Utilising Uncertainty for Efficient Learning of Likely-Admissible Heuristics

Or Marom, Benjamin Rosman
University of the Witwatersrand

Previous Work
Handcrafting heuristics is prohibitive. Alternative: learn heuristics from data with Machine Learning.

Bootstrap Learning Heuristics
1: initialise weak heuristic $h$
2: repeat
3: generate training tasks randomly
4: try solve each task in time-limit
5: if can’t solve enough then
6: increase time-limit
7: else
8: update $h$ with plans of solved tasks
9: end if
10: until forever

Shortcomings:
- inefficient to train (random task generation)
- high suboptimality (train on non-optimal plans)

Our Contribution
Epistemic Uncertainty: lack of training data
Aleatoric Uncertainty: variation in training data

Used to:
- systematically explore task-space (more efficient than random tasks)
- plan with likely-admissible heuristic (train on likely-optimal plans)

Efficient Task Generation
- start at goal state
- seek state with high epistemic uncertainty
- generate task with this start state
- solve and learn from plan

Reduce Suboptimality
- plan with conservative $y_{low}$
- i.e. $P(y_{low} \leq y) = 0.95$ (likely-admissible heuristic)
- produces likely-optimal plans
- mitigates compounding errors during training

Our Algorithm
Efficiently Learn Likely-Admissible Heuristic
1: initialise weak heuristic $h$
2: repeat
3: generate training tasks efficiency
4: try solve each task with $y_{low}$
5: if can’t solve enough then
6: make $y_{low}$ less conservative
7: else
8: update $h$ with plans of solved tasks
9: end if
10: until forever

How to model these uncertainties? We use Weight Uncertainty Neural Networks.

Results
Efficiency results for 15-puzzle
Suboptimality results for 24-puzzle