Thompson sampling for gridworld

Reinforcement Learning project

Lorenzo Basile, Irene Brugnara

Problem statement

- An agent moves in a discrete rectangular gridworld;
- In an unknown cell there is a source that emits a signal;
- The probability of sensing the signal decreases with the distance from the source;
- The agent's aim is to reach the source in the shortest time.

Modelling the problem as a POMDP

Set of states:

$$S = \{(x, y) | x \in [1, n_x], y \in [1, n_y]\}$$

Set of actions:

$$A = \{up, down, left, right, still\}$$

Model of environment:

$$p(s_t | s_{t-1}, a_{t-1}) = \mathbb{1}(s_t = s_{t-1} + a_{t-1})$$

Reward function:

$$r(s_{t-1}, a_{t-1}, s_t) = \mathbb{1}(s_t = s^*)$$

where s^* is the position of the source.

Goal:

$$\text{maximize} \quad \mathbb{E}\left(\sum_{t=1}^{\infty} \gamma^t r(s_{t-1}, a_{t-1}, s_t)\right)$$

Modelling the problem as a POMDP

Observation model:

$$f(y|s, s^*) = Bernoulli\left(\frac{1}{d(s, s^*) + 1}\right)$$

where d is the Manhattan distance, s is the current state and s^* is the source position.

Belief: discrete probability distribution over the state space for the source position

$$b(s) = Prob(s = s^*)$$

Bayesian belief update:

$$b^t(s|s_t, y) = \frac{f(y|s_t, s)b^{t-1}(s)}{\sum_{\bar{s} \in S} f(y|s_t, \bar{s})b^{t-1}(\bar{s})}$$

Thompson sampling

Thompson sampling is a heuristic for choosing actions to balance between exploration and exploitation.

It requires keeping a probability distribution over the unknown parameters of the MDP and performing bayesian updates of it as evidence is accumulated.
A policy is randomly selected according to the probability that it is optimal under the current posterior distribution. This approach has the advantage over a greedy strategy that it enhances exploration.

The algorithms

Algorithm 1: Thompson Gridworld

initialize $b^0(\cdot), \tau, s_0, s^*, t = 0$;
repeat
  sample $\hat{s} \sim b^t(\cdot)$;
  $i = 0$;
  while $s_t \neq \hat{s}$ *and* $i < \tau$ **do**
    choose a_t to get one step closer to $\hat{s}$;
    apply a_t;
    $i = i + 1,\ t = t + 1$;
    reach s_t;
    observe $y \sim f(\cdot | s_t, s^*)$;
    update $b^t(\cdot)$;
  end
until $r(s_{t-1}, a_{t-1}, s_t) = 1$;

Algorithm 2: Greedy Gridworld

initialize $b^0(\cdot), \tau, s_0, s^*, t = 0$;

repeat

choose $\hat{s} = \operatorname{argmax} b^t(\cdot)$;

$i = 0$;

while $s_t \neq \hat{s}$ *and* $i < \tau$ **do**

choose a_t to get one step closer to $\hat{s}$;

apply a_t;

$i = i + 1,\ t = t + 1$;

reach s_t;

observe $y \sim f(\cdot | s_t, s^*)$;

update $b^t(\cdot)$;

end

until $r(s_{t-1}, a_{t-1}, s_t) = 1$;

Deep exploration

The parameter τ is the exploration depth. When $\tau > 1$ we are performing *deep exploration*.

Once every τ steps a target position is sampled and the agent follows the optimal policy for this target position until next sampling (or until the sampled target is reached). This technique makes the algorithm more effective when the posterior distribution is multimodal and reduces the search time.

Performance assessment

Bayesian regret:

$$Regret = \mathbb{E}\left(\sum_{s \in S} \rho(s)\left(V_{\pi^*}(s) - V_{\pi}(s)\right)\right)$$

where ρ is the uniform initial state distribution and the expectation is taken over the prior target distribution (as well as the algorithm's randomisation); π is the policy implied by the algorithm and π^* is the optimal policy assuming the target position is known.

Performance assessment

Empirically, the regret is computed as

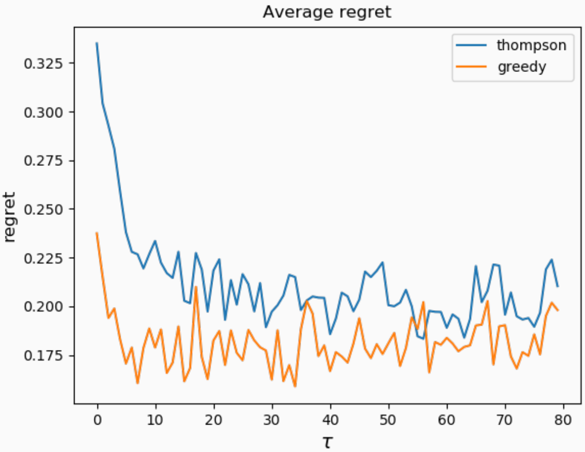
$$Regret = \frac{1}{m}\left(\sum_{i=1}^{m}\left(\gamma^{t^*} - \gamma^{t}\right)\right)$$

where m is the number of source and initial state configurations, t^* is the Manhattan distance between them and t is the number of steps taken by the algorithm.

Results

Parameters: $n_x = 40, n_y = 40, \gamma = 0.999, m = 100$

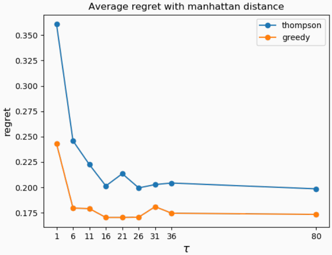
Average regret vs. exploration depth

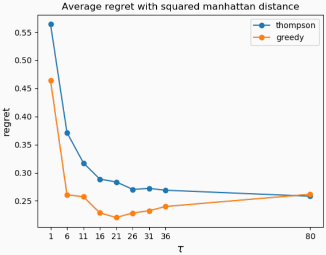


Results

Parameters: $n_x = 40, n_y = 40, \gamma = 0.999, m = 400$

Changing the dependency of the signal on distances:

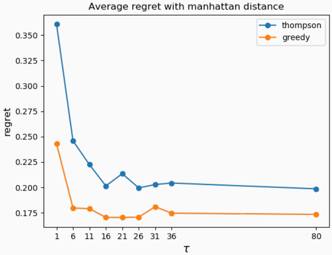


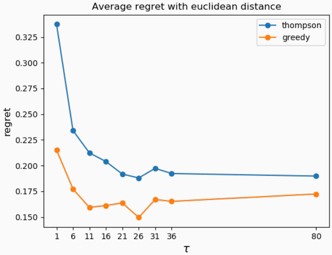


Results

Parameters: $n_x = 40, n_y = 40, \gamma = 0.999, m = 400$

Changing the distance metric:





Conclusions

- Both algorithms attain much better performance when some deep exploration is allowed ($\tau > 1$);
- Greedy algorithm generally solves the search problem in less time, especially with low exploration;
- Thompson sampling becomes more competitive when the likelihood of observations decreases more sharply with distance;
- Changing the distance metric does not particularly affect the behaviour of the algorithms.

References

- I. Osband, D. Russo, and B. Van Roy. "(More) efficient Reinforcement Learning via Posterior Sampling." arXiv:1306.0940, 2013
- D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen. "A tutorial on Thompson Sampling." arXiv:1707.02038, 2017
- M. Strens. "A Bayesian Framework for Reinforcement Learning." Proceedings of the Seventeenth International Conference on Machine Learning, 2000