$$

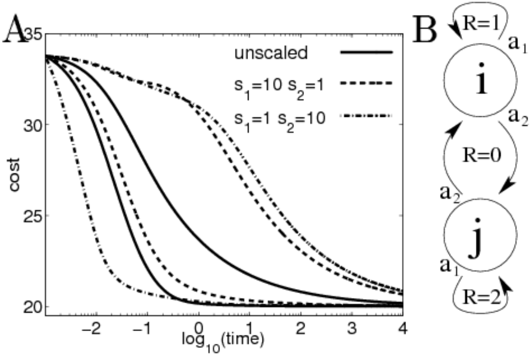
Learning rule (Kakade, 2001/2)

$$\omega_{t+1} = \omega_t + \beta_t\theta_t$$

Remarks:

- Natural gradient (S. Amari: Natural gradient works efficiently in learning, NC 10, 251-276, 1998)
- Examples by Bagnell and Schneider (2003) and Jan Peters (2003, 2008)

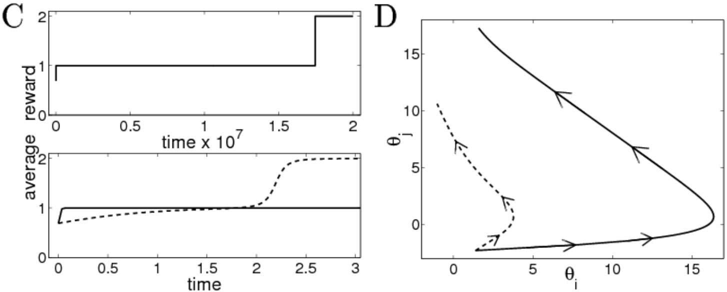
Kakade's Example



Three right curves: standard gradient, three left curves: natural gradient

Policy $\pi\left(a|x;\omega\right) \sim \exp\left(\omega_1 s_1 x^2 + \omega_2 s_2 x\right)$

Starting conditions: $\omega_1 s_1 = \omega_2 s_2 = -0.8$



Left: average reward for the policy
$\pi\left(a=1|s;\omega\right)\sim\exp\left(\omega\right)/\left(1+\exp\left(\omega\right)\right)$

Lower plot represents the beginning of the upper plot (different scales!): dashed: natural gradient, solid: standard gradient.

Right: Movement in the parameter space (axes are actually ω_i!)

Summary

- A systematic approach for continuous actions and space (time is discrete)
- Policy gradient as maximisation of the averaged state-action value
- Natural gradient leads to a very simple form
- Model-free reinforcement learning

Acknowledgements

Some material was adapted from web resources associated with Sutton and Barto's Reinforcement Learning book.

Today mainly based on C. Szepesvári: *Algorithms for RL*, Ch. 3.4.

See also: David Silber's Lecture 7: Policy Gradient