Title: Upper Confidence Bound

URL: https://en.wikipedia.org/wiki/Upper_Confidence_Bound

PageID: 80260162

Categories: Category:Algorithms, Category:Machine learning, Category:Statistical algorithms

---

Upper Confidence Bound (UCB) is a family of algorithms in machine learning and statistics for solving the multi-armed bandit problem and addressing the exploration–exploitation trade-off. UCB methods select actions by computing an upper confidence estimate of each action's potential reward, thus balancing exploration of uncertain options with exploitation of those known to perform well. Introduced by Auer, Cesa-Bianchi & Fischer in 2002, UCB and its variants have become standard techniques in reinforcement learning , online advertising , recommender systems , clinical trials , and Monte Carlo tree search . [1] [2] [3]

Background

The multi-armed bandit problem models a scenario where an agent chooses repeatedly among K options ("arms"), each yielding stochastic rewards, with the goal of maximizing the sum of collected rewards over time. The main challenge is the exploration–exploitation trade-off: the agent must explore lesser-tried arms to learn their rewards, yet exploit the best-known arm to maximize payoff. [3] Traditional ε-greedy or softmax strategies use randomness to force exploration; UCB algorithms instead use statistical confidence bounds to guide exploration more efficiently. [2]

The UCB1 algorithm

UCB1, the original UCB method, maintains for each arm i:

the empirical mean reward _μ■ i _,

the count _n i _ of times arm i has been played.

At round _t_, it selects the arm maximizing:

U C B 1 i (t) = μ ^ i + 2 ln ■ t n i {\displaystyle \mathrm {UCB1} _{i}(t)\;=\;{\hat {\mu }}_{i}\;+\;{\sqrt {\frac {2\ln t}{n_{i}}}}\,}

Arms with _n i =0_ are initially played once. The bonus term √(2 ln t / n i) shrinks as _n i _ grows, ensuring exploration of less-tried arms and exploitation of high-mean arms. [1]

Pseudocode

Theoretical properties

Auer et al. proved that UCB1 achieves **logarithmic regret**: after _n_ rounds, the expected regret _R(n)_ satisfies

R (n) = O (∑ i : Δ i > 0 ln ■ n Δ i) , {\displaystyle R(n)=O{\Bigl (}\sum _{i:\Delta _{i}>0}{\frac {\ln n}{\Delta _{i}}}{\Bigr)},}

where _Δ i _ is the gap between the optimal arm's mean and arm _i_'s mean. Thus, average regret per round → 0 as _n_→∞, and UCB1 is near-optimal against the Lai-Robbins lower bound. [1] [4]

Variants

Several extensions improve or adapt UCB to different settings:

UCB2

Introduced in the same paper, UCB2 divides plays into epochs controlled by a parameter α, reducing the constant in the regret bound at the cost of more complex scheduling. [1]

UCB1-Tuned

Incorporates empirical variance _V i _ to tighten the bonus: μ ^ i + ln ■ t n i min { 1 / 4 , V i } . {\displaystyle {\hat {\mu }}_{i}+{\sqrt {{\frac {\ln t}{n_{i}}}\min\{1/4,\,V_{i}\}}}.} This often outperforms UCB1 in practice but lacks a simple regret proof. [1]

KL-UCB

Replaces Hoeffding's bound with a Kullback–Leibler divergence condition, yielding asymptotically optimal regret (constant = 1) for Bernoulli rewards. [5] [6]

Bayesian UCB (Bayes-UCB)

Computes the (1−δ)-quantile of a Bayesian posterior (e.g. Beta for Bernoulli) as the index. Proven asymptotically optimal under certain priors. [7]

Contextual UCB (e.g., LinUCB)

Extends UCB to contextual bandits by estimating a linear reward model and confidence ellipsoids in parameter space. Widely used in news recommendation. [8]

Applications

UCB algorithms' simplicity and strong guarantees make them popular in:

Online advertising & A/B testing : adaptively allocate traffic to maximize conversion rates without fixed split ratios. [3]

Monte Carlo Tree Search : UCT uses UCB1 at each tree node to guide exploration in games like Go. [9] [10]

Adaptive clinical trials : assign patients to treatments with highest upper confidence on success, improving outcomes over randomization. [11]

Recommender systems : personalized content selection under uncertainty.

Robotics & control : efficient exploration of unknown dynamics.

See also

Multi-armed bandit

Reinforcement learning

Monte Carlo tree search

References