Count-Based Exploration in Feature Space for Reinforcement Learning[1]

Marco Carollo

Reinforcement Learning
DSSC
Università degli Studi di Trieste

February 9, 2024

The problem at hand

If the state-action space of a Markov Decision Process is large, the agent will only visit a fraction of that space.

The problem at hand

If the state-action space of a Markov Decision Process is large, the agent will only visit a fraction of that space.
A key shortcoming of tabular MDB solutions is the inability to **generalize** what is learnt in one context to another.

The problem at hand

If the state-action space of a Markov Decision Process is large, the agent will only visit a fraction of that space.
A key shortcoming of tabular MDB solutions is the inability to **generalize** what is learnt in one context to another.
A possible solution is to estimate the value of **non-visited** states while using Linear Function Approximation (LFA).

The uncertainty framework

Let $\phi : \mathcal{S} \to \mathcal{T} \subset \mathbb{R}^M$ be the feature mapping from the state space into an M-dimensional feature space $\mathcal{T}$.

- States with less frequently observed features are deemed more uncertain.

The uncertainty framework

Let $\phi : \mathcal{S} \to \mathcal{T} \subset \mathbb{R}^M$ be the feature mapping from the state space into an M-dimensional feature space $\mathcal{T}$.

- States with less frequently observed features are deemed more uncertain.
- The uncertainty of a state influences its *generalized state-visit count*. Higher uncertainty is indicative of a less visited state.

The uncertainty framework

Let $\phi : \mathcal{S} \rightarrow \mathcal{T} \subset \mathbb{R}^M$ be the feature mapping from the state space into an M-dimensional feature space $\mathcal{T}$.

- States with less frequently observed features are deemed more uncertain.
- The uncertainty of a state influences its *generalized state-visit count*. Higher uncertainty is indicative of a less visited state.
- An exploration bonus is assigned for each visit; the more uncertain a state is, the higher the exploration bonus.

Implementation

A similarity measure between states is constructed. The more dissimilar to previously visited states, the more uncertain a state is.

Implementation

A similarity measure between states is constructed. The more dissimilar to previously visited states, the more uncertain a state is.

Instead of computing the similarity between the new state and all the history of visited states, a **density model** over the feature space is constructed: higher probabilities are assigned to states that share more features with more frequently observed states.

Implementation

A similarity measure between states is constructed. The more dissimilar to previously visited states, the more uncertain a state is.

Instead of computing the similarity between the new state and all the history of visited states, a **density model** over the feature space is constructed: higher probabilities are assigned to states that share more features with more frequently observed states.
This density model induces a **similarity measure** *on the feature space.*

The density model

Indicating with $N_t(\phi_i)$ the number of times ϕ_i[1] has occurred, a count-based estimation is used for the density ρ_t^i of each feature ϕ_i, at timestep t.

$$\rho_t^i(\phi_i) = \frac{N_t(\phi_i) + \frac{1}{2}}{t + 1}$$

[1] To simplify the notation, here ϕ_i is actually $\phi_i(s)$, with the observed state $s \in \mathcal{S}$.

The density model

Indicating with $N_t(\phi_i)$ the number of times ϕ_i[1] has occurred, a count-based estimation is used for the density ρ_t^i of each feature ϕ_i, at timestep t.

$$\rho_t^i(\phi_i) = \frac{N_t(\phi_i) + \frac{1}{2}}{t + 1}$$

The density model on the feature space is then defined as follows:

$$\rho_t(\phi(s)) = \prod_{i=1}^{M} \rho_t^i(\phi_i)$$

[1] To simplify the notation, here ϕ_i is actually $\phi_i(s)$, with the observed state $s \in \mathcal{S}$.

The ϕ-pseudocount

The density model is then connected to a more traditional idea of counting with the ϕ-pseudocount $\tilde{N}_t^{\phi}(s)$.

The ϕ-pseudocount

The density model is then connected to a more traditional idea of counting with the ϕ-pseudocount $\tilde{N}_t^{\phi}(s)$.
Naively, it's defined as:

$$\tilde{N}_t^{\phi}(s) = t \cdot \rho_t(\phi(s))$$

A state more similar to previously visited states will have a higher pseudocount.

Optimistic exploration

An exploration bonus is computed from the ϕ-pseudocount in the following way:

$$\mathcal{R}_t^{\phi}(s, a) = \frac{\beta}{\sqrt{\tilde{N}_t^{\phi}(s)}}$$

These bonuses are added to the estimated state/action value. Lower counts entail higher bonuses, so the agent is effectively *optimistic* about the value of less frequently visited regions of the environment. This drives the agent to visit states about which it is uncertain.

Key points

Some key points of this count-based exploration are:

Key points

Some key points of this count-based exploration are:

- computing a generalized state visit-count, which allows the agent to estimate the uncertainty associated with any state.

Key points

Some key points of this count-based exploration are:

- computing a generalized state visit-count, which allows the agent to estimate the uncertainty associated with any state.
- exploring in feature space rather than in the untransformed state space, resulting in a simpler and less computationally expensive method.

References

[1] Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning, 2017.

Appendix: Pseudocode

Algorithm 1 Reinforcement Learning with LFA and ϕ-EB.

Require: β, t_{end}

while $t < t_{\text{end}}$ **do**

  Observe $\boldsymbol{\phi}(s)$, r_t

  Compute $\rho_t(\boldsymbol{\phi}) = \prod_i^M \rho_t^i(\phi_i)$

  for i in $\{1,\ldots,\text{M}\}$ **do**

    Update ρ_{t+1}^i with observed ϕ_i

  end for

  Compute $\rho_{t+1}(\boldsymbol{\phi}) = \prod_i^M \rho_{t+1}^i(\phi_i)$

  Compute $\hat{N}_t^{\phi}(s) = \frac{\rho_t(\boldsymbol{\phi})(1-\rho_{t+1}(\boldsymbol{\phi}))}{\rho_{t+1}(\boldsymbol{\phi})-\rho_t(\boldsymbol{\phi})}$

  Compute $\mathcal{R}_t^{\phi}(s,a) = \frac{\beta}{\sqrt{\hat{N}_t^{\phi}(s)}}$

  Set $r_t^+ = r_t + \mathcal{R}_t^{\phi}(s,a)$

  Pass $\boldsymbol{\phi}(s)$, r_t^+ to RL algorithm to update $\boldsymbol{\theta}_t$

end while

return $\boldsymbol{\theta}_{t_{\text{end}}}$