Utilizing Landmarks in Euclidean Heuristics for Optimal Planning

Qiang Lu*, **Wenlin Chen**†, **Yixin Chen**†, **Kilian Q. Weinberger**†, **Xiaoping Chen***
*School of Computer Science and Technology, University of Science and Technology of China
†Department of Computer Science and Engineering, Washington University in St. Louis
{qianglu8, xpchen}@ustc.edu.cn, {wenlinchen, chen, kilian}@wustl.edu

Abstract

An important problem in AI is to construct high-quality heuristics for optimal search. Recently, the Euclidean heuristic (EH) has been proposed, which embeds a state space graph into a Euclidean space and uses Euclidean distances as approximations for the graph distances. The embedding process leverages recent research results from manifold learning, a subfield in machine learning, and guarantees that the heuristic is provably admissible and consistent. EH has shown good performance and memory efficiency in comparison to other existing heuristics. Our recent works have further improved the scalability and quality of EH. In this short paper, we present our latest progress on applying EH to problems in planning formalisms, which provide richer semantics than the simple state-space graph model. In particular, we improve EH by exploiting the landmark structure derived from the SAS+ planning formalism.

Euclidean Heuristics with Manifold Learning

Fast heuristic search is needed in many systems such as robot planning software, GPS navigation systems, and video games. Such systems need to find the shortest path between two states *efficiently* and *repeatedly* (Sturtevant 2007; Geisberger et al. 2008). As the processor capabilities of these devices and the patience of the users are both limited, the quality of the search heuristic is of great importance. This importance only increases as more and more *low powered* devices such as smart-phones are used.

Many systems are very time-sensitive, as the search response time greatly affects the end users' satisfaction of the products. Further, embedded devices have limited memory, demanding a compact representation of the heuristic. Given n states, a perfect heuristic can be found by storing all true distances between any two states—however the $O(n^2)$ memory requirement renders this approach impractical. We conclude the following key requirements for the heuristic functions in such systems: 1) For a heuristic search to be optimal, the heuristic must be admissible, 2) For a search to be fast, heuristics need to be accurate and consistent, and 3) For a heuristic to fit in memory (of an embedded device), it must require low memory footprint.

Rayner, Bowling, and Sturtevant (2011) propose the *Euclidean heuristic (EH)*. EH embeds each state s into a point in a d-dimensional space, $x_s \in \mathcal{R}^d$. The EH heuristic value between any two states s and s' is the Euclidean distance $\|x_s - x_{s'}\|_2$. The heuristic is provably admissible, accurate and consistent, and the memory requirement is only $O(dn)$, where d can be small (usually $d = 3$ or $d = 4$). The embedding is found by minimizing the gaps between the heuristic estimates and the true graph distances, while strictly maintaining local distance constraints. The resulting optimization problem is identical to Maximum Variance Unfolding (MVU), a manifold learning algorithm that finds low-dimensional embeddings of nonlinear manifolds in high-dimensional spaces (Weinberger and Saul 2006).

However, MVU requires expensive optimization and can only handle up to a few thousand states, even under a relaxed semi-definite programming (SDP) formulation. We recently developed a variation, Maximum Variance Correction (MVC), that scales up MVU by several orders of magnitude using optimization decomposition (Chen, Weinberger, and Chen 2013). MVC can embed graphs with $200K$ states or more and is naturally parallelizable. MVC achieves its drastic speedups by post-processing embeddings from faster manifold learning algorithms (Saul et al. 2006; Weinberger, Packer, and Saul 2005) to become feasible MVU solutions and guarantee admissibility and consistency.

With the development of MVC, EH becomes an attractive and scalable choice for computing heuristics. In a paper accepted at AAAI'13, we have proposed two techniques to improve EH (Chen et al. 2013). The first technique is called goal-oriented EH (GOEH). In GOEH, we observe that the objective of MVC minimizes the sum of the distance gaps between all pairs of states, while A^* search is guided only by the heuristic distance to the goal state. In many problems, we can identify a small set of possible or likely goal states in the state space. GOEH changes the optimization objective so that it only minimizes the distance gaps towards possible goal states. The second technique is called state heuristic enhancement (SHE). For each state, SHE stores the minimum gap between the Euclidean and true distances to all the goal states. Such values are used to enhance the heuristic values during search. Our results show that both GOEH and SHE lead to improvements in the heuristic quality over the previous EH and differential heuristics, resulting in faster search.

Exploiting Landmarks in EH for Planning

Automated planning is a core area in AI. It provides expressive formalisms to describe search problems and enables automatic generation of the state-space graph. An important question is whether we can extend EH from plain state-space graph formulations to automated planning and exploit the rich structural information encoded in planning formalisms.

In our study, we use the popular SAS+ planning formalism (Bäckström and Nebel 1996) and exploit so called *landmarks* (Porteous, Sebastia, and Hoffmann 2001) that can be automatically derived with algorithms such as the LAMA planner (Richter and Westphal 2010).

There are two phases for using EH: 1) an *offline* phase, that learns a low-dimensional embedding for all the states, and 2) an *online* phase, that computes heuristics based on the embedding. We propose to exploit landmarks in both phases.

Offline phase. We derive a different embedding than GOEH in the offline phase. Given a planning domain, we denote its state-space graph to be $G = (V, E)$ where V are states and $(v, w) \in E$ iff. $w \in V$ is a successor state of $v \in V$. Let $V_G \subseteq V$ be the set of possible goal states for online search. Note that MVC only works on undirected graphs so we are currently limited to problems with undirected state-space graphs.

We define a set of *landmark* goal states, $V_L = \{g \in V_G \mid \exists(w, g) \in E \wedge w \notin V_G\}$, such that $V_G - V_L$ includes those goal states that are "surrounded" by other goal states. It is easy to see that any solution path must first visit a landmark state in V_L before reaching any other goal state.

In the offline embedding, GOEH uses MVC to minimize the total Euclidean distances between states (v, g) with $v \in V, g \in V_G$. By analyzing the landmark goals, we modify the objective to only include pairs of states (v, ℓ) such that $v \in V$ and $\ell \in V_L$. Intuitively, since any optimal path must reach a goal state in V_L, an optimal search only needs to be guided by heuristic distances to the states in V_L.

The technique is particularly useful for planning since typically a planning instance only specifies some goal *facts*, which may lead to many possible goal *states*, making GOEH less effective. By reducing V_G to V_L ($|V_L| \ll |V_G|$), we exclude many spurious goal states. Hence, the embedding is further "stretched" towards those few landmark goal states in V_L, improving the quality of GOEH.

We also use landmarks to enhance the SHE technique. Let $d_{s,g}$ denote the true distance from $s \in V$ to $g \in V$. In the original SHE, we store a value $\eta(s) = \min_{g \in V_G}\{d_{s,g} - \|x_s - x_g\|_2\}$, for each state $s \in V$. $\eta(s)$ is the minimal distance gap to any goal state. During the online search, we approximate $d_{s,g}$ with $\|x_s - x_g\|_2 + \max[\eta(s), \eta(g)]$. The resulting heuristic is still admissible. Now we modify the SHE value to $\eta(s) = \min_{\ell \in V_L}\{d_{s,\ell} - \|x_s - x_g\|_2\}$. Namely, we replace the goal set V_G by the landmark goal state set V_L. It gives better enhancement because it takes the minimum from a smaller set of distance gaps.

Online phase. During the online search phase, we are given an initial state and goal facts each time. The SAS+ formalism provides expressive semantics from which we can derive key structural information. For example, once we know the initial state, we can use algorithms such as LAMA to find landmark facts, which can help further improve GOEH.

A landmark fact f is a fact that must be made true before reaching the goal. For each landmark fact f, we find $V_f \in V$, the set of all these states in which f is true. During the search, before f is made true, the heuristic distance from a state s to the goal state g can be improved to

$$h_f(s, g) = \min_{v \in V_f}\{\|x_s - x_v\|_2 + \|x_v - x_g\|_2\}.$$

This new heuristic, $h_f(s, g)$ is often better and never worse than the original Euclidean heuristic. Due to the triangle inequality in the Euclidean space, we know that $\|x_s - x_g\|_2 \leq h_f(s, g)$. Further, $h_f(s, g)$ is still admissible due to the fact that f is a landmark fact and any solution path must visit a state in V_f.

When there are multiple conjunctive landmark facts $\{f_1, \cdots, f_C\}$, which means any solution path must make all of $\{f_1, \cdots, f_C\}$ true, we take the maximum of $h_{f_i}(s, g)$, $i = 1, \cdots, C$.

Evaluation and Outlook

We are developing a planning-based solution for the Home Service Robot Competition problems at the RoboCup. We formulate this problem in the PDDL planning model. We implement our solver in the Fast Downward planner (Helmert 2006) which translates PDDL into SAS+ and finds landmarks. Each domain has a service robot, humans, small objects, and obstacles. The robot is asked to perform various daily tasks such as picking up a small object and bring it to the human. In our approach, we first learn a d-dimensional embedding of all the states in the offline phase and store the embedding in the robot's memory, which will be used by the robot to compute EH heuristics in the online phase.

Our preliminary results show that the set of landmark goal states typically contains only about 10% of all goal states. We also find that the proposed offline and online techniques lead to reductions in the number of expanded states. EH requires up to several hours of MVC optimization in the offline phase. However, EH is intended for the applications where the offline preprocessing time is of no particular importance, but the online search time is crucial.

In summary, we reported recent work on exploiting the landmark structure in planning problems to enhance EH. Our approach is intended for repeatedly solving different instances under a given planning domain. It is most useful for embedded systems such as robots and GPS where the online search speed is crucial. Limited by the memory in these systems, our current approach cannot handle planning domains with a very large state space. To handle such large problems, a future work of ours is to learn manifolds in certain subsets (e.g. involving a few, but not all, domains transition graphs in the SAS+ formalism) of the state space. We will also study the interaction between EH and other search-space reduction techniques (Chen and Yao 2009; Chen, Xu, and Yao 2009). We believe much more progress will be made along this exciting direction at the confluence of machine learning and planning.

Acknowledgements

WC and YC are supported in part by NSF grants CNS-1017701 and CCF-1215302. KQW is supported by NIH grant U01 1U01NS073457-01 and NSF grants 1149882 and 1137211. QL and XC are supported by Natural Science Foundation of China grant 61175057 and China Postdoctoral Science Foundation grant 2013M531527.

References

Bäckström, C., and Nebel, B. 1996. Complexity results for SAS+ planning. *Computational Intelligence* 11:625–655.

Chen, Y., and Yao, G. 2009. Completeness and optimality preserving reduction for planning. In *Proc. IJCAI*.

Chen, W.; Chen, Y.; Weinberger, K.; Lu, Q.; and Chen, X. 2013. Goal-oriented euclidean heuristics with manifold learning. In *Proc. AAAI*.

Chen, W.; Weinberger, K.; and Chen, Y. 2013. Maximum variance correction with application to A^* search. In *Proc. ICML*.

Chen, Y.; Xu, Y.; and Yao, G. 2009. Stratified planning. In *Proc. IJCAI*.

Geisberger, R.; Sanders, P.; Schultes, D.; and Delling, D. 2008. Contraction hierarchies: faster and simpler hierarchical routing in road networks. In *Proc. ICEA*, 319–333.

Helmert, M. 2006. The Fast Downward planning system. *Journal on Artificial Intelligence Research* 26:191–246.

Porteous, J.; Sebastia, L.; and Hoffmann, J. 2001. On the extraction, ordering, and usage of landmarks in planning. In *Proc. ECP*.

Rayner, C.; Bowling, M.; and Sturtevant, N. 2011. Euclidean Heuristic Optimization. In *Proc. AAAI*, 81–86.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research* 39:127–177.

Saul, L.; Weinberger, K.; Ham, J. H.; Sha, F.; and Lee, D. D. 2006. *Spectral methods for dimensionality reduction*. chapter 16, 293–30.

Sturtevant, N. R. 2007. Memory-efficient abstractions for pathfinding. In Schaeffer, J., and Mateas, M., eds., *Proc. AIIDE*, 31–36.

Weinberger, K., and Saul, L. 2006. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision* 70:77–90.

Weinberger, K.; Packer, B.; and Saul, L. 2005. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *Proc. International Workshop on Artificial Intelligence and Statistics*.