Title: Exploration–exploitation dilemma

URL: https://en.wikipedia.org/wiki/Exploration%E2%80%93exploitation_dilemma

PageID: 73127816

Categories: Category:Cognition, Category:Decision theory, Category:Machine learning, Category:Optimal decisions, Category:Strategy

-----

The exploration–exploitation dilemma , also known as the explore–exploit tradeoff , is a fundamental concept in decision-making that arises in many domains. It is depicted as the balancing act between two opposing strategies. Exploitation involves choosing the best option based on current knowledge of the system (which may be incomplete or misleading), while exploration involves trying out new options that may lead to better outcomes in the future at the expense of an exploitation opportunity. Finding the optimal balance between these two strategies is a crucial challenge in many decision-making problems whose goal is to maximize long-term benefits.

Application in machine learning

In the context of machine learning, the exploration–exploitation tradeoff is fundamental in reinforcement learning (RL), a type of machine learning that involves training agents to make decisions based on feedback from the environment. Crucially, this feedback may be incomplete or delayed. The agent must decide whether to exploit the current best-known policy or explore new policies to improve its performance.

Multi-armed bandit methods

The multi-armed bandit (MAB) problem was a classic example of the tradeoff, and many methods were developed for it, such as epsilon-greedy, Thompson sampling , and the upper confidence bound (UCB). See the page on MAB for details.

In more complex RL situations than the MAB problem, the agent can treat each choice as a MAB, where the payoff is the expected future reward. For example, if the agent performs an epsilon-greedy method, then the agent will often "pull the best lever" by picking the action that had the best predicted expected reward (exploit). However, it would pick a random action with probability epsilon (explore). Monte Carlo tree search , for example, uses a variant of the UCB method.

Exploration problems

There are some problems that make exploration difficult.

Sparse reward. If rewards occur only once a long while, then the agent might not persist in exploring. Furthermore, if the space of actions is large, then the sparse reward would mean the agent would not be guided by the reward to find a good direction for deeper exploration. A standard example is Montezuma's Revenge .

Deceptive reward. If some early actions give immediate small reward, but other actions give later large reward, then the agent might be lured away from exploring the other actions.

Noisy TV problem. If certain observations are irreducibly noisy (such as a television showing random images), then the agent might be trapped exploring those observations (watching the television).

Exploration reward

This section based on.

The exploration reward (also called exploration bonus) methods convert the exploration-exploitation dilemma into a balance of exploitations. That is, instead of trying to get the agent to balance exploration and exploitation, exploration is simply treated as another form of exploitation, and the agent simply attempts to maximize the sum of rewards from exploration and exploitation. The exploration reward can be treated as a form of intrinsic reward .

We write these as r t i , r t e {\displaystyle r_{t}^{i},r_{t}^{e}} , meaning the intrinsic and extrinsic rewards at time step t {\displaystyle t} .

However, exploration reward is different from exploitation in two regards:

The reward of exploitation is not freely chosen, but given by the environment, but the reward of exploration may be picked freely. Indeed, there are many different ways to design r t i {\displaystyle r_{t}^{i}} described below.

The reward of exploitation is usually stationary (i.e. the same action in the same state gives the same reward), but the reward of exploration is non-stationary (i.e. the same action in the same state should give less and less reward).

Count-based exploration uses N n (s) {\displaystyle N_{n}(s)} , the number of visits to a state s {\displaystyle s} during the time-steps 1 : n {\displaystyle 1:n} , to calculate the exploration reward. This is only possible in small and discrete state space. Density-based exploration extends count-based exploration by using a density model ρ n (s) {\displaystyle \rho _{n}(s)} . The idea is that, if a state has been visited, then nearby states are also partly-visited.

In maximum entropy exploration , the entropy of the agent's policy π {\displaystyle \pi } is included as a term in the intrinsic reward. That is, r t i = − ∑ a π (a | s t) ln ■ π (a | s t) + ■ {\displaystyle r_{t}^{i}=-\sum _{a}\pi (a|s_{t})\ln \pi (a|s_{t})+\cdots } .

Prediction-based

This section based on.

The forward dynamics model is a function for predicting the next state based on the current state and the current action: f : (s t , a t) ■ s t + 1 {\displaystyle f:(s_{t},a_{t})\mapsto s_{t+1}} . The forward dynamics model is trained as the agent plays. The model becomes better at predicting state transition for state-action pairs that had been done many times.

A forward dynamics model can define an exploration reward by r t i = ■ f (s t , a t) − s t + 1 ■ 2 2 {\displaystyle r_{t}^{i}=\|f(s_{t},a_{t})-s_{t+1}\|_{2}^{2}} . That is, the reward is the squared-error of the prediction compared to reality. This rewards the agent to perform state-action pairs that had not been done many times. This is however susceptible to the noisy TV problem.

Dynamics model can be run in latent space . That is, r t i = ■ f (s t , a t) − φ (s t + 1) ■ 2 2 {\displaystyle r_{t}^{i}=\|f(s_{t},a_{t})-\phi (s_{t+1})\|_{2}^{2}} for some featurizer φ {\displaystyle \phi } . The featurizer can be the identity function (i.e. φ (x) = x {\displaystyle \phi (x)=x}), randomly generated, the encoder-half of a variational autoencoder , etc. A good featurizer improves forward dynamics exploration.

The Intrinsic Curiosity Module (ICM) method trains simultaneously a forward dynamics model and a featurizer. The featurizer is trained by an inverse dynamics model, which is a function for predicting the current action based on the features of the current and the next state: g : (φ (s t) , φ (s t + 1)) ■ a t {\displaystyle g:(\phi (s_{t}),\phi (s_{t+1}))\mapsto a_{t}} . By optimizing the inverse dynamics, both the inverse dynamics model and the featurizer are improved. Then, the improved featurizer improves the forward dynamics model, which improves the exploration of the agent.

Random Network Distillation (RND) method attempts to solve this problem by teacher–student distillation . Instead of a forward dynamics model, it has two models f , f ′ {\displaystyle f,f'} . The f ′ {\displaystyle f'} teacher model is fixed, and the f {\displaystyle f} student model is trained to minimize ■ f (s) − f ′ (s) ■ 2 2 {\displaystyle \|f(s)-f'(s)\|_{2}^{2}} on states s {\displaystyle s} . As a state is visited more and more, the student network becomes better at predicting the teacher. Meanwhile, the prediction error is also an exploration reward for the agent, and so the agent learns to perform actions that result in higher prediction error. Thus, we have a student network attempting to minimize the prediction error, while the agent attempting to maximize it, resulting in exploration.

The states are normalized by subtracting a running average and dividing a running variance, which is necessary since the teacher model is frozen. The rewards are normalized by dividing with a running variance.

Exploration by disagreement trains an ensemble of forward dynamics models, each on a random subset of all (s t , a t , s t + 1) {\displaystyle (s_{t},a_{t},s_{t+1})} tuples. The exploration reward is the variance of the models' predictions.

Noise

For neural network –based agents, the NoisyNet method changes some of its neural network modules by noisy versions. That is, some network parameters are random variables from a probability distribution. The parameters of the distribution are themselves learnable. For example, in a linear layer y = W x + b {\displaystyle y=Wx+b} , both W , b {\displaystyle W,b} are sampled from Gaussian distributions N (μ W , Σ W) , N (μ b , Σ b) {\displaystyle {\mathcal {N}}(\mu _{W},\Sigma _{W}),{\mathcal {N}}(\mu _{b},\Sigma _{b})} at every step, and the parameters μ W , Σ W , μ b , Σ b {\displaystyle \mu _{W},\Sigma _{W},\mu _{b},\Sigma _{b}} are learned via the reparameterization trick .

References

Amin, Susan; Gomrokchi, Maziar; Satija, Harsh; Hoof, van; Precup, Doina (September 1, 2021). "A Survey of Exploration Methods in Reinforcement Learning". arXiv : 2109.00157 [cs.LG].

v

t

e

Ambiguity aversion

Bounded rationality

Choice architecture

Expected utility

Expected value

Hyperbolic discounting

Leximin

Loss aversion

Multi-attribute utility

Path dependence

Principle of indifference

Prospect theory

Rational choice theory

Risk aversion

Risk-seeking

Satisficing

Strategic dominance

Subjective expected utility

Sure-thing

Utility theorem

Anscombe-Aumann framework

Causal decision

Decision field theory

Emotional choice

Evidential decision

Fuzzy-trace theory

Intertemporal choice

Naturalistic decision

Normative model

Quantum cognition

Recognition-primed decision

Rubicon model

Savage's subjective expected utility model

Analytic hierarchy process

Analytic network process

Cost–benefit analysis

Cost-effectiveness analysis

Cost–utility analysis

Decision conferencing

Decision curve analysis

Decision rule

Decision support system

Decision table

Decision tree

Decision matrix

Decisional balance sheet

Gittins index

Influence diagram

Minimax

MCDA

Scoring rule

Value of information perfect sample uncertainty

perfect

sample

uncertainty

Allais paradox

Certainty effect

Cognitive bias

Decoy effect

Disposition effect

Ellsberg paradox

Endowment effect

Framing effect

Heuristics

Newcomb's paradox

Pseudocertainty effect

Reference dependence

Regret

St. Petersburg paradox

Status quo bias

Sunk cost

Deep uncertainty

Exploration–exploitation

Info-gap

Pignistic probability

Robust decision-making

Behavioral economics

Game theory

Operations research

Social choice theory

Utility theory

David Blackwell

Bruno de Finetti

Morris H. DeGroot

Peter C. Fishburn

Gerd Gigerenzer

Itzhak Gilboa

Daniel Kahneman

R. Duncan Luce

Oskar Morgenstern

Howard Raiffa

Leonard J. Savage

David Schmeidler

Herbert Simon

Amos Tversky

John von Neumann