第 20 卷 第 2 期
1994 年 3 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 20, No. 2
Mar., 1994

短 文

基于函数逼近的学习式搜索[1)]

张 伟　刘积仁　李华天

(东北大学计算机系　沈阳　110006)

摘　要

本文将函数逼近的方法引入到学习式搜索中，使学习式搜索通过一定数量的解题训练后可以建立起一个任意一致逼近理想函数 $h^*(\cdot)$ 的启发估计函数 $h(\cdot)$. 本文给出了一个这样的学习式搜索算法 A-Bn，并证明了当训练例子集充分大后，A-Bn 可在多项式复杂度内解决任一后来提交的同类问题.

关键词：人工智能，启发式搜索，机器学习，复杂度.

1 前　　言

搜索是人工智能的基本解题策略[1]. 然而，最近的一项研究表明，不存在一个可采纳搜索算法，它能够永远避开“指数爆炸”[2,3]. 笔者认为，导致这种结果的原因在于以往所有的启发式搜索都是一次性的搜索，即它们假设搜索算法每次解题时面临的都是一个陌生的庞大的问题状态空间，对同一问题状态空间的多次搜索被视为彼此独立的过程，前一次搜索对后一次搜索不产生任何影响.

搜索问题被这样刻画以后，使得启发式搜索作为人工智能解题策略的基础显得不尽合理.

一个智能的解题策略，实际上是一种多次性的解题策略，对同一类型的问题，常常是多次反复地求解，而且解题的速度越来越快. 当这种解题策略被描述为搜索时，该搜索实际上是一种多次性的搜索，即搜索算法是有机会获得有关问题状态空间的经验信息的. 因为这种搜索受以前搜索经验的影响，故亦称为“学习式搜索”.

2 学习式搜索的问题描述

在经典的启发式搜索问题模型中，每次解题都是一个独立的过程，启发函数 h 是先验的和不易改变的. 学习式搜索则与此不同，它假设搜索算法曾有机会获得启发信息，因此 h 不再是先验的，而是可由算法根据经验重新构造的后验函数. 定义学习式搜索的问题

1) 本课题由国家自然科学基金资助.
本文于 1992 年 1 月 7 日收到

模型如下:

定义 1. 给定一个问题状态空间 $PS=(\Sigma, \text{STATES}, \text{OPERATORS})$ 和一个训练例子集 $TP_t=\{(PS, \text{start}_i, \text{goal}_i)|\text{start}_i, \text{goal}_i \in \text{STATES}$; 当 $i \neq j$ 有 $(\text{start}_i, \text{goal}_i) \neq (\text{start}_j, \text{goal}_j)$; $i=1,2,\cdots,t$; 三元组 (PS, s, g) 表示 PS 中的一个搜索问题$\}$, 记问题 $(PS, \text{start}_i, \text{goal}_i)$ 的最佳解路径长为 $l^*(\text{start}_i, \text{goal}_i)$, 记一搜索算法 α 经 TP_t 训练后求解新问题 $(PS, \text{start}', \text{goal}')$ 的复杂度为 $E^{\alpha}_{TP_t}(PS, \text{start}', \text{goal}')$, 并以 $\text{poly}(x)$ 表示 x 的某一多项式阶函数,试设计一个搜索算法 Γ, 使之求解任一新给的问题 $(PS, \text{start}^*, \text{goal}^*)$ 时满足

$$\lim_{t \to |\text{STATES}|} E^{\Gamma}_{TP_t}(PS, \text{start}^*, \text{goal}^*) = O[\text{poly}(l^*(\text{start}^*, \text{goal}^*))].$$

目前,学习式搜索的实现方法有几种[4],基于函数逼近的学习式搜索是其中之一. 此种方法主张通过 TP_t 建立 $h(\cdot)$ 函数去一致地逼近 $h^*(\cdot)$ 函数,以精确的 $h(\cdot)$提高搜索的效率.

3 基于函数逼近的学习式搜索

3.1. 函数逼近理论简介

函数逼近问题可简单描述为, 当函数 $f(x)$ 不能直接得到而只是给出了函数的若干个值时,如何寻求某一函数 $\varphi(x)$ 去逼近 $f(x)$.

关于这个问题有两个重要的定理.

Weierstrass 定理. 设 $f(x)$ 是 $[a, b]$ 上的连续函数, 则对于任何 $\varepsilon > 0$, 存在一多项式 $P(x)$ 使不等式

$$|f(x) - P(x)| < \varepsilon, \text{ 对于 } a \leqslant x \leqslant b \text{ 一致成立.}$$

Bernstein 定理. 若记

$$B_n(f, x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) \cdot C_n^k \cdot x^k \cdot (1-x)^{n-k},$$

则有

$$\lim_{n \to \infty} B_n(f, x) = f(x), \quad 0 \leqslant x \leqslant 1.$$

3.2. 基于函数逼近的学习式搜索

假设问题状态空间是一棵 m 元一致树,每个节点与其相邻节点的距离均为 1; 树深有限, 其最大值为 M; 唯一的目标节点在深度 N 处. 采用 $f=g+h$ 型的启发函数来制导搜索(其中 $h(x, y)$ 是对节点 x 到节点 y 的实际距离 $h^*(x, y)$ 的估计),与以往不同的是,现在的 $h(x, y)$ 不再是先验的,而是随着 TP_t 的增长而逼近于 $h^*(x, y)$ 的.

记 y 为 (x, y) 的康托数在[0,1]内的映射. 将 $h(y) \triangleq h(x, y)$ 定义为 Bernstein 多项式

$$h(y) = B_n(h^*, y) = \sum_{k=0}^{n} h^*\left(\frac{k}{n}\right) \cdot C_n^k \cdot y^k \cdot (1-y)^{n-k},$$

其中 $h^*\left(\frac{k}{n}\right)$ 为学习到的康托数为 $\frac{k}{n}$ 的两节点间的实际距离。

当训练例子集 TP_i 为上述函数提供越来越多的函数值 $h^*(\cdot)$ 时，$B_n(h^*, y)$ 的阶数 n 就越来越大，那么由 Bernstein 定理可知，$B_n(h^*, y)$ 将一致地趋近于 $h^*(\cdot)$。当 $|h(\cdot) - h^*(\cdot)| < \varepsilon$ 成立以后（ε 是预先给定的常数），学习式搜索的复杂度将降为多项式阶。

基于上述思想的一个函数逼近式学习式搜索算法 A-Bn 可描述如下：

算法 A-Bn (for trees)

step1. 若 A-Bn 是首次求解 TP_i 中的问题，则

{置 $n \leftarrow 0$，$k_{\max} \leftarrow 0$，启发函数 $h_n(y) \equiv 0$;}

否则　/*已学到次数为 n 的启发函数 $h_n(\cdot)$ */

{置 $h(x, t) \triangleq h_n(y) = B_n(h^*, y)$; 其中 y 是 (x, t) 的康托数的映射值.};

step 2. 将初始节点 s 放入 OPEN 表，计算 $f(s) \leftarrow g(s) + h(s)$;

step 3. 若 OPEN 表空，失败退出；否则在 OPEN 中选择 f 值最小的节点，记其为 n';

step 4. 若 n' 是目标节点 t，则

{调用学习子过程 L-Bn ($h_n, n, k_{\max}, n'$); 成功退出；解路径可追踪 n' 的指针而得到.};

否则继续;

step 5. 将 n' 放入 CLOSED 表，扩展 n' 生成其全部子节点 n'_i，为 n'_i 计算 $f(n'_i)$，将 n'_i 放入 OPEN 表，并为其设置指向 n' 的指针;

step 6. 转向 step. 3.

算法 A-Bn 调用的几个过程的定义为

1) 计算两节点 $(\dot{x}, \dot{y})$ 的康托数之[0,1]映射值的公式为

$$A = \min\{\text{code}(\dot{x}) | \dot{x} \in \text{STATES}\},\quad B = \max\{\text{code}(\dot{y}) | \dot{y} \in \text{STATES}\},$$

$$y = \frac{\Pi^2[\text{code}(\dot{x}), \text{code}(\dot{y})] - A}{B - A}.$$

2) 学习子过程 L-Bn. 该子过程根据新学习到的解路径，抽取新的 $h^*(\cdot)$ 值，用来提高 $h_n(y)$ 的次数 n

L-Bn($h_n, n, k_{\max}, t$)/*$k_{\max}$ 是已学到的 $h_{n+1}(y)$ 部分系数的个数*/

{若 $\frac{k_{\max}+1}{n+1} \in \{y(a_i, t) | a_i \in \text{path}^*(s, t)\}$，则

置 $k_{\max} \leftarrow k_{\max} + 1$，并记取 $h^*\left(\frac{k_{\max}}{n+1}\right)$ 之值;

若 $k_{\max}$ 等于 $n+1$，则

置 $n \leftarrow n+1$，$k_{\max} \leftarrow 0$，$h(x, t) = B_{n+1}(h^*, y)$;

}.

4 对算法 A-Bn 的复杂度分析

对搜索算法的复杂度,本文采用 Nilsson 的定义[1],以算法所扩展的不同节点总数来度量一搜索算法的复杂度.

关于算法 A-Bn 在解题次数充分多之后的渐近最坏复杂度,可以证得以下结论:

定理 1. 当算法 A-Bn 构造的启发函数 $h_n(y)$ 的次数 $n \geqslant \frac{M}{(\log_m N)^3}$ 之后,其最坏复杂度是解路径长 N 的多项式函数,即 $E_{\text{A-Bn}}(N)|_{n>\frac{M}{(\log_m N)^3}} = O(\text{poly}(N))$.

证明(梗概). 由 Bernstein 定理可知, n 取定后, 有 $\varepsilon = \sqrt[3]{\frac{4m\theta^2}{n}}$, 其中 θ 是 $h^*(\cdot)$变化率的最大值.再由 A-Bn 的择点原则及 PS 是 m 元一致树可推得: 搜索树上任意一节点 x 均满足 $\text{branch}(x) \leqslant \varepsilon(n)$, 其中 branch (x) 表示 x 离开主路径的距离. 于是可将算法 A-Bn 的最坏复杂度表示为 (n, N) 的函数

$$E_{\text{A-Bn}}(n, N) = (m-1)\left[\frac{m^{\varepsilon(n)}-1}{m-1}\right][N-\varepsilon(n)] + [N-\varepsilon(n)] + \frac{m^{\varepsilon(n)+1}-1}{m-1},$$

其中 $0 \leqslant \varepsilon(n) \leqslant N$. 当 n 增大至 $n \geqslant \frac{M}{(\log_m N)^3}$ 时, 有 $\varepsilon(n) = \sqrt[3]{4\theta^2} + 1$, 代入上式即可得到本定理的结论.

定理 1 表明,算法 A-Bn 是满足定义 1 的学习式搜索算法之一.

5 结 语

人工智能搜索理论的发展表明,搜索必须与学习相结合[2,4], 因此学习式搜索被提出来. 在学习式搜索中,不再追求搜索算法每次解题都仅用多项式阶的复杂度(已经证明这是不可能的),而是要求算法在解决了充分多的同类问题之后, 能够用多项式阶的复杂度解决所有后来提交的同类问题. 本文给出了一个这样的学习式搜索算法.

参 考 文 献

[1] Hart P, Nilsson N J and Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE trans. on Systems, Man and Cybernet*,1968,**4**(2): 100—107.

[2] 张 伟,俞瑞钊,何志均,可采纳搜索算法最坏复杂度的下确界,计算机学报,1990,**13**(6): 450—455.

[3] Zhang B. and Zhang, L., A new heuristic search technique——algorithm SA, *IEEE Trans on PAMI*, 1985,**7**(1): 103—107.

[4] 张 伟,俞瑞钊. 模式分类式学习搜索算法 SCDF. 模式识别与人工智能,1991,**4**(3): 10—16.

POLYNOMIAL APPROXIMATION BASED LEARNING SEARCH

ZHANG WEI　LIU JIREN　LI HUATIAN

(Computer Science Department, Northeast University of Technology, Shenyang 110006)

ABSTRACT

In this paper, Polynomial Approximation method and theory are introduced into the research of Learning Search of Artificial Intelligence. In this way, we can use a search algorithm repeatedly to construct a heuristic estimate function $h(\cdot)$ which uniformly approximates to the optimal estimate function $h^*(\cdot)$ with arbitrarily high precision. One of such learning setrch algorithms, A-B_n, is presented and it is shown that, when the number of training samples becomes large enough, the worst-case complexity of A-B_n can be reduced to O(poly(N)), where N is the length of the optimal solution path, poly (N) is a polynomial of N.

Key words: Artificial Intelligence; Heuristic Search; Machine Learning; Complexity.