Muninn

Simon Ståhlberg[1], **Blai Bonet**[2], **Hector Geffner**[3,1]

[1]Linköping University, Linköping, Sweden
[2]Universitat Pompeu Fabra, Spain
[3]RWTH Aachen University, Germany
simon.stahlberg@liu.se, bonetblai@gmail.com, hector.geffner@ml.rwth-aachen.de

Abstract

Our planner, Muninn, utilizes a message-passing neural network model to learn and optimize value functions for a given domain. Muninn employs a two-step learning process to first obtain an optimal value function and then a suboptimal value function. Initially, the model learns an optimal value function that represents the fewest actions needed to reach a goal state. Subsequently, the value function is fine-tuned to generate suboptimal solutions when used with a greedy policy. The final learned value function is utilized in two ways. Firstly, it is employed in a hill-climbing search algorithm. Secondly, it serves as a heuristic function within an A* search algorithm.

Overview

Muninn is a planner that utilizes a message-passing neural network architecture to learn a value function for a given planning domain. In our previous work (Ståhlberg, Bonet, and Geffner 2022a,b), we learned general policies by training value functions and employing a greedy policy based on them. However, although not formally proven, it is likely that our model's expressive power is limited by C_2 (two-variable first-order logic with counting quantifiers). If the planning domain necessitates features that cannot be captured by C_2, the greedy policy based on the learned value function will fail to generalize to larger instances. Moreover, there is a possibility that the learned value function cannot even solve the training instances. Consequently, the inclusion of search algorithms becomes crucial to compensate for this limitation in expressive power.

To learn a value function, Muninn follows a two-step learning process, dedicating an equal amount of time to each step. Initially, the model learns an optimal value function that represents the minimum number of actions required to reach a goal state. We learn this value function in a supervised fashion by expanding the entire reachable state space and computing the value for each state. However, the lack of expressive power is not the only scenario where policies based on learned value functions fail. In cases where finding optimal solutions is NP-hard but finding suboptimal solutions is feasible, it is preferable to learn a value function that yields suboptimal plans when used with a greedy policy. Thus, in the second step, Muninn fine-tunes the optimal value function to generate suboptimal solutions when used as a greedy policy. While it is possible to directly learn suboptimal policies, it generally takes longer to converge. Hence, the first step aims to speed up learning process.

The learned value function in Muninn is used in two ways, dedicating an equal amount of time to each way. Firstly, it is utilized in a hill-climbing search algorithm, where A* is employed from the current state to identify another state where the value is significantly better than the current value. At this point, the search is reset to avoid maintaining a large open list. This approach tends to find solutions quickly but often yields plans of lower quality. Secondly, the learned value function acts as a heuristic function within an A* search algorithm, which typically discovers solutions of high quality but at the cost of increased time and memory consumption. We note that we use our value function as a heuristic, but it is not admissible. In other words, even when employed with A* algorithm, it does not guarantee optimal solutions.

For detailed information on learning an optimal value function through supervised learning, we kindly direct the reader to our ICAPS paper (Ståhlberg, Bonet, and Geffner 2022a), where our architecture was originally introduced. Additionally, we invite the reader to explore our KR paper (Ståhlberg, Bonet, and Geffner 2022b) for insights into how we learn suboptimal policies using value functions. This paper also delves into the trade-off between optimality and generality for NP-hard domains. In these papers, we refrain from incorporating the learned value functions into search algorithms. Our aim was to obtain crisp results, avoiding any potential masking of flaws in the learned models by the search algorithm. However, in a competitive environment, the goal is to maximize coverage; thus, using search algorithms is very useful.

References

Ståhlberg, S.; Bonet, B.; and Geffner, H. 2022a. Learning General Optimal Policies with Graph Neural Networks: Expressive Power, Transparency, and Limits. In *Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling, ICAPS 2022, Singapore (virtual), June 13-24, 2022*, 629–637.

Ståhlberg, S.; Bonet, B.; and Geffner, H. 2022b. Learning Generalized Policies without Supervision Using GNNs. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel. July 31 - August 5, 2022*.