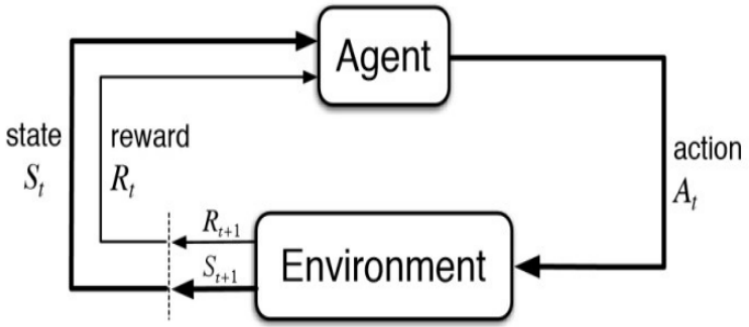
RL Reliability

Ref

Measuring the Reliability of Reinforcement Learning Algorithms
(https://arxiv.org/abs/1912.05663)

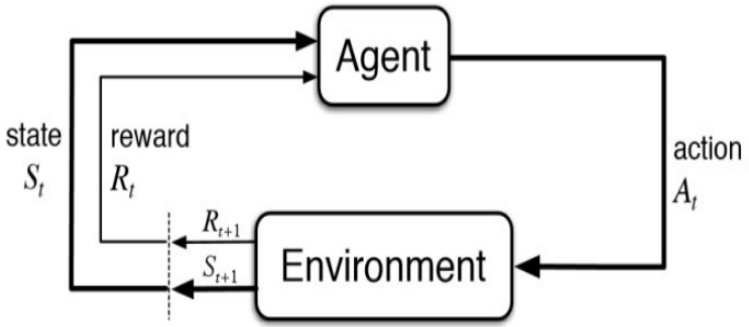
Reinforcement Learning(RL)



Reinforcement Learning(RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

- both supervised and reinforcement learning use mapping between input and output,
- supervised learning where feedback provided to the agent is correct set of actions for performing a task,
- reinforcement learning uses rewards and punishment as signals for positive and negative behavior.

Reinforcement Learning(RL)



- the goal in unsupervised learning is to find similarities and differences between data points,
- In reinforcement learning the goal is to find a suitable action model that would maximize the total cumulative reward of the agent.

Elements of a RL problem

Environment: Physical world in which the agent operates

State: Current situation of the agent

Reward: Feedback from the environment

Policy: Method to map agent’s state to actions

Value: Future reward that an agent would receive by taking an action in a particular state

Markov Decision Processes (MDPs) are mathematical frameworks to describe an environment in reinforcement learning and almost all RL problems can be formalized using MDPs.

An MDP consists of:

- a set of finite environment states S,
- a set of possible actions A(s) in each state,
- a real valued reward function R(s) and a transition model P(s', s | a).

However, real world environments are more likely to lack any prior knowledge of environment dynamics. Model-free RL methods come handy in such cases.

Model-free RL

Q-learning is a commonly used model free approach.

It revolves around the notion of updating Q values which denotes value of doing action *a* in state *s*.

The value update rule is the core of the Q-learning algorithm.

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Reproducibility Problem in Reinforcement Learning

[Robotics Case Study: https://arxiv.org/pdf/1909.03772.pdf]

Problems

Common issues when replicating ML research:

Omission of one or more hyperparameter choices in the manuscript.:

- Often hyper-parameters have significant impacts on how the algorithm performs so it is critical to report their values including how they were obtained.
- Reason for neglecting to report hyper-parameters: a) they are not considered important or b) their value is simply the default value, specified by the underlying implementation used.
- Both reasons are obviously important challenges to handle but are often hard to discover and subsequently enforce.

Problems

RL methods require large amounts of data:

- Results from the real world are thus expensive to obtain.
- Further, many industrial researchers are forced by their company's legal department to omit specific details to remain in front of their competitors

Problems

- differences in evaluation metrics
- lack of significance testing in the field of DRL

Effects:

- misleading reporting of results
- **With no statistical evaluation of the results, it is difficult to conclude if there are meaningful improvements.**
- If results are to be trusted, complete and statistically correct evaluations of proposed methods are needed.

Requirements

- reliability metrics
- practical recommendations for statistical tests to compare metric results

RELIABILITY METRICS: AXES OF VARIABILITY

1. During training: Across Time (T)

- In the setting of evaluation during training, one desirable property for an RL algorithm is to be **stable "across time"** within each training run.
- In general, smooth monotonic improvement is preferable to noisy fluctuations around a positive trend, or unpredictable swings in performance.

2. During training: Across Runs (R)

- During training, RL algorithms should have easily and consistently reproducible performances across multiple training runs.

AXES OF VARIABILITY

3. After learning: Across rollouts of a fixed policy (F)

When evaluating a fixed policy, a natural concern is the variability in performance across multiple rollouts of that fixed policy.

Each rollout may be specified e.g. in terms of a number of actions, environment steps, or episodes.

MEASURES OF VARIABILITY

Two kinds of measures: dispersion and risk.

[Dispersion: width of the distribution.]

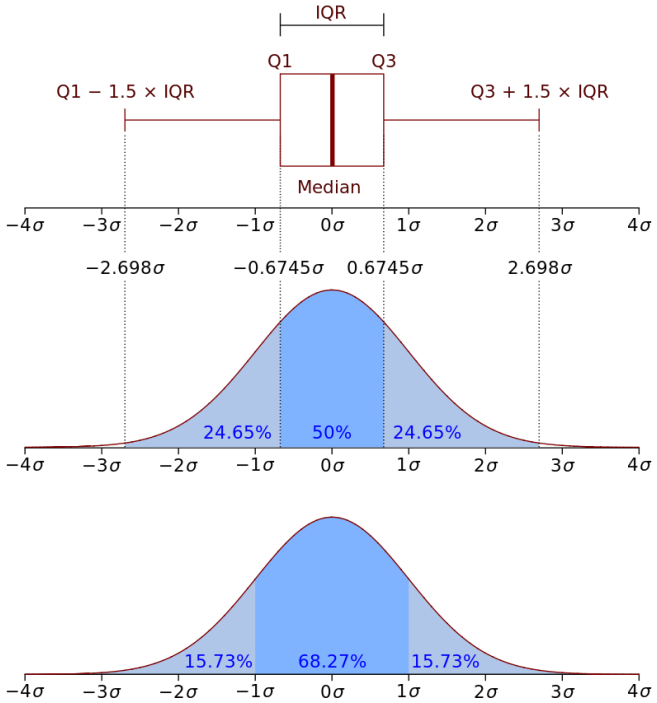
To measure dispersion:

- Inter-quartile range (IQR) (i.e. the difference between the 75th and 25th percentiles)
- Median absolute deviation (MAD).
- prefer to use IQR over MAD, because it is more appropriate for asymmetric distributions

Quartile and IQR

In statistics, a **quartile** is a type of quantile which divides the number of data points into four parts, or *quarters*, of more-or-less equal size. The data must be ordered from smallest to largest to compute quartiles; as such, quartiles are a form of order statistic. The three main quartiles are as follows:

- The first quartile (Q_1) is defined as the middle number between the smallest number (minimum) and the median of the data set. It is also known as the *lower* or *25th empirical* quartile, as 25% of the data is below this point.
- The second quartile (Q_2) is the median of a data set; thus 50% of the data lies below this point.
- The third quartile (Q_3) is the middle value between the median and the highest value (maximum) of the data set. It is known as the *upper* or *75th empirical* quartile, as 75% of the data lies below this point.



Median absolute deviation

For a univariate data set $X_1, X_2, \ldots, X_n$, the MAD is defined as the median of the absolute deviations from the data's median $\tilde{X} = \operatorname{median}(X)$:

$$\operatorname{MAD} = \operatorname{median}(|X_i - \tilde{X}|)$$

[Univariate is a term commonly used in statistics to describe **a type of data which consists of observations on only a single characteristic or attribute**.]

Example:

Consider the data (1, 1, 2, **2**, 4, 6, 9). It has a median value of 2. The absolute deviations about 2 are (1, 1, 0, 0, 2, 4, 7) which in turn have a median value of 1 (because the sorted absolute deviations are (0, 0, 1, **1**, 2, 4, 7)). So the median absolute deviation for this data is 1.

Risk

- In many cases, we are concerned about the worst-case scenarios.
- Therefore, we define risk as the heaviness and extent of the lower tail of the distribution.
- This is complementary to measures of dispersion like IQR, which cuts off the tails of the distribution.
- To measure risk, we use the Conditional Value at Risk (CVaR), also known as “expected shortfall".

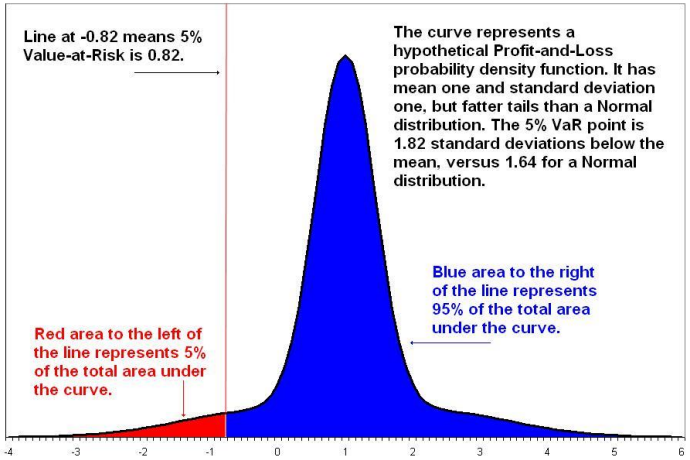
Conditional Value at Risk (CVaR)

- CVaR measures the expected loss in the worst-case scenarios, defined by some quantile α.
- It is computed as the expected value in the left-most tail of a distribution
- definition for the CVaR of a random variable X for a given quantile α:

$$\mathbf{CVaR}_\alpha(X) = \mathbb{E}\left[X | X \leq VaR_\alpha(X)\right]$$

where α ∈ (0, 1) and the VaR_α (Value at Risk Cut Off) is just the α-quantile of the distribution of X.

- Originally developed in finance, CVaR has also seen recent adoption in Safe RL as an additional component of the objective function.



METRIC DEFINITIONS

Dispersion across Time (DT): IQR across Time

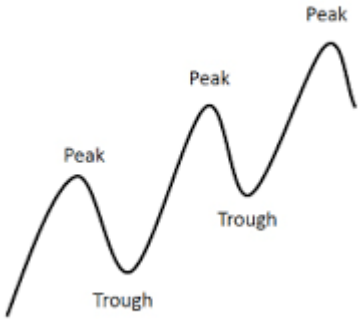
- isolate higher-frequency variability, rather than capturing longer-term trends

Short-term Risk across Time (SRT): CVaR on Differences

- measure the most extreme short-term drop over time

Long-term Risk across Time (LRT): CVaR on Drawdown

- able to capture whether an algorithm has the potential to lose a lot of performance relative to its peak, even if on a longer timescale, e.g. over an accumulation of small drops.
- [A maximum **drawdown** (MDD) is the maximum loss from a peak to a trough of a portfolio]



METRIC DEFINITIONS

- Dispersion across Runs (DR): IQR across Runs
- Risk across Runs (RR): CVaR across Runs
- Dispersion across Fixed-Policy Rollouts (DF): IQR across Rollouts
- Risk across Fixed-Policy Rollouts (RF): CVaR across Rollouts

Basic Criteria

- A minimal number of configuration parameters – to facilitate standardization
- Robust statistics, when possible. Robust statistics are less sensitive to outliers and have more reliable performance for a wider range of distributions.
- Invariance to sampling frequency – results should not be biased by the frequency at which an algorithm was evaluated during training
- Enable meaningful statistical comparisons on the metrics, while making minimal assumptions about the distribution of the results.

Code Ref

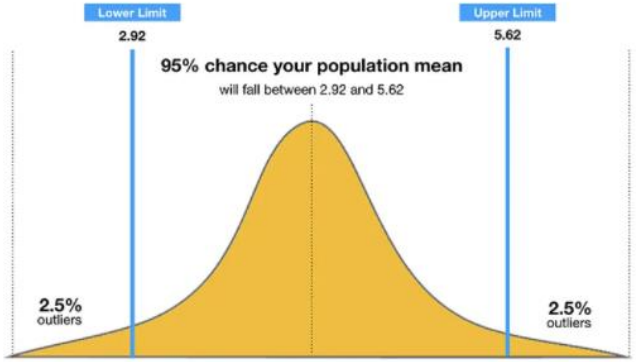
https://github.com/google-research/rl-reliability-metrics

[The RL Reliability Metrics library provides a set of metrics for measuring the reliability of reinforcement learning (RL) algorithms. The library also provides statistical tools for computing confidence intervals and for comparing algorithms on these metrics.

As input, this library accepts a set of RL training curves, or a set of rollouts of an already trained RL algorithm. The library computes reliability metrics across different dimensions (additionally, it can also analyze non-reliability metrics like median performance), and outputs plots presenting the reliability metrics for each algorithm, aggregated across tasks or on a per-task basis. The library also provides statistical tests for comparing algorithms based on these metrics, and provides bootstrapped confidence intervals of the metric values.]

CONFIDENCE INTERVALS AND STATISTICAL SIGNIFICANCE TESTS FOR COMPARISON

- A confidence interval **displays the probability that a parameter will fall between a pair of values around the mean**.
- Compare algorithms evaluated on a fixed set of environments.
- To determine whether any two algorithms have statistically significant differences in their metric rankings, we perform an exact permutation test on each pair of algorithms.
- Such tests allow us to compute a p-value for the null hypothesis (probability that the methods are in fact indistinguishable on the reliability metric).
- that runs are exchangeable across the two algorithms being compared.



P-value	Decision
Less than 0.05*	**Reject Null** (H_0) Hypothesis Statistical difference between groups
Greater than 0.05*	**Fail to Reject** Null (H_0) Hypothesis No statistical difference between groups, or not enough evidence (data) to find a difference

* Assuming $\alpha = 0.05$