Learning and Modeling Real-Time A*

Zeyi Wang
Department of Computing Science
University of Alberta
Edmonton, Alberta, T6G 2E8, Canada
zeyi2@ualberta.ca

Abstract

Learning and Modeling Real-Time A* builds a model for the map as the algorithm explores the search space and improve its heuristic based on the built model.

1 Introduction

Real-time search problems are search problem that focus on the utilization of local information. Learning algorithms that aim to solve this type of problem usually address learning by updating heuristic values of visited states (e.g., LRTA*, LSS-LRTA*, RTAA*, daLSS-LRTA*, ~~e.t.c~~ etc.) However, there's usually some hidden features of the target map, and the algorithms above does not take that into account. Here I present Learning and Modeling Real-Time A* (LMRTA*), a learning algorithm that models the map as the algorithm explores the search space.

2 Problem Formulation

Searching in real-time can be interpreted in two ways. One way is to interweave planning and taking actions. As information of states is only available in their adjacent states, an agent has to explore to some degree to retrieve that information. The other way is to limit the amount of computation an agent could use between taking moves. It has been proven essential for an agent in such setting to have at least some form of memory to reach the goal state. The question lies in what kind of memory should the agent have? The minimum would be an open set and a closed set. In addition to that, LRTA* proposes heuristic updates that overwrites the original value of the heuristic. (other algorithms, blah, blah) However, none of these algorithms utilize the fact that, though maps can be drastically different from each other, most maps have some kind of intrinsic feature to the maps. In other words, most maps are somewhat isometric within themselves. This could be local features like corridors, rooms and basins. There also could be features in a global sense, like the map could be somewhat smooth, sheer, or having a lot of corners. By building a model of the map, the heuristic then has access to this information and can (hopefully) make educated guess on the real heuristic of undiscovered regions.

Since building a map model is time consuming, I would like to focus on reducing the number of explored states when comparing with other algorithm. This means this algorithm should be applied where the cost of exploring a state is expensive by doing computations between moves is more liberal. It is still possible for the algorithm to be efficient in time-bound applications, especially when the map's large, since the operation of updating the model and querying the model could be constants.

3 Related Work

I couldn't find any prior work both mentions models and real-time search so I assume there's none.

4 Proposed Approach

Right now I have a very vague idea of doing this. First of all, we start with a standard LRTA* procedure. For each updates to the heuristic, we record the features of the state (x/y coordinates, f, g) as the model's input (x) and the actual h after the update as the model output (y). As the agent explores the map, we repetitively train the model so it makes better prediction on the h value regarding to the basic features. As the agent explore more, we put more weight on the model's prediction as a part of the heuristic and hopefully the knowledge of the map and gives the algorithm an edge when exploring large map with intrinsic features. (I have no idea if this works, let's find out.)

5 Theoretical Analysis

N/A

6 Empirical Evaluation

Not available yet.

What do you expect to do here?

7 Discussion

Not available yet.

8 Future Work

Not available yet.

9 Conclusions

Not available yet.

Acknowledgments

Not available yet.

Bibliography

?

I approve the proposal.

An interesting idea. So you are really estimating heuristic for unseen areas of the search graph instead of using LRTA*'s formula + table representation of h.

Have a look at RL w/ function approximation.

35/100