Objective of MDP: maximize the expected sum of discounted rewards

Bayes-Adeptive MDP: augment the regular state of the agent with the information it has acquired about the dynamics.

Bayes-optimal behavior:
chooses actions that maximize expected reward as a function of belief-state

learning + planning ⇒ planning

environment : MDP. $= \langle S, A, T, R, \gamma \rangle$
$T: S \times A$ $R: S \times A$

Policy $\pi: S \to A$

~~Value V: function of A~~

optimal policy:

$$\pi^* = \arg\max_a Q(s,a)$$

$$Q(s,a) = E[R(s,a)] + \gamma \underset{s' \sim T(s,a)}{E}[V(s')]$$

$$V(s) = \max_a Q(s,a).$$

history H: every finite sequence of steps in the environment (MDP)

state space of the belief MDP: (belief-state)

1. $S = S \times H$ pairs real states with histories
2. grows exponentially with the length of the amount of history to be considered.

approximate Bayes-optimal behavior:
the value of each action selected is within ε of the value of the exact Bayes-optimal action.

Bayesian Sparse Sampling:
preferentially expands only promising parts of the tree by performing rollouts, or simulated trajectories, up to the specified depth.

Sparse Sampling:
recursively expanding a full search tree up to a certain depth d.

- At the root state, each of the A actions is chosen a constant number of times C. yielding a set of $A \cdot C$ children
- Once the tree is fully created, the leaves are each assigned a value of zero.
- Starting at the leaves, the values are backed up and combined via Bellman. giving the parents' values, until the root's value is determined.
- Nodes in the search tree: $(A \cdot C)^d$.

goal: addressing exploration vs exploitation dilemma

policy: action value function. (Q function)
$Q(S, a)$.

Q function satisfies Bellman equation.

$$Q(s_t, a_t) = E(r_t | s_t a_t) + \gamma E\left[\max_{a \in A} Q(s_{t+1}, a) | s_t a_t\right]$$

RL : learning policy.

② Value based: (learning + planning)
estimate the optimal Q-function directly Q-learning
from which a greedy policy is recovered.

policy based: (learning + planning)
estimate a good policy directly.

model based: (planning)
estimate the transition and reward models, and then determines a policy by solving the planning problem in the learned model

Classical action selection:

△ ε-greedy: with probability $1-\varepsilon$ choose the current best estimate $a^* = \arg\max_a \hat{Q}(s,a)$. Otherwise choose a random action $a \in A$

△ Boltzmann: Sample a random action according to $P(a|s) = \exp(\hat{Q}(s,a)/\tau)/Z$ where τ is a temperature parameter and Z is a normalization constant.

△ Interval estimation: Choose an action according to

$$a = \arg\max_a [\hat{Q}(s,a) + U(s,a)]$$

where $U(s,a)$ is a $(1-\delta)$ upper confidence interval on the point estimate $\hat{Q}(s,a)$.

Bayesian RL: model based.

a prior distribution is defined over transition and reward models: $P(\theta, \mu | s_0)$

given experience (history)

$s_0 a_0 r_0 s_1 \cdots s_t a_t r_t s_{t+1}$.

one determines the posterior distribution

$$P(\theta, \mu | s_0 a_0 r_0 s_1 \cdots s_t a_t r_t s_{t+1})$$

learning: updating the posterior

learning ⇒ planning (action selection)

A Bayesian approach to learning optimally in a MDP is essentially equivalent to solving a POMDP.

belief-state MDP
=
Bayes adaptive MDP.
=
Meta-level MDP.

state: b_t is given by a current base-level state s_t and a posterior distribution over the base-level transition model θ and reward model μ.

$$b_t = \langle P_t^\theta P_t^\mu s_t \rangle$$

$$\begin{cases} P_t^\theta = P(\theta | s_0 a_0 \cdots s_{t-1} a_{t-1} s_t) \\ P_t^\mu = P(\mu | s_0 a_0 r_0 \cdots s_{t-1} a_{t-1} r_{t-1}) \end{cases}$$

(3)

meta-level states

$b_t = \langle P_t^\theta P_t^\mu s_t \rangle$

$\|$

histories

$b_t \equiv s_0 a_0 r_0 \cdots s_{t-1} a_{t-1} r_{t-1} s_t$.

the only guaranteed way to approximate Bayes optimal action selection at a given belief state is to simulate the belief state MDP to the effective horizon.

State transition probability:

the probability of a particular history extension r_t, s_{t+1} given the current history $s_0 a_0 r_0 \cdots s_{t-1} a_{t-1} r_{t-1} s_t$ and action a_t.

meta-level belief states = base-level histories

reward:

$$P(r_t | P_t^\theta P_t^\mu s_t a_t) = \int_\mu P(r_t | s_t a_t \mu) P_t^\mu(\mu) d\mu$$

$$= P(r_t | s_0 \cdots s_t a_t)$$

transition:

$$P(P_{t+1}^\theta P_{t+1}^\mu s_{t+1} | P_t^\theta P_t^\mu s_t a_t)$$

$$= 1_{[P_{t+1}^\theta = P(\theta | s_0 a_0 \cdots s_{t+1})]} \cdot 1_{[P_{t+1}^\mu = P(\mu | s_0 \cdots s_t a_t r_t)]} \int_\theta P(s_{t+1} | s_t a_t \theta) P_t^\theta(\theta) d\theta \cdot \int_{r_t} \int_\mu P(r_t | s_t a_t \mu) P_t^\mu(\mu) d\mu dr_t.$$

$$= P(r_t s_{t+1} | s_0 \cdots s_t a_t)$$

Action selection strategy in Bayes-RL setting (myopic).

△ Thompson sampling:

given a current belief state $b_t = \langle P_t^\theta P_t^\mu s_t \rangle$.

sample a transition and reward model, θ and μ, from the belief state distributions P_t^θ and P_t^μ, solve for the optimal Q-function $Q_{\theta\mu}(s,a)$ for this model, then select the optimal action $a_t = \arg\max_{a \in A} Q_{\theta\mu}(s_t, a)$.

action selection strategy in Bayes-RL setting (optimal).

enumerating possible futures, averaging according to their realization probabilities, and choosing the best action.

Paper:
Support planning in domains with richly structured prior knowledge.

MDP: 5-tuple

$M = \langle S, A, P, R, \gamma \rangle$

S: states

A: actions

P: $S \times A$ transition probability.

R: $S \times A$ bounded reward function.

γ: discount factor

assume P unknown ("dynamics").

p is a latent variable subject to $P(p)$

observing a history of actions and states

$h_t = s_1 a_1 s_2 a_2 \cdots a_{t-1} s_t$

posterior belief on p updated using Bayes' rule.

$$P(p \mid h_t) \propto P(h_t \mid p) P(p).$$

(augmented transition)

$$P^+(\langle s, h \rangle, a, \langle s', h' \rangle) = \mathbb{1}[h' = h, a, s'] \int_p P(s, a, s') P(p \mid h) dp$$

(transition) (dynamics)

$$R^+(\langle s, h \rangle, a) = R(s, a).$$

(augmented reward)

augmented MDP: Bayes-adaptive MDP = belief state MDP : meta-level MDP.

$$M^+ = \langle S^+, A, P^+, R^+, \gamma \rangle$$

dynamics of the BAMDP are known. (P^+).

$$Q^*(\langle s_t, h_t \rangle, a) = \max_\pi E_\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \mid q_t = a \right]$$

1. use UCT to allocate search effort to promising branches of the state-action tree
2. use sample-based rollouts to provide value estimates at each node.
3. only sample a single transition model p^i from the posterior at the root of the search tree at the start of each simulation i.
4. use p^i to generate all the necessary samples during this simulation

essensially BAMDP is a POMDP

~~this root~~

☆ state node:

belief states : $\langle s, h \rangle$

☆ action nodes : a.

☆ visit counts: $N(\langle s, h \rangle)$ for state nodes

$N(\langle s, h \rangle, a)$ for action nodes

initialized to 0.

updated throughout search.

☆ Value. $Q(\langle s, h \rangle, a)$ initialized to 0.

maintained for each action node.

☆ each simulation traverses the tree without backtracking by following the UCT policy.

at state nodes defined by

$$\arg\max_a \; Q(\langle s,h\rangle,a) + C\sqrt{\frac{\log(N\langle s,h\rangle)}{N(\langle s,h\rangle,a)}}$$

⇓

a ⇒ traverse the tree

☆ given an action, the transition distribution $\hat{P}$ corresponding to the current simulation i is used to sample the next state

if the transition parameters for (o,ϕ) different states s and actions a are independent, we can completely forgo sampling a complete P, and instead draw any necessary parameters individually for each state-action pair

measure metrics in evaluation:

total undiscounted reward over many steps

while the paper optimizes for the discounted reward from the start state

Lazy Sampling

Sample P lazily, creating only the particular transition probabilities that are required as the simulation traverses the tree, and also during the rollout.

$P(s,a,\cdot)$ to be parametrized by a latent variable $\theta_{s,a}$ for each state and action pair depending on an additional set of latent variables ϕ.

$$P(\Theta|h) = \int_\phi P(\Theta|\phi,h)\,P(\phi|h)\,d\phi.$$

$$\Theta = \{\theta_{s,a} \mid s\in S,\ a\in A\}$$

$$\Theta_t = \{\theta_{s_1,a_1},\ldots,\theta_{s_t,a_t}\}.$$

$$P(\Theta|\phi,h) = P(\theta_{s_1,a_1}|\phi,h)\,P(\theta_{s_2,a_2}|\theta_1,\phi,h)$$
$$\cdots P(\theta_{s_T,a_T}|\Theta_{T-1},\phi,h)\,P(\Theta|\Theta_T,\Theta_T,\phi,h).$$

results

△ planning time, performance trade-off

△ performance scale as a function of planning time.

Compatible with arbitrary priors (P).

Motivation:

Solve large tasks with uncertainty about the dynamics

Want to take advantage of structured prior knowledge (sample latent param)

rewards with discounts. "transitions correlated"

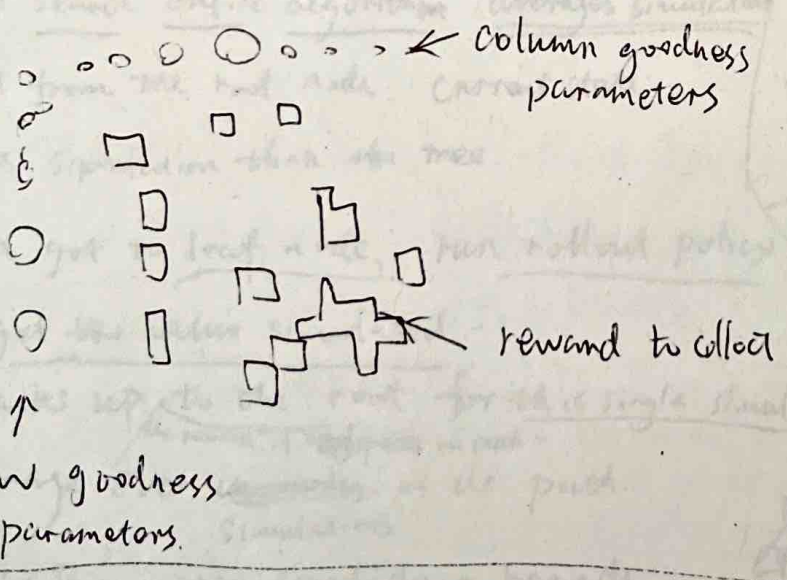
can plug in arbitrary prior to it.

formalism: Bayesian model-based RL

distribution over R, P

evaluate: infinite grid task.

2D



Typical MDP : $M = \langle S, A, P, R, \gamma \rangle$

but P is a latent variable with prior $P(P)$ (arbitrary, / prior knowledge)

Goal: Find exploration policy that maximizes expected sum of discounted rewards

$$\int_P P(P) \; \mathop{E}_{M(P)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, h_0 = s \right] dP$$

over all possible dynamics that can happen.

all possible transitions according to your posterior

Equivalent to solving an augmented MDP in belief space with known dynamics (P^+)

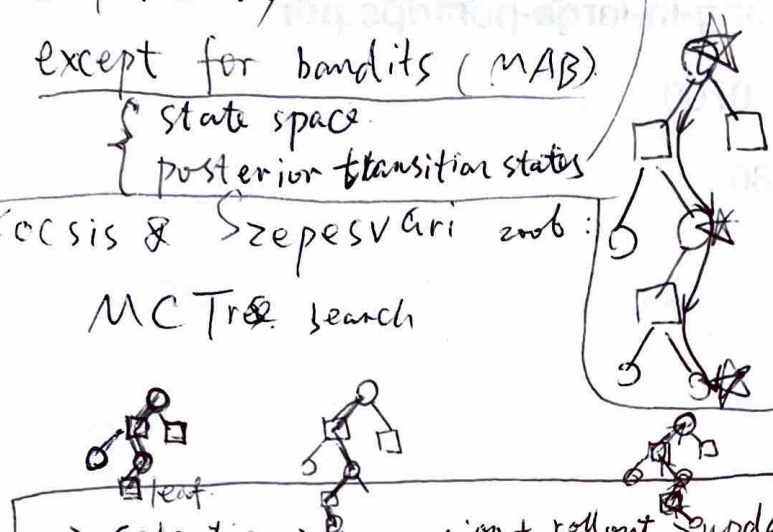
$$P^+(\langle s,h \rangle, a, \langle s',h' \rangle) = \mathbb{1}[h' = h,s] \int_P P(s,a,s') P(P|h) dP$$

previously: { don't scale
don't work with any prior
Dirichlet multinomial models

Major obstacle:
computationally intractable to solve.
except for bandits (MAB)
{ state space
posterior transition states

Bayes optimal BOP policy

Kocsis & Szepesvari 2006:
MCTree search



forward search online algorithm. averages simulation

△ start from the root node: current state
△ runs simulation thru the tree
△ when get to leaf node, run rollout policy to get the value simulated
△ backs up to the root for this single simulation. the reward of each nodes on path
△ average over the nodes on the path. simulations

UCB: upper confidence bounds.
UCT: UCB applied to trees

In one simulation △ treat each state node as a multi-armed bandit problem (use UCB1)

state node decides where to go in the tree.
put more effort to where it seems fruitful.
(can see most of the rewards)

Solution 1:
Bayesian adaptive
BA-UCT: leverage modern sample-based MDP solver:
MCTS/UCT applied to BAMDP.

Issue: expensive belief updates at every tree node.
multiple simulation, each with integrating over posterior and sample from transition posterior.
belief update
{ 100x simulations for each
needs to run many simulations
not feasible for generating priors
Monte Carlo.

Solution 2:
BA-UCT + Root Sampling
Silver & Veness 2010.
(belief update).
Sample P at root for each sim

{ one belief update per step
still Bayes-optimal in the limit
use this single P (sampled dynamics) throughout the sim.

Issue: wasteful sampling
each sim only requires a small subset of all parameters of P.
[factorization of the prior]

develop some conditional independence [only sample parameters we need

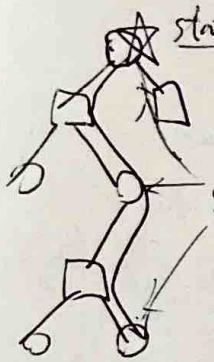
2895

Solution:

BA-UCT + RS. + Lazy Sampling (LS):

Sample MDP param on demand
based on current simulation.

Idea: ✓ minimize Sampling overhead, put more effort on search.



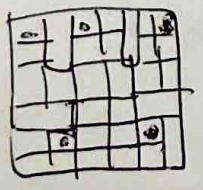
BA-UCT + RS + LS + Rollout Learning.

learn better rollout policy online.

→ Bayes-adaptive Monte Carlo planning.

1-step rollout: choose action $\pi_k(s_1)$ and simulate the system to get reward r_1, then we use this reward to update our estimate of the value function: $V_{k+1}(s_1) = (1-\gamma)V_k(s_1) + \gamma \cdot r_1$

eval:

① 

Dearden's Maze.
- 264 states
- Sparse Dirichlet-Multinomial prior.

② 2D-maze with infinite state space.

planning time ↑, return ↗

~~prior of transition distri~~ matters a lot.

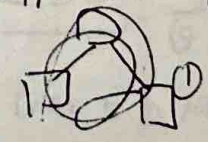
prior affects behavior and performance. (return, reward)

⑤ Standard domains.
- double-loop
- Grid 5
- Grid 10
- Dearden Maze.

BAMCP get best sum of rewards despite the fact that it optimizes for discounted rewards ~~infinite~~ sum of infinite.

planning time ↑, return — or ↓, for other algo because of over~~~~ optimistic

Appendix:



Solution 2:

① set up the MCMC at the root node.

② feed samples of P to UCT algorithm.