Learning Inadmissible Heuristics During Search

Jordan Thayer, Austin Dionne, and Wheeler Ruml

University of New Hampshire

jtd7, ajv2, ruml at cs.unh.edu

With thanks to NSF Grant IIS-0812141
and DARPA CSSG Grant N10AP20029
A*
greedy best-first search
greedy best-first search with learning
● motivation

● building inadmissible heuristics during search
  observing error
  correcting for error

● performance of learned heuristics
  suboptimal search - greedy best-first search
  bounded suboptimal search - skeptical search
Motivation For Our Approach

- goal: work out of the box on single instances
- avoid offline training
- avoid domain specific features
- rely on data easily available in any best-first search

boost any suboptimal search
Learning
Observing Error Between Parent and Best Child

p

\[ c(p, bc) \]

bc

\[ \star \]
Observing Error Between Parent and Best Child

\[ f(p) \text{ should equal } f(bc) \]
Observing Error Between Parent and Best Child

\[ f(p) \text{ should equal } f(bc) \]

\[
\begin{align*}
    f^*(p) &= f^*(bc) \\
    g(p) + h^*(p) &= g(bc) + h^*(bc) \\
    h^*(p) &= h^*(bc) + c(p, bc)
\end{align*}
\]
$f(p)$ should equal $f(bc)$

\[ f^*(p) = f^*(bc) \]

\[ g(p) + h^*(p) = g(bc) + h^*(bc) \]

\[ h^*(p) = h^*(bc) + c(p, bc) \]

\[ h(p) = h(bc) + c(p, bc) - \epsilon_h \]

\[ \epsilon_h = h(bc) + c(p, bc) - h(p) \]
Observing Error Between Parent and Best Child

\[ f(p) \text{ should equal } f(bc) \]

\[ f^*(p) = f^*(bc) \]
\[ g(p) + h^*(p) = g(bc) + h^*(bc) \]
\[ h^*(p) = h^*(bc) + c(p, bc) \]

\[ h(p) = h(bc) + c(p, bc) - \epsilon_h \]
\[ \epsilon_h = h(bc) + c(p, bc) - h(p) \]

\[ \hat{h}(n) = h(n) + \bar{\epsilon}_h \cdot d(n) \]
\[ \hat{h}(n) = h(n) + \bar{\epsilon}_h \cdot \hat{d}(n) \]
$\hat{h}(n) = h(n) + \bar{\epsilon}_h \cdot \hat{d}(n)$

how do we estimate $\bar{\epsilon}_h$ from $\epsilon_h$? simple global average
\[ \hat{h}(n) = h(n) + \bar{\epsilon}_h \cdot \tilde{d}(n) \]

how do we estimate \( \bar{\epsilon}_h \) from \( \epsilon_h \)?

simple global average or ...
\[ \hat{h}(n) = h(n) + \bar{\epsilon}_h \cdot \hat{d}(n) \]

how do we estimate \( \bar{\epsilon}_h \) from \( \epsilon_h \)?

simple global average or ...

\[ \bar{\epsilon}_h = 0 \]

\[ \epsilon_h = 0.5 \]

\[ \bar{\epsilon}_h = 0 \]

\[ \epsilon_h = 0 \]

\[ \epsilon_h = 0 \]

\[ \epsilon_h = 0 \]

\[ \epsilon_h = 0 \]
\[ \hat{h}(n) = h(n) + \bar{\epsilon}_h \cdot \hat{d}(n) \]

how do we estimate \( \bar{\epsilon}_h \) from \( \epsilon_h \)?

simple global average or ...

\[ \bar{\epsilon}_h = 0 \quad \bar{\epsilon}_h = 1 \]

\[ \bar{\epsilon}_h = 0.5 \quad \bar{\epsilon}_h = 1 \]
Summary

- a parent and its best child should have same $f$
- every expansion provides information – use it!
- single step error can be measured during search and we can use those corrections during that search
Performance
motivation

building inadmissible heuristics during search

performance of learned heuristics
  suboptimal – greedy best-first search
  bounded suboptimal – skeptical search
Life Four-way Grids 35% Obstacles

- Manhattan + learning
- Manhattan
Greedy Best First Search

Korf's 100 15 Puzzles - Inverse Cost

Solution Quality vs. total raw cpu time

- base + learning
- base
- motivation
- building inadmissible heuristics during search
- performance of learned heuristics
  - suboptimal – greedy best-first search
  - bounded suboptimal – skeptical search
given a suboptimality bound $w$, find a solution within the bound as quickly as possible
given a suboptimality bound $w$,  
find a solution within the bound as quickly as possible. 

use optimistic framework (Thayer and Ruml, ICAPS-08):

1. run weighted $A^*$ with an **inadmissible heuristic**

   $f'(n) = g(n) + w \cdot \hat{h}(n)$

2. after a solution is found expand node with lowest $f$ value 
   continue until $w \cdot f(best_f) \geq f(sol)$ 
   this ’clean up’ guarantees solution quality

(no ad hoc optimism parameter!)
Performance In Bounded Suboptimal Search

Introduction

Learning

Performance

- Greedy Search
- Bounded Quality

Conclusions

Backup Slides

100 Inverse 15 Puzzles

Optimistic

wA*

Skeptical

log10 total raw cpu time

Suboptimality

Learning During Search – 14 / 20
Dock Robot

Suboptimality vs. log10 total raw cpu time

- wA*
- Optimistic
- Skeptical

Performance In Bounded Suboptimal Search
Performance In Bounded Suboptimal Search

Introduction
Learning
Performance
- Greedy Search
- Bounded Quality
Conclusions
Backup Slides

Heavy Vacuum World

log10 total raw cpu time vs. Suboptimality

- wA* (Optimistic)
- Skeptical

Jordan Thayer (UNH)
- accuracy less important than relative ordering
- instance specific learning truly beneficial
- distance estimates very helpful for non-unit cost problems
- skeptical proof of bounded suboptimality
we can learn inadmissible heuristics
these improve search guidance, make search go fast

we can learn them online, during search
no dependence on domain specific information
no offline training
can learn instance specific correction

skeptical search
removes parameter of optimistic search
state of the art performance
Tell your students to apply to grad school in CS at UNH!

- friendly faculty
- funding
- individual attention
- beautiful campus
- low cost of living
- easy access to Boston, White Mountains
- strong in AI, infoviz, networking, systems, bioinformatics
Heuristic Accuracy

Eight Puzzle

Error

Base | Off-line ANN | Off-line LMS | On-line ANN | On-line LMS | Single Step Path | Single Step Global

Jordan Thayer (UNH)
It Doesn’t Always Work

- Accuracy
- Counter Example
- Use $d$
Life Four-way Grids 35% Obstacles

- Solution Cost vs. total raw cpu time

- Just h Path
- Just h Global
- Single Step Path

- 3e+06
- 2.9e+06
- 2.8e+06