Introduction

Problem
- Improve the efficiency of finding shortest paths between arbitrary points in a search graph
- Example applications: road networks, virtual worlds

Approach
- Assign each state \( i \) in the input graph a point \( y_i \in \mathbb{R}^d \)
- Use interpoint distances as heuristics for distances in the graph
- Arrange the points in such a way as to minimize the error between the estimated distances and the true distances

Contributions
- Link between heuristic construction and manifold learning
- Promising empirical results on a range of test domains

Euclidean Heuristics

A Euclidean heuristic is a heuristic function \( h \) for any state pair that can be computed from distances between points:

\[
h(i, j) = \| y_i - y_j \|
\]

The arrangement of the points \( Y \) defines \( h \).

An optimal Euclidean heuristic minimizes the loss \( L \) between the true distances \( \delta(i, j) \) and the heuristics given by the points \( Y \):

\[
\minimize \ L(Y) \quad \text{subject to} \quad Y \text{ is admissible and consistent}
\]

The Loss Function

Favor larger errors by squaring terms, but admit a weight \( W_{ij} \) on the relative importance of each pair \((i, j)\):

\[
L(Y) = \sum_{ij} W_{ij} \| \delta(i, j) - \| y_i - y_j \| \|^2
\]

Admissibility \( \delta(i, j)^2 \geq \| y_i - y_j \|^2 \) permits a simpler loss:

\[
L(Y) = \sum_{ij} W_{ij} \| \delta(i, j) - \| y_i - y_j \| \|^2 \equiv - \sum_{ij} W_{ij} \| y_i - y_j \|^2
\]

The resulting optimization problem:

\[
\maximize Y \sum_{ij} W_{ij} \| y_i - y_j \|^2 \\
\text{subject to} \quad \forall (i, j) \in E \| y_i - y_j \| \leq \delta(i, j)
\]

Connections

Manifold Learning
This is a weighted generalization of MVU (Weinberger et al.)
- Nonlinear dimensionality reduction
- MVU’s semidefinite reformulation renders the optimization tractable

Differential Heuristics

\( W \) can be parametrized to reproduce differential heuristics:

\[
W_{\text{diff}} = \begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & \epsilon & \epsilon \\
1 & \epsilon & 0 & \epsilon \\
1 & \epsilon & \epsilon & 0
\end{bmatrix} (\epsilon = 10^{-3})
\]

Suppose pivot is state 1:
- Push points away from pivot
- Pull points into each other

Constraints

\( Y \) is admissible between adjacent states

\( Y \) is admissible and consistent

Experiments

Comparison against differential heuristics using \( A^* \):

Cube World
- 20 x 20 x 20 = 8,000 states
- Agent increments any/all coordinates by 1; transition costs are the edge lengths
- Multi-dimensional search spaces target a weakness of differential heuristic

Video Game Maps
- 168-6,240 states
- Octile connectivity: diagonals cost 1.5
- Maps are like corridors to which differential heuristics are well suited; a competing \( W_{\text{diff}} \) is introduced:

\[
W_{\text{diff}} = \begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & \epsilon & \epsilon \\
1 & \epsilon & 0 & \epsilon \\
1 & \epsilon & \epsilon & 0
\end{bmatrix} (\epsilon = 10^{-3})
\]

Four-Letter Word Graph
- Connected graph of 4,820 words
- Agent changes 1 letter per step
- High dimensional domain

Cube World
Game Maps
Word Graph

- Bar plots: variance in each dimension of the uniformly weighted embedding
- Optimal Euclidean heuristics show promise, storing more in less memory

Thank you
Ariel Felner
Anonymous reviewers
NSERC and iCore

1. Department of Computing Science University of Alberta
2. Computer Science Department University of Denver

Differential heuristics are well suited; a weakness of differential heuristic

\[
\text{Connections}
\]

\[
\text{Constraints}
\]

\[
\text{Experiments}
\]

\[
\text{Introduction}
\]

\[
\text{Euclidean Heuristics}
\]

\[
\text{The Loss Function}
\]

\[
\text{Connections}
\]

\[
\text{Constraints}
\]

\[
\text{Experiments}
\]

\[
\text{Thank you}
\]

Chris Rayner\(^1\) Michael Bowling\(^3\) and Nathan Sturtevant\(^2\)