

**2) MDP**

**Policy Evaluation (PE)**: “the prediction prblm”

= *how* to compute v fn for arbitrary policy pi

**Value fn** (self-consistent, linear)

**1) Markov Process**

Tuple (S, P\_ss’) = (set of states, state transition proba. matrix)

Stationary/homogeneous chain: P\_ss’ only depends on s,s’ (not t)

Transient states (round) vs terminal states (square) 

Generates chain of **Markov states** governed by prob. transitions

Une image contenant texte, Police, écriture manuscrite, blanc

Description générée automatiquementUne image contenant croquis, cercle, dessin, diagramme

Description générée automatiquement

**Iterative PE Algo:** apply BEq to obtain better Vi(s)…Vk(s) estimates iteratively (until conv.): better approx. across steps + termination cond (largest diff in in v fn btw 2 iterations < threshold). Does full backup (all s’ considered).

**Backup Diagram:** transfer state value info from all s’ to s (update/backup op)

Une image contenant cercle, ligne

Description générée automatiquement

To compute v(s): average over all possible traces and their reward

Une image contenant texte, reçu, Police, blanc

Description générée automatiquement

**MRP**: Markov chain which emits rewards (S, P, R, gamma)

Une image contenant Police, typographie, texte, blanc

Description générée automatiquement

**Another version of Bellman Eq (BEq)**

Expected immediate reward collected upon departing s, at t+1



V^pi has

unique sol

Discount factor (see advt.)

Une image contenant Police, Graphique, symbole, logo

Description générée automatiquement

Une image contenant texte, Police, blanc, écriture manuscrite

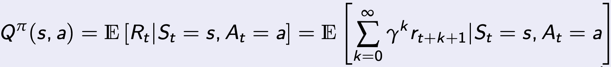
Description générée automatiquement

Value fns satisfy a set of recursive consistency eqs.

Return = total discounted reward from t

**State-Action value fn** (cost-to-go) **relationship btw Q and V**

Une image contenant Police, écriture manuscrite, blanc, calligraphie

Description générée automatiquementUne image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

**Ordering of policies: iff**

Une image contenant Police, typographie, conception

Description générée automatiquementUne image contenant Police, typographie, calligraphie, Graphique

Description générée automatiquementUne image contenant Police, symbole, logo, Graphique

Description générée automatiquementUne image contenant texte, Police, blanc

Description générée automatiquementUne image contenant Police, typographie, symbole, blanc

Description générée automatiquementUne image contenant Police, blanc, écriture manuscrite, texte

Description générée automatiquementUne image contenant Police, Graphique, logo, symbole

Description générée automatiquementUne image contenant Police, typographie, calligraphie, texte

Description générée automatiquement

**Optimal V\*:**

Deterministic: vs stochastic:

**Probabilistic/stochastic or deterministic policy**

**Bellman Eq = linear, self-consistent** => can solve **v** directly if small MRP (matrix inv.). **Iterative Methods** for larger MRPs: DP/MC/TD

If |S|=n, n eqs

If |S|=n, n-dim vector **v**

**Bellman Eq** for MRPs (start from above)

**State-value fn** = expected return R starting from state s at time t

Une image contenant Police, écriture manuscrite, calligraphie, texte

Description générée automatiquement

**Optimal pi\*:**

**Optimal Q\*:**



**B. Optimality Eqs for V\* and Q\***

if |S|=n, there are n V\* eqs; unique sol V\* indpt of pi.

**Backup diagram for Q^pi(s,a)**

Une image contenant texte, Police, écriture manuscrite, blanc

Description générée automatiquementUne image contenant ligne, cercle

Description générée automatiquement

Une image contenant texte, Police, écriture manuscrite, blanc

Description générée automatiquement

**BOEqs: *direct* sol to find pi\*** but hard + relies on assumptions: know env dynamics; comp resources; Markov property.

**BOEq Convergence Theorem** =>

Une image contenant texte, Police, capture d’écran, information

Description générée automatiquement

**3) DP (solve/find pi\* known MDPs)** Assume: finite MDP + know perfect model of env (P & R) + prblm w/ optimal substructure and overlapping subprblms. Simplify prblm by breaking it down into simpler subprblms recursively (Principle of Optimality; BEq as relation btw value of larger prblm & values of subprblms).

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementUne image contenant texte, diagramme, ligne, Police

Description générée automatiquementUne image contenant texte, Police, capture d’écran, blanc

Description générée automatiquementUne image contenant texte, Police, blanc, algèbre

Description générée automatiquementUne image contenant texte, Police, capture d’écran, blanc

Description générée automatiquementUne image contenant texte, Police, blanc

Description générée automatiquement

**4) MC (for V estimation)**

Learn from complete episodes of sample traces (no bootstrapping). V(s) = mean over empirical returns observed after visits to s (*not* expected R; will converge to it) = MC Policy Evaluation by sampling values for V. Incremental updates MC.

**Vanilla MC:** update pi *once* after X episodes. **Batch MC**: update pi every *batch\_size* episodes. **Online MC**: update pi *every* episode

**First-visit vs Every-visit MC:** record return of episode E from 1st occurrence vs every occurrence of s in E, for all s.

**Generalised PI Algo**

**Advt:**

**-** can do synchr (all states backed up in \\: 2 copies of V) or asynchr (1 copy of V, in-place; sig. reduced computation + still conv. guaranteed) updates

**-** bootstrapping: efficient use of data (thx optimal substructure

**Disv:**

**-** model-based

**-** curse of dimensionality (>mil. states)

**-** even 1 backup too costly

**Value Iteration (VI): PI w/** 1 iteration only of pi eval

BOEq = update rule applied iteratively (1-step look ahead)

**k = 1 => VI**

**Principle of Optimality**

**Policy Iteration (PI):** conv. guaranteed; eval⬄imprv

**Policy Improvement Theorem**

Une image contenant texte, Police, reçu, blanc

Description générée automatiquement

Une image contenant Police, écriture manuscrite, blanc, calligraphie

Description générée automatiquement

Une image contenant texte, Police, blanc, calligraphie

Description générée automatiquement

**5) TD (for V estimation)**

**Incremental update after each t**

**Unlike PI, no explicit policy pi:**



**Advt:**

**-** combines desirable properties of DP & MC: bootstrapping & sampling

**-** model-free

**-** learn directly from episodes of exp; works for non-episodic episodes (incomplete) too

**-** learn from incomplete episodes or w/o terminal state or before terminal state (online after every step).

**-** low variance: TD target (depends on one random A,P,R) is much lower variance than the sampled return (used in MC) that depends on *many* random A,P,R

**-** usually more efficient than MC

**-** TD, esp. TD(0), converges to V^pi(s)

**MC update at t**

**Alpha = lr**

Une image contenant texte, Police, blanc

Description générée automatiquement

**Advt:**

**-** model-free (no MDP knowledge needed: R/P) since R sampled

**-** fights curseOfDim through sampling (sample backups)

**-** cost of backups = constant (not exp.) + indept of N=|S|

**-** zero bias

**-** good convergence properties (even w/ FA)

**-** not v sensitive to initial value

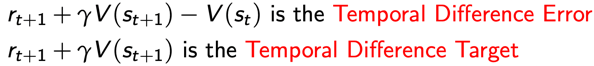
**-** v simple to understand and use

**Disv:**

**-** onlyepisodic MDPs w/ terminal states

**-** high variance: sampled R depends on many random A,P,R

Usually more effective in non-Markov env (no Markov assumed); TD exploits Markov property so TD more efficient in Markov env.



**Disv:**

**-** some bias: TD target is biased estimate of V^pi(s) as it relies on estimate of state s\_t+1 (bootstrapping)

**-** convergence not guaranteed w/ FA

**-** more sensitive to initial value than MC

**Bootstrapping = update involves an estimate**

**Sampling = update does not involve an expected value**

**6) Model-Free Control:** V/Q learned by MC/TD & follow GPI Algo w/ approximate V/Q and pi converging to pi\*

**MC Policy Improvement:** make pi (e-)greedy wrt current **Q fn** (**no** model needed to build greedy pi since Q, not V)

**Model-free prediction** (policy eval): **estimate** v fn of **unknown** MDP **Model-free control** (policy improv): **optimise** v fn **=> see 6)**

Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement

**TD control:** on-policy (**SARSA**) vs off-policy (**Q-learning**)

Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement

Apply TD to Q(S,A) + use e-greedy policy improv. + update every t

**SARSA**: Use GPI like in DP/MC, but eval./pred. w/ TD on Q (conv.)

For the approx. V/Q -> true V/Q and the Policy Improv. Theorem w/ MC to work, assume: inf. num of episodes + exploring starts (random s0). Then, MC can find pi\* given only sample episodes + no knowledge of env dynamics



**Q-learning** (w/ assumpt. of coverage): improve pi (greedy wrt Q(s,a)) **and** pi’ (e-greedy wrt Q(s,a)). Q-learning update (*R* = immediate r):

=> **avoid exploring starts assumption**: ensure agent continues to select them => on/off policy methods

**On-policy** (learn pi from exp sampled from pi) vs **Off-policy** (learn **target pi** from exp sampled from **behavior pi’**)

on-policy method -> soft policies -> epsilon-greedy policies)

Une image contenant texte, Police, écriture manuscrite, blanc

Description générée automatiquement

No explicit policies in algo: pi implicit in greedy term; pi’ = e-greedy version of pi. Both updated every *t* bc Q updated every *t*.

**TD(0)** here: immediate r + 1-step-look-ahead with Q(s’,a’)

e.g., on-policy first-visit MC control (for e-soft policies), batch or iterative learning for control



Une image contenant texte, Police, capture d’écran, algèbre

Description générée automatiquement

**7) Function Approx. (FA)**

**Prblm** (tabular RL): too many states/actions (CurseOfD) to

be storing values of V(S) and Q(S,A) as lookup tables.

**SOL**: estimate V/Q fn w/ FA. Generalise from seen states to

unseen states. Update param w using MC/TD learning.

**MC/TD for evaluation (V/Q FA)**

Pseudocode for MC learning **w**

Une image contenant texte, Police, blanc, algèbre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, algèbre

Description générée automatiquementUne image contenant Police, écriture manuscrite, blanc, typographie

Description générée automatiquement

FA: ANN, decision tree, nearest-neighbour…

**Assumpt**: differentiable fn, non-stationary & non-iid data

**GD goal:** find **w** minimizing MSE btw approx. Vhat(s,w) and

true V^pi(s) fn

Une image contenant Police, typographie, blanc, texte

Description générée automatiquementUne image contenant Police, blanc, ligne, texte

Description générée automatiquement

Une image contenant texte, ligne, Police, diagramme

Description générée automatiquementUne image contenant Police, blanc, ligne, texte

Description générée automatiquementUne image contenant Police, blanc, ligne, texte

Description générée automatiquement

**From eval. to control: FA in GPI**

Une image contenant Police, blanc, ligne, texte

Description générée automatiquement

**SGD** samples the gradient & the average update = full update



**States represented as hand-engineered feature vectors x(s)**

Une image contenant croquis, cercle, conception, art

Description générée automatiquement

**1) Coarse coding:** represent s w/ overlapping

binary features (if s in a circle, 1, else 0).

e.g. **tile coding** (suited for computer, efficient

on-line learning)

Update **rule** = lr x pred error x feature value **x**

MC/TD FA = **linear** V fn approx.:

Une image contenant Police, texte, blanc, écriture manuscrite

Description générée automatiquement

**Control = find v/q\***

Une image contenant Police, ligne, écriture manuscrite, blanc

Description générée automatiquement

**2) Radial Basis Fns (RBF)**: generalise coarse coding to continuous-valued features ([0,1]).

Une image contenant texte, Police, blanc

Description générée automatiquementUne image contenant texte, Police, blanc

Description générée automatiquement

**Engineering needed to solve Atari games w/ DQN (“regular Deep-Q learning”):**

**8) Deep-Q Learning**

**1) Experience Replay:** CNN overfitting latest experienced episodes. Inefficient use of interactive experience + highly correlated training samples (agent’s recent actions generated from recent policy outputted). **SOL**: experiences (traces) stored and replayed in mini-batches to train on more than just last episode (makes data more iid); instead of running Q-learning on each s-a pair as they occur, experience replay buffer stores the traces sampled. Exp reuse (random sampled) => data-efficient (higher learning speed) + avoid catastrophic forgetting of R associated w/ replayed transitions

**3) Clipping of Rewards:** all positive rewards = +1, all negative rewards = -1 to avoid diff reward scales making training unstable.

Use DL to replace hand-engineering of state space features w/ learning features from state data directly.

**DQN (Mnih et al., 2015) Atari:** Q(S,A) approx. by CNN w/ raw pixel inputs and discrete action outputs

**4) Skipping of Frames:** reduce comp. cost and accelerate training time + make the game run at speed comparable to human RT by only using every 4 video game frames as input (60Hz => 15Hz).

Une image contenant capture d’écran, bleu vert, printemps, conception

Description générée automatiquementUne image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

**DQN config** (49 Atari 2600 games)

Input: 84x84x4 preprocessed image. 3 conv. => 2 FC (ReLU) => FC output layer RMSProp, minibatch=32, e-greedy behav. pi (e=1->0.1)/Trained for 50million frames (38days game exp) + replay memory = 1million

**2) Target Network:** unstable training (e.g. resonance effect, divergence) bc bootstrapping a continuous state space repr. SOL: slow learning down (resonance dampener) using target network Q’. Initialise main network Q and target network Q’. Use Q’ to calculate TD-error. Infrequently set Q’=Q (params). Gives highly fluctuating Q time to settle (relaxation time) before updating Q’

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Une image contenant texte, Police, blanc, algèbre

Description générée automatiquement

Une image contenant texte, Police, blanc, algèbre

Description générée automatiquement

Une image contenant texte, Police, blanc, algèbre

Description générée automatiquement

**Prblm** w/ **regular Deep-Q learning**: **maximisation bias** = when taking max over all actions w/ (very) finite data, we may always **overestimate** values. **SOL: Double Q-learning (DDQN)**

DDQN (van Hasselt et al., 2017, AAAI): Use target network Q’ for estimating best action selection; regular Q for estimating Q-value of s-a pair (or converse). Reduces frequency by which max Q-value may be overestimated bc less likely that both networks are overestimating the same action.

Une image contenant texte, Police, capture d’écran

Description générée automatiquementUne image contenant texte, Police, capture d’écran

Description générée automatiquement

**DDQN learning**: more realistic (closer to final values), more stable, less biased (far less systematic overshooting).

Also see **Double Dueling Q networks** for Double-Q learning

**9) Policy Gradients. Policy-based methods:** find pi\* without V/Q fns. Faster convergence and often better for continuous and stochastic envs than **value-based methods** 1)-8).

**=** look directly at parametrised pi\_theta w/ params theta. Optimise by looking at traces that a policy pi\_theta rolls out to correlate them w/ the rewards they incur.

Probability to observe a trace tau depends on (the policy weights theta) of pi:

Une image contenant texte, Police, blanc, ligne

Description générée automatiquementUne image contenant Police, texte, blanc, typographie

Description générée automatiquement

**Hard to measure**

**Hard to model**

**pi\* obtained from theta\*** instead of V\*/Q\*, where theta\* = theta giving maximum average return.

**Rationale**: for continuous envs, infinite states/actions to estimate => value-based method = comp. expensive,

esp. w/ GPI where policy improv step needs full scan of action space (max over A). A value fn may be used to learn theta but is not required for action selection.

**Finite horizon**

**Infinite horizon**

Une image contenant texte, Police, capture d’écran, algèbre

Description générée automatiquementUne image contenant Police, texte, blanc, ligne

Description générée automatiquement****

**Approach 1: Finite Difference Grad *Approx.* (numerical)**

Can use any parametric supervised ML model to learn pi(a|s; learned theta).

Use **gradient ascent** as we want to max performance.

Une image contenant Police, écriture manuscrite, blanc, texte

Description générée automatiquement

Simple, noisy, inefficient, sometimes effective; works for arbitrary pi (even non-diff ones)

**Difficulty of computing policy gradient:**

Depends on traces tau: meaning we need derivatives on action selection and the stationary dist of states p(s), both determined by pi(a|s, theta). Given env is also generally unknown, difficult to estimate the effect of a policy update on state dist.

**Performance measure (episodic case)**

Une image contenant Police, texte, blanc, écriture manuscrite

Description générée automatiquement



**Approach 2: Direct Policy Gradients (using log trick)**

1. average return approx. by empirical mean over N traces

2. log trick to derive gradient

Une image contenant Police, ligne, texte, diagramme

Description générée automatiquementUne image contenant Police, texte, blanc, écriture manuscrite

Description générée automatiquement

PG = trial-and-error like MC learning

Une image contenant Police, typographie, calligraphie, blanc

Description générée automatiquement

**REINFORCE**: insightful first-shot at PGs (popularising it)

Une image contenant Police, écriture manuscrite, calligraphie, blanc

Description générée automatiquement

= one version of the **Policy Gradient Theorem (policy-centered equivalent of Bellman Theorem.** Implemented in **REINFORCE**

Une image contenant texte, Police, blanc

Description générée automatiquement

In the limit of large amounts of data => model will converge to theta\*

No bias but high variance in the sampled trajectories => difficult to stabilise theta. Any erratic journey can cause suboptimal shift in the policy dist. Reduce variance (being smarter correlating rewards with trajectories) by subtracting a baseline from reward term (keep values smaller) or using an advantage term (Schuman et al., 2016) (**see 10)**).

**10) Actor-Critic Methods** => model-learning to improve the model (!= direct RL improving V/Q/pi, as in 1-9)). A-C = improving V/Q/pi via model = **indirect/model-based RL => get better pi w/ fewer interactions**. Can have both indirect and direct methods => Split RL model into an **actor** (compute a based on s) and **critic** (produce Q-values of s,a = “model”).

Actor: input=s, output=a. Controls how agent behaves by learning pi\* (policy-based learning). Critic: evaluates a (value-based learning). The two models “compete” and each gets better at its role: key point = **combined arch learns better than the 2 separate networks would individually**.

**DDPG:** deterministic policy, model-free, off-policy, A-C. Continuous actions version of DQN. Uses bootstrapping to learn Q fn, and to learn pi from estimated Q fn. Explore in continuous space using Gaussian policy (~= discrete action e-greediness). Replay buffer, minibatch, target network (=DQN) control learning variability. MSBE. PG-based actor. Smooth updates to A&C target networks

Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

**Advantage A-C (A2C):** learn A, not Q:

Eval. of action a based on how much it *improves* s value. A2C reduces high var. of actor (=PG method) w/o adding bias, stabilises model in training

**Asynchronous (A3C):** multiple indpt agents (networks) interact w/ diff env copy in // => explore bigger part of S-A space in much less time. Trained in // and update periodically (asynchronously) a global network holding shared params. After each update, each agent copies global params + resume indpt exploring until next update (cool if large scale simulators). Info flow btw agents and global network, and btw agents (given param reset).



