ISSN1000-9825 Journal of Software 软件学报 2000,11(1):85~90

基于遗传算法的Fuzzy规则自动获取的研究*

陈明[1,2] 王静[1,2] 沈理[3]

[1](石油大学计算机科学与技术系 北京 102200)

[2](中国科学院机器人学开放研究实验室 沈阳 110015)

[3](中国科学院计算技术研究所 北京 100080)

摘要 为了实现Fuzzy规则自动获取,进而构造高性能智能系统和解决智能系统的瓶颈问题,研究了利用遗传算法自动获取规则的方法以及遗传算法的组合优化能力.模拟结果表明,这是一种有效地获取Fuzzy规则的方法.

关键词 遗传算法,Fuzzy规则,规则自动获取.

中图法分类号 TP181

机器学习的研究已成为人工智能的核心之一,知识获取问题已成为各类智能系统的瓶颈.模糊高性能智能系统相对于传统的方法具有开发周期短、不需要建立数学模型以及具有非线性等特点,因而得到了广泛的应用.模糊高性能智能系统中的一个困难问题是模糊规则的获取.通常,规则是由具有多年经验的领域专家总结或者向领域专家学习而得出的,而对专家的知识进行分析进而抽取规则是比较困难的工作.于是,关于模糊规则的自动获取的研究引起了人们的重视,近年来已成为一个较活跃的研究领域.本文通过对遗传算法的研究,提出了一种基于遗传算法的模糊规则的自动获取方法.

在Darwinian进化论、门德尔-摩根遗传学说的启示下,在70年代初期,美国Michigan大学的John Holland教授从对生物例子的研究中提出了遗传算法.它模拟生物进化过程(通过细胞核中染色体及其基因的遗传变异机制),利用简单的编码技术作用于称为染色体的二进制串来解决复杂的规划、优化问题.它体现了Darwinian进化论中"物竞天择,适者生存"的基本思想.遗传算法植根于自然遗传学与计算机科学,能够提供鲁棒性搜索,是一种多点并行、依据概率传输规则进行搜索的算法.该算法对问题本身(如是否可导、单峰值、连续等)几乎没有要求,它所需要的仅仅是对算法产生的每个染色体进行评价,计算每个染色体的适应度,并基于适应度来选择染色体,使适应度好的染色体比适应度差的染色体有更多的繁殖机会.因此,该算法尤其适合解决组合优化及规则获取等非线性问题[1,2].

1 遗传算法

遗传算法(GA)可形式化描述为 $GA=(J,P_{init},R_M,\otimes_M,\nabla_M,\Gamma,\tau)$.其中$J$为种群规模.$P_{init}=(P_1,P_2,\ldots,P_J)$为初始种群,$P_i(1\leqslant i\leqslant J)$为种群中的个体,是特定长度的二进制串,$P_i$由串中二进制位(基因)决定.$R_M$,$\otimes_M$,$\nabla_M$分别为遗传算法中的3个基本算子:再生($R_M$:reproduction)、交叉($\otimes_M$:crossover)和突变($\nabla_M$:mutation),通过这3个基本算子就可以从一个初始种群P_{init}出发,产生期望的改良种群.Γ是评价个体适应度的函数,$f_i=\Gamma(P_i)$,f_i表示个体P_i的适应度.Γ可表示为从M到R的映射$\Gamma:M\to R$,其中M是所有个体组为的集合,R为实数集合.τ为算法终止条件,通常表示为算法执行的最大迭代次数的形式.

* 本文研究得到国家自然科学基金(No.19971053)、国家863高科技项目基金(No.863-306-01-07-2)和中国科学院机器人学开放研究实验室基金资助.作者陈明,1949年生,教授,主要研究领域为计算智能,分布并行计算.王静,女,1964年生,讲师,主要研究领域为计算智能,分布并行计算.沈理,1937年生,研究员,博士生导师,主要研究领域为软计算,计算机体系结构.

本文通讯联系人:陈明,北京 102200,石油大学计算机科学与技术系

本文1998-11-13收到原稿,1999-04-13收到修改稿

1.1 再生操作

再生操作根据种群中每个个体的适应度,按与适应度成比例的概率选择参与产生下一代的个体.再生操作的作用效果是提高了种群的平均适应度,并是以损失种群的多样性为代价的,它不产生新的个体.这是一种自然选择的人工模型[3].

记 R^n 为 n 维实空间,则种群适应度 $F=\{f_1,f_2,\ldots,f_n\}\in R^n$,其中 $f_i(1\leqslant i\leqslant n)$ 为个体 P_i 的适应度.

定义 1. 设 $\mathscr{R}$ 是所有种群组成的集合,种群 $\mathscr{P}\in\mathscr{R}$,$M$ 是所有个体组成的集合,种群适应度 $F\in R^n$,再生算子 $R_M:\mathscr{R}\times R^n\to M$ 定义为

$$R_M(P,F)=P_i,$$

其中 R_M 为论域 M 上的再生操作符;个体 $P_i\in P$,且 P_i 按概率 $f_i\Big/\sum_{1\leqslant j\leqslant n}f_j$ 被选择.

1.2 交叉操作

与再生算子不同,交叉算子可以产生新的个体,从而检测搜索空间中新的点.再生算子每次仅作用于 1 个个体上,而交叉算子每次作用在种群中随机选取的两个个体上.交叉算子产生两个子代串,它们一般与父代串不同,每个子代串都包含两个父代串的遗传物质.

定义 2. 设 l 为个体 P_i 的长度,$\Pi=\{1,2,\ldots,l\}$ 为个体 P_i 上位置指标的集合,定义所有可能的交叉位段的集合为

$$\Delta=\{(i,j)\mid(i,j)\in\Pi\times\Pi,i\leqslant j\}.$$

定义 3. 设 $Z\subseteq\Delta$,M 是所有个体组成的集合,个体 $A,B,A',B'\in M$,交叉算子 $\otimes_M:M\times M\times\Delta\to M\times M$ 定义为

$$\otimes_M(A,B,Z)=\{A',B'\}.$$

其中 $\forall i\in\Pi$:

$$\begin{cases}A'_i=B_i\\B'_i=A_i\end{cases}\quad\text{当 } n\leqslant i\leqslant m,(n,m)\in Z;$$

$$\begin{cases}A'_i=A_i\\B'_i=B_i\end{cases}\quad\text{其他情况下.}$$

A_i,B_i,A'_i 和 B'_i 分别表示个体 A,A,A' 和 B' 中的第 i 位的值.

在最简单的情况下,取 $Z=\{(k,l)\}$,我们称这种最简单的交叉算子为单点交叉算子,称 k 为交叉位置.

例如,在初始种群中两个体 A_1,A_2 被选中进行交叉,个体长度 $l=5$,设交叉前:

个体的位置指标	1	2	3	4	5
A_1	0	1	1	0	1
A_2	1	1	0	0	0

若 $Z=\{(4,5)\}$,A'_1 和 A'_2 为 A_1 和 A_2 交叉后产生的两个新个体,则交叉后:

个体的位置指标	1	2	3	4	5
A'_1	0	1	1	0	0
A'_2	1	1	0	0	1

根据交叉位段集合的不同,还可进行多点交叉.只有经过交叉,才能出现新的个体.正是再生和随机化交叉的信息交换给予了遗传算法巨大的生命力.

1.3 突变操作

突变算子以一个很小的概率 P_m 随机地改变个体串上的某些位,若参数本身为二进制编码,则突变只是简单地将 1 变为 0 或将 0 变为 1.

定义 4. 设 $\Omega=P(\Pi)$ 是 Π 的幂集,突变算子 $\nabla_M:M\times\Omega\to M$ 定义为

$$\nabla_M(A,X)=A'.$$

其中$\forall i\in\Pi$:

$$A'_i=\sim A_i, \quad \text{当 } i\in X, X\in\Omega;$$
$$A'_i=A_i, \quad \text{其他情况下.}$$

例如,在初始种群中,个体A被选中进行突变操作,个体长度$l=5$,设突变前:

$$A=0\quad 1\quad 1\quad 0\quad 1,$$

若$X=\{2,4\}$,A'为进行突变操作后产生的新个体,则

$$A'=0\quad 0\quad 1\quad 1\quad 1.$$

比之再生和交叉算子,突变算子是遗传算法中次要的算子.它在恢复种群多样性方面具有潜在的作用.在人工遗传系统中,突变操作是防止重要特征过早丢失的一种保险策略.通过对生物例子的研究,还可总结出许多其他遗传操作符,然而再生、简单交叉和突变已被证明,它们不仅计算简单而且处理许多重要的优化问题时也非常有效[4].遗传算法的主要步骤如下:

(1) 初始化第0代种群P_{init}(随机产生);

(2) 对种群P_{init}迭代执行步骤(i)~(iii),直到满足终止条件τ;

(i) 依概率选择遗传算子R_M,$\otimes_M$,∇_M;

(ii) 计算种群中每个个体的适应度$f_i=\Gamma(P_i)$,$1\leqslant i\leqslant\Pi$;

(iii) 根据种群中每个个体的适应度选择参与产生下一代的个体;

(iv) 将所选择的遗传算子作用于选择的个体,并根据所产生的新个体更新种群;

(3) 将任意一代中出现的最好结果指定为遗传算法的结果.

2 模糊推理

对于一个模糊变量x,U是一特定的论域,在U上可定义一个相应的模糊集A,该模糊集由一个个语言项构成.x属于A的程度用隶属度函数$\mu_A(x)\in[0,1]$表示.由于在推理过程中使用的规则为化简模糊规则,因此称其为模糊推理.其规则的一般形式为if〈条件〉then〈结论〉,条件部分是输入变量属于相应的模糊集的隶属度的合取,结论部分是一个常量,它表示输出变量的状态.例如下面的规则i:

$$\text{if } I_1^i \text{ and } I_2^i \text{ and } \dots \text{ and } I_N^i \text{ then } O^i.$$

其中$I_j^i(i\leqslant 1\leqslant K, 1\leqslant j\leqslant N)$表示规则$i$中$x_j$所对应的模糊集的项,具有相同下标的模糊集的项构成一条模糊规则.K是构成一个模糊逻辑系统的模糊规则的条数,N为模糊变量的维数.x_j在规则i中的隶属度函数为$\mu_{I_j^i}(x_j)$.于是,规则i的激活程度H_i为

$$H_i=\bigcap_{j=1}^{N}\mu_{I_j^i}(x_j).$$

计算模糊集的项的合取的方法有很多,这里使用如下方法:

$$H_i=\prod_{j=1}^{N}\mu_{I_j^i}(x_j).$$

对于一个给定的输入向量$X=\{x_1,x_2,\dots,x_N\}$,最后的推理输出(预测输出)是所有规则的合成,其推理输出Q为

$$Q=\sum_{i=1}^{K}(H_iO_i)\Big/\sum_{i=1}^{K}H_i.$$

3 规则表示及规则集的评估

应用遗传算法获取模糊规则,需要解决模糊规则的表示形式和对规则集的评估方法这两个问题.

3.1 规则的表示

个体是遗传操作的信息单元,因此需要首先将模糊规则转换为个体的形式.规则左侧的隶属度可有多种表示形式,采用等腰三角形表示,于是,一个隶属度可由隶属度三角形的底边中心c_i和底边宽w_i来确定.规则的条

件部分被表示为一系列的 c_i 和 w_i，而结论部分则是其本身.因此，第 i 条规则：if I_1^i and I_2^i and ... and I_N^i then O^i 可以表示为

$$(c_1^i, w_1^i)\ (c_2^i, w_2^i)\ \ldots\ (c_N^i, w_N^i);\ O^i.$$

假设每个个体是由 K 条规则组成的，那么每个个体相当于一个模糊推理机.长度为 K 的个体可被表示为

$$\text{rule}_1\quad \text{rule}_2\quad \ldots\quad \text{rule}_K$$

3.2　规则集的评估

遗传算法需要根据适应度函数的值进行搜索.因此，利用遗传算法获取规则必须解决个体评估问题，以便评价每个个体的优劣.在每一个学习周期，适应度较低的个体将被淘汰，适应度最高的个体被认为是最满意解.对于每个训练实例，预测输出与实际输出之间的平均距离越小，该个体的适应度就越高，预测输出就越接近实际输出.在这里，适应度的评估覆盖所有训练实例.假设 fitness 表示某个个体评估函数，则有

$$fitness = \frac{1}{L} \cdot \sum_{j=1}^{L} \frac{\alpha}{1 + \beta |(Q'_j - Q_j)/Q_j|},$$

其中 L 为训练实例数目；Q_j 为实例 j 的期望输出；Q'_j 为个体 i 在实例 j 上的预测(实际)输出；α, β 为比例系数.

4　模糊规则获取的遗传算法

这里采用 3 种基本的遗传操作.个体为规则链，基因为一条规则.算法描述如下.

算法 1. 模糊规则获取的遗传算法

//gen 为代计数器，MAX*GEN* 表示要求指标

//count 为个体计数器，交叉概率为 P_c，突变概率为 P_m

//在整个过程中保持种群大小不变，算法终止条件 τ 为迭代次数小于 MAX*GEN*

(1) 初始化第 0 代种群 P_{init}，$gen \leftarrow 0$；

(2) while $gen <$ MAX*GEN* do

(i) 对种群中每一个体进行评估；

(ii) 使用适应度概率(p_s)策略随机选择一对参与产生下一代的个体；

(iii) 对父代个体以概率 p_c 进行交叉操作；

(iv) 对新产生的个体以概率 p_m 进行突变操作；

(v) 根据新产生的个体更新种群；

(vi) $gen \leftarrow gen + 1$.

endwhile

(3) 将任意一代中出现的最好结果指定为遗传算法的结果

其中，

· 个体的选择概率为 $p_s(i)$，与其适应度成正比，计算方法如下所示：

$$P_s = fitness \Big/ \sum_{j=1}^{n} fitness_j.$$

· p_m 根据经验选取，突变操作按以下方式进行：

$$c(\text{gen}+1) = c(\text{gen}) * (1 + \text{rnd}(a)),$$

$$w(\text{gen}+1) = w(\text{gen}) * (1 + \text{rnd}(a)),$$

rnd(a)为随机数发生器，用来产生小于 a 的随机数.

· 第 0 代种群 P_{init} 由训练实例产生，算法描述如下：

算法 2. 初始化种群 P_{init}

//c_1, c_2 分别为个体及基因计数器

//POPSIZE 为种群大小，*LINDIVIDUAL* 为每个个体中的规则数

(1) $c_1 \leftarrow 0; c_2 \leftarrow 0$；

(2) **while** $c_1 \leqslant POPSIZE$ **do**

(i) **while** $c_2 \leqslant LINDIVIDUAL$ **do**

(a) 从 *examplelist* 中取出一个训练实例,产生 c_i, w_i

(b) 根据例子输出产生 o_i

(c) $c_2 \leftarrow c_2+1$;

endwhile

(ii) $c_1 \leftarrow c_1+1$;

endwhile

· 交叉采用单点交叉,交叉位置随机产生,如下所示:

交叉前:

individual1: rule1 rule2 | rule3 rule4 rule5 rule6

individual2: rule7 rule8 rule9 | rule10 rule11 rule12

交叉后:

newindividual1: rule1 rule2 rule10 rule11 rule12

newindividual2: rule7 rule8 rule9 rule3 rule4 rule5 rule6

5 模拟结果

通过对函数 $f(x)=x$ 及 $f(x)=(x-2)^2$ 进行模拟,说明遗传算法获取模糊规则的效果.

5.1 函数 $f(x)=x$ 的模拟结果

5.1.1 模拟过程

根据 $f(x)=x$ 生成训练实例集合 *examplelist*,由该例子序列初始化第 0 代种群,依据上述遗传算法进行操作,经过 82 代,从最终种群中选取适应度最高的个体作为最后结果.依据该个体,计算出 300 个输入变量的实际输出.

参数初值设定及模拟结果见表 1.

Table 1

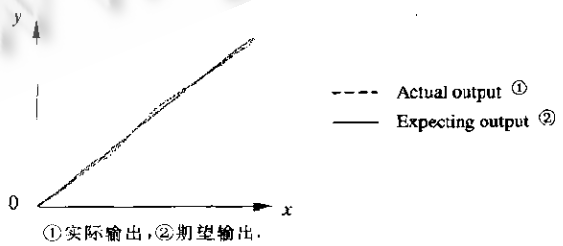
表 1

Rule number①	Error②	The base line width of membership triangle③	Mutation probability④	Scale constants⑤ α,β	Number of training example⑥	Number of simulation variable⑦
10	0.037 6	0.6	0.003	$\alpha=1,\beta=1$	120	300

①规则数目,②误差,③隶属度三角形底边宽,④突变概率,⑤比例系数,⑥训练实例数目,⑦模拟变量数目.

其中变量数目是指在对获取的规则进行测试时,输入变量的数目;误差为实际输出和期望输出之间的均方误差.

5.1.2 实际输出与期望输出的比较(如图 1 所示)



①实际输出,②期望输出.

Fig. 1 Comparing between actual output and expecting output of $f(x)=x$ function

图 1 函数 $f(x)=x$ 的实际输出与期望输出的比较

5.2　函数 $f(x)=(x-2)^2$ 的模拟结果

5.2.1　模拟过程

根据生成训练实例集合 *examplelist*,由该例子序列初始化第 0 代种群,依据上述遗传算法进行操作,经过 96 代,从最终种群中选取适应度最高的个体作为最后结果.依据该个体,计算出 300 个输入变量的实际输出.

参数初值设定及模拟结果见表 2.

Table 2

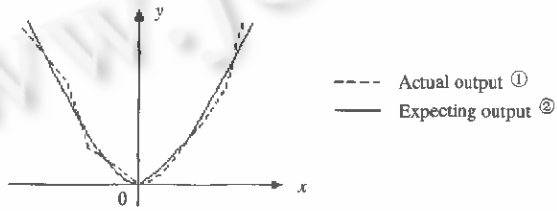
表 2

Rule number①	Error②	The base line width of membership triangle③	Mutation probability④	Scale constants⑤ α,β	Number of training example⑥	Number of simulation variable⑦
14	0.090 2	0.25	0.005	$\alpha=1,\beta=1$	180	300

①规则数目,②误差,③隶属度三角形底边宽,④突变概率,⑤比例系数,⑥训练实例数目,⑦模拟变量数目.

其中,变量数目是指在对获取的规则进行测试时,输入变量的数目;误差为实际输出和期望输出之间的均方误差.

5.2.2　实际输出与期望输出的比较(如图 2 所示)



①实际输出,②期望输出.

Fig. 2　Comaring between output and expecting output of $f(x)=x^2$ function

图 2　函数 $f(x)=x^2$ 的实际输出与期望输出的比较

以上模拟结果表明,遗传为解决智能系统中知识获取这个瓶颈问题提供了一种新的、前景广阔的方法.基于遗传算法的 Fuzzy 规则的方法是一种非常简单而又非常有效的方法.在智能系统的知识获取领域必将得到广泛的应用.

参考文献

1　Goidberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. New York: Addison-Wesly Publishing Company, Inc., 1989

2　Kitano H. Genetic algorithms. Journal of Japanese Society for Artificial Intelligence, 1992,7(1):26～37

3　Holland J H. Adaptation in Natural and Artificial Systems. Michigan: University of Michigan Press, 1975

4　Michalewicz Z. Genetic Algoritms+Data Structure=Evolutionary Programs. Berlin: Springer-Verlag, 1994

Research on Automatic Fuzzy Rule Acquisition Based on Genetic Algorithms

CHEN Ming[1,2]　WANG Jing[1,2]　SHEN Li[3]

[1](*Department of Computer Science and Technology　University of Petroleum　Beijing*　102200)

[2](*Robot Opening Research Laboratory　The Chinese Academy of Sciences　Shenyang*　110015)

[3](*Institute of Computing Technology　The Chinese Academy of Sciences　Beijing*　100080)

Abstract　In order to realize fuzzy rule acquisition automatically, and to construct high performance intelligent system and solve the bottle neck problem in the intelligent system, the method of automatic rule acquisition using genetic algorithms and the combination optimization ability of genetic algorithms have been studied. The simulation results show that this is an effective fuzzy rule acquisition method.

Key words　Genetic algorithm, fuzzy rule, automatic fuzzy rule acquisition.