Brian Karlberg

Machine Learning 643

Dr. Xubo Song

Spring 2021

Project report

Abstract

Reinforcement learning (RL) is an effective machine learning technique to achieve tasks across a variety of environments. RL works by making decisions of what actions to take in an environment based on generating experience of actions likely to maximize reward based on the state of the environment. Here, an RL algorithm is developed for learning to tune a predictive classifier for a well-behaved dataset. This algorithm is then tested on a cancer subtype prediction task using genomic data.

Background

The decision process of an RL algorithm requires the sensing the state of the environment and predicting the future value of alternative actions.1 One important aspect of predicting this future value is the problem of long-term credit assignment.2 This problem relates to assigning outcomes to actions that may have occurred many time-steps prior to the outcome. A common implementation of an RL algorithm tracks each iteration of interacting with its environment in a Q-table. The values in the Q-table result from evaluating a value function corresponding to the predicted state-qualities resulting from taking different actions. More sophisticated versions of RL algorithms solve systems of Bellman equations in evaluating a value function.3

Data

The datasets used for this project were the iris flower dataset and a genomic cancer dataset of adrenocortical carcinoma. The iris dataset is open-source with 150 samples and three evenly distributed classes and used widely in the field of statistics for benchmarking different analysis methods. The adrenocortical cancer dataset is controlled data from the Genomic Data Analysis Network Tumor Molecular Pathology (GDAN TMP) working group project. This cancer data is methylation features and has unevenly distributed classes across 76 samples.

Method

An exploratory grid search of hyperparameter combinations of an adaboost classifier from SciKit Learn was conducted on the iris data. An 8x8 grid with a total runtime of 72 seconds was selected via trial and error to characterize the learning environment. The environment was finally instantiated as a list of 95 values for number of estimators and 31 values for learning rates yielding 2945 possible states.

An RL algorithm was written to learn the optimal tuning of hyperparameters for the Adaboost classifier. The state of the RL algorithm was initialized as a random combination of learning rate and number of estimators of the Adaboost classifier. Importantly, the quality of the predictive states associated with these combinations was tuned with cross-fold validation. 25 train-test splits per classification was selected via trial-and-error to balance minimization of wall time with minimization of the variance of accuracy. The predictive accuracy of each hyperparameter combination was the primary component of the reward signal sent to the RL algorithm. Additionally, this reward signal incorporated information about the variance of the predictive accuracy.

The RL algorithm tested the predictive accuracy of possible future states by individually changing each hyperparameter value in each direction by a number of steps. These predictions were recorded as possible future state-quality values in a Q-table. The Q-table values were the weighted sum of the prediction accuracy and the variance of each potential change to the hypermeters. The RL algorithm selected the maximum value to decide which hyperparameter change to implement as an action then repeated the state-prediction cycle from this new state.

To address the fundamental trade-off between exploration and exploitation, the RL algorithm would test the predictive accuracy of a random state every five state-changes and compare this value with the current value. If the tested value was greater than the current value, then the state would jump to these randomly selected hyperparameter values with higher predictive accuracy and continue local optimization from there.

Results

The RL algorithm demonstrates minimum viable functionality for hyperparameter tuning by improving classification scores of the environment classifier. The RL algorithm makes decisions based on iterative updates to a state table and a Q-table. The algorithm was run for 12 test cycles on the iris flower data. In each test cycle, the ending hyperparameter combination of the environment classifier resulted in a higher predictive accuracy than the starting combination. Each of these test cycles was a run of 40 state-changes with random states explored and acted upon to maximize predictive accuracy.

To asses generalizability, the RL algorithm was tested on the adrenocortical carcinoma data for six test cycles. The ending predictions of each of these test-cycles were more accurate than the initial predictions.

Discussion

Additional functionality could be added by giving the reinforcement learning algorithm access to select from multiple feature sets or additional types of classifiers. Further work could integrate the not-chosen random jump predictions into the Q-table as a potential source of information for the RL algorithm to tune its own rate of exploration. Additionally, the RL algorithm could be given a mechanism to tune the step-sizes of changes to the classifier hyperparameters when predicting the value of different potential states.

References

1. Silver, David, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. “Reward Is Enough.” *Artificial Intelligence* 299 (October): 103535.
2. Badia, Adrià Puigdomènech, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. 2020. “Agent57: Outperforming the Atari Human Benchmark.” In *International Conference on Machine Learning*, 507–17. PMLR.
3. Ng, Andrew. 2008. “Lecture 16, topic of reinforcement learning”. *Stanford CS 229, Machine Learning.* https://youtu.be/RtxI449ZjSc