

遗传算法

2019年度南京大学“专创融合”特色示范课程培育项目

高 阳

<http://cs.nju.edu.cn/rl>, 2019.11.05

遗传算法

从神经机制到自然选择

高 阳

<http://cs.nju.edu.cn/gaoy>, 2019.11.05

大纲

动机

遗传算法

模式理论

学习分类系统

动机

人工神经网络：从生物神经系统中得到灵感

涌现学习模型(Emergent)：模仿生物的生命演化形式

遗传算法：

- ✓ 学习是问题候选假设(解)在进化中的一种竞争
- ✓ 较好的候选假设(解)在自然选择中不断演化

人工生命：

- ✓ 模拟生物进化的条件



生物进化和文化进化

□ 达尔文认为 “... 对于这种缓慢地并美好地使每一类型适应于最复杂的生命关系的力量，我无法看到有什么限制 ...”

- ✓ 在连续的世代中有选择地淘汰适应性较低的个体。通过这样简单地过程，生物体的适应性得到提高
- ✓ 演化和涌现出现在群体中
- ✓ 选择的压力不仅来自外部环境，也来自群体中个体相互作用中



大纲

动机

遗传算法

模式理论

学习分类系统

遗传算法 (genetic algorithms)

□ 遗传算法GA：受生物进化启发的学习算法

✓ 假设搜索：一般到特殊序？简单到复杂序？ ✗

✓ 通过对当前最好的假设重组来产生后续假设。 ✓

✓ 生成并测试(generate-and-test)的柱状搜索(beam-search)。

假设的各个部分相互作用，每一部分对总体的影响难以建模

遗传算法一般形式

1. $t:=0$, 初始化种群 $P(t)$;

2. 如果不满足终止条件

- 评估种群 $P(t)$ 中每个染色体的适应度
- 根据适应度函数选择部分染色体
- 根据所选择的染色体产生后代
- 根据 $P(t)$ 中染色体的适应度, 选择被替换的染色体, 以后代替换
- $t:=t+1$

3. 终止

如何表示染色体?  什么是适应度函数?

 如何选择染色体?  如何产生后代?

假设的表示

□ If-then规则： 二进制位串的编码

□ 例：

✓ Outlook: Sunny, Overcast, Rain

✓ Wind: Strong, Weak

✓ PlayTennis: Yes, No

方法一： 每个属性-值用二进制的1位表示

假设的表示

□ 单属性Outlook

- ✓ 100: (Outlook=Sunny)
- ✓ 110: (Outlook=Sunny \vee Overcast)
- ✓ 111: (Outlook=Sunny \vee Overcast \vee Rain)

□ 多属性合取 (Outlook=Sunny \vee Rain) \wedge (Wind=Strong)

- ✓ Outlook Wind
- ✓ 101 10

□ If-then规则 If Wind=Strong Then PlayTennis=yes

- ✓ Outlook Wind PlayTennis
- ✓ 111 10 10



PlayTennis = '11'?

语义上没有解释

假设的表示

□ 决策属性PlayTennis

- ✓ 用一位表示
- ✓ 1: (PlayTennis=Yes)
- ✓ 0: (PlayTennis=No)
- ✓ #: (No Care, PlayTennis=No \vee Yes)

□ 思考：旅行商问题解的表示？

- ✓ 表示的自然性，易用性
- ✓ 所有假设的合法性
- ✓ 潜在假设的语义丰富性

应用实例

有一个贼想在一次抢劫中带走赃物，他只有一个能承受50磅的包，现在这里有8件物品，每件物品都有自己的货币价值和重量，他应该如何选取来使自己取得货币价值总和最高的几件物品？

| | | |
|---------|--------|--------|
| ✓ 皇冠 | \$5000 | 3 磅 ✓ |
| ✓ 收藏册 | \$2200 | 5 磅 ✓ |
| ✓ 电视 | \$2100 | 40 磅 ✓ |
| ✓ 笔记本电脑 | \$2000 | 8 磅 |
| ✓ 餐具 | \$1200 | 10 磅 |
| ✓ 音响 | \$800 | 25 磅 |
| ✓ 掌上电脑 | \$600 | 1 磅 ✓ |
| ✓ 时钟 | \$300 | 4 磅 |



贪心策略

依次选择价值最高的
物品

是最优解吗？

应用实例

□ 染色体: 一个8位的串, 每一位对应一个物品

✓ 1 代表相应的物品被选了; 0 代表没有被选

✓ 染色体=11100000 → (皇冠 + 收藏册 + 高清电视)

✓ 染色体=01101000 → (收藏册 + 高清电视 + 餐具)

□ 适应函数(fitness): 目的是找更高的值集合

✓ $\text{fit}(\text{染色体}) = \text{所有物品的货币价值和}$, 当总重量 >50 时设为0

✓ $\text{fit}(11100000) = 9300$

✓ $\text{fit}(01101000) = 0$

染色体的选择

□ 基于适应度函数的选择

- ✓ 轮盘赌选择(roulette wheel selection)
 - ✓ 与适应度成比例选择
- ✓ 锦标赛选择(tournament selection)
 - ✓ 按预定义概率p选择较大适应度的假设
 - ✓ 按概率1-p选择其他假设
- ✓ 排序选择(rank selection)
 - ✓ 根据序而不是适应度进行选择

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^p Fitness(h_j)}$$



□ 假设选择的重要性

- ✓ Exploration 与 Exploitation 的平衡!

遗传算子

□ 遗传算子 (GA operator)

- ✓ 对从当前群体中选择的染色体进行重组，以产生后代

□ 交叉(crossover)

- ✓ 选择两个候选个体，分解每一个个体，然后交换分量形成两个新的候选个体
 - ✓ 单点交叉
 - ✓ 两点交叉
 - ✓ 均匀交叉



遗传算子

□ 变异(mutation)

- ✓ 选择一个候选个体，随机的选择一位，然后取反

变异前：

000001110000000010000

变异点

变异后：

000001110001000010000

应用实例

□ 第0代(随机选取)

✓ 11100000 (fit = 9300)

01101000 (fit = 0)

✓ 11001011 (fit = 9300)

11010000 (fit = 9200)

✓ 00010100 (fit = 2800)

01001011 (fit = 4300)

✓ 11110111 (fit = 0)

10011000 (fit = 8200)

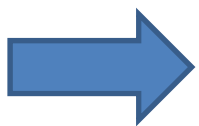
| | | |
|---------|--------|--------|
| ✓ 皇冠 | \$5000 | 3 磅 ✓ |
| ✓ 收藏册 | \$2200 | 5 磅 ✓ |
| ✓ 电视 | \$2100 | 40 磅 ✓ |
| ✓ 笔记本电脑 | \$2000 | 8 磅 |
| ✓ 餐具 | \$1200 | 10 磅 |
| ✓ 音响 | \$800 | 25 磅 |
| ✓ 掌上电脑 | \$600 | 1 磅 ✓ |
| ✓ 时钟 | \$300 | 4 磅 |

应用实例

□ 第1代(随取最好的4个，交叉并伴随更小概率的变异)

✓ 111000|00 (fit = 9300)

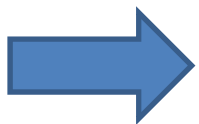
110010|11 (fit = 9300)



11100011, 11001001

✓ 110|10000 (fit = 9200)

100|11000 (fit = 8200)



11011000, 10010000

应用实例



| | |
|-------------------------|-----------------------|
| ✓ 11100000 (fit = 9300) | 01101000 (fit = 0) |
| ✓ 11001011 (fit = 9300) | 11010000 (fit = 9200) |
| ✓ 00010100 (fit = 2800) | 01001011 (fit = 4300) |
| ✓ 11110111 (fit = 0) | 10011000 (fit = 8200) |

□ 第1代(替换第0代适应度最低的4个)

| | |
|-------------------------|------------------------|
| ✓ 11100000 (fit = 9300) | 11100011 (fit = 0) |
| ✓ 11001011 (fit = 9300) | 11010000 (fit = 9200) |
| ✓ 11001001 (fit = 8700) | 11011000 (fit = 10400) |
| ✓ 10010000 (fit = 7000) | 10011000 (fit = 8200) |

应用实例

□ 第2代(随取最好的4个，交叉并伴随更小概率的变异)

✓ 1101|1000 1110|0000  11101000 11101000
✓ 1100101|1 1101000|0  11001010 11010001

□ 第2代(替换第1代适应度最低的4个)

✓ 11100000 (fit = 9300) 11010000 (fit = 9200)
✓ 11001011 (fit = 9300) 11010000 (fit = 9200)
✓ 11101000 (fit = 0) 11011000 (fit = 10400)
✓ 11001010 (fit = 9000) 11010001 (fit = 9500)

讨论



□ 优势

- ✓ 无需理解问题内部的相关性和因果性
- ✓ 以一个随机的群体开始，以适应度作为某种”启发式”
- ✓ ”进化论”保证整个种群的演化

并行爬山法的优势



讨论

□ 未解决的问题

- ✓ **表示的问题：**编码不规范及编码存在表示的不准确性
- ✓ **约束的问题：**单一的遗传算法编码不能全面地将优化问题的约束表示出来。考虑约束的一个方法就是对不可行解采用阈值，这样，计算的时间必然增加

思考旅行商问题的表示和约束！

- ✓ **搜索效率的问题：**遗传算法通常的效率比其他传统的优化方法低；遗传算法容易出现过早收敛
- ✓ **理论保证的问题：**遗传算法对算法的精度、可行度、计算复杂性等方面，还没有有效的定量分析方法

大纲

动机

遗传算法

模式理论

学习分类系统

模式定理

□ 模式定理(Schema theorem)

- ✓ Short schemata with large fitness will increase their representation in the population during the evolution

□ 模式

- ✓ 是由0、1和#组成的任意串(#代表No Care)
- ✓ 如模式0#10，包含两个染色体0010, 0110
- ✓ 如模式###11，包含00011, 00111...等8种染色体

John Holland于上世纪70年代提出。

模式定理

□ 模式H的阶, $o(H)$

- ✓ 模式H中确定位置的个数
- ✓ $o(\#\#1\#0)=2$
- ✓ $o(1\#\#11)=3$

□ 模式H的长度, $d(H)$

- ✓ 模式H中第一个确定的位置到最后一个确定位置的距离
- ✓ $d(\#\#1\#0)=2$
- ✓ $d(1\#\#11)=4$

模式的进化

□ $m(s,t)$ 表示在第 t 代种群 p_t 中模式 s 的实例数量

□ 模式理论

- ✓ 根据GA的原理，去推断 $m(s,t+1)$ 的期望值
- ✓ $f(h)$: 染色体 h 的适应度
- ✓ $\bar{f}(t)$: 第 t 代种群染色体的平均适应度
- ✓ n : 种群中个体的总数量
- ✓ $h \in s \cap p_t$: 染色体 h 属于模式 s ，又是 p_t 的成员
- ✓ $\hat{u}(s,t)$: 第 t 代中模式 s 的染色体的平均适应度

模式的进化

□ 轮盘赌选择

在后代中，高于平均适应度的模式出现频率会升高

$$\Pr(h) = \frac{f(h)}{\sum_{i=1}^n f(h_i)} = \frac{f(h)}{n\bar{f}(t)}$$

□ 选择的假设是模式s的实例的概率

$$\Pr(h \in s) = \sum_{h \in s \cap p_t} \frac{f(h)}{n\bar{f}(t)} = \frac{\hat{u}(s, t)}{n\bar{f}(t)} m(s, t)$$

□ 如果选择n次，得到s的实例的期望值是

$$E[m(s, t + 1)] = \frac{\hat{u}(s, t)}{\bar{f}(t)} m(s, t)$$

模式的进化

- 单点交叉的概率 p_c
- 任意染色体任意位变异的概率 p_m
- 模式 s 的阶(确定位数的个数) $o(s)$
- 模式 s 的长度(从最左确定位到最右确定位的距离) $d(s)$
- 染色体的长度 l

$$E[m(s, t + 1)] \geq \frac{\hat{u}(s, t)}{\bar{f}(t)} m(s, t) \left(1 - p_c \frac{d(s)}{l - 1} \right) (1 - p_m)^{o(s)}$$

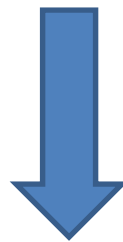
选择步的影响

交叉后仍属于模式 s 的概率

变异后仍属于模式 s 的概率

模式定理

$$E[m(s, t + 1)] \geq \frac{\hat{u}(s, t)}{\bar{f}(t)} m(s, t) \left(1 - p_c \frac{d(s)}{l - 1}\right) (1 - p_m)^{o(s)}$$



- ✓ 适应度越高的模式影响力越大
- ✓ 包含较少确定位的模式(也就是有较多#)影响力越大
- ✓ 确定位彼此靠近的模式影响力越大

大纲

动机

遗传算法

模式理论

学习分类系统

历史发展

■ --1979

- ✓ 1971, John H. Holland, *Classifier system*, “Processing and processors for Schemata”
- ✓ 1978, Holland and Reitman, *Cognitive system/Learning classifier system*, “Cognitive Systems Based on Adaptive Algorithms”

● Two ideas

Survival of the fittest. The First Principle of Evolutionary Computation, Adaptive Behavior and Artificial Life.

Trial and error. Been formalized and developed in the area of Reinforcement Learning.

Two major application areas

Classification problems. / *Adaptive behavior problems*.

历史发展

1980~1989

- ✓ 1980~1986, Holland revised the framework to define what would become the standard system.
- ✓ 1989, Holland's full system was somewhat complex and practical experience found it difficult to realize the envisaged behavior/performance.

Research interesting waned.

1990~1999

1994, Wilson, ZCS, keep much of Holland's original framework but simplifies it to increase understandability and performance.

1995, Wilson, XCS, be applied to a number of hard real-world problems such as data mining, simulation modeling, robotics, and adaptive control.

The following decade has seen resurgence in the use of LCS.

机器学习中的基本问题

- Rule representation
- Credit assignment
- Rule discovery

机器学习中的基本问题

■ Rule representation

- Use a ternary alphabet $\{0,1,\#\}$ to represent rule conditions. And sets of rule conditions are in Disjunctive Normal Form.
- LISP S-expressions.
- More complex data structures such as trees and graphs.

■ Credit assignment

■ Rule discovery

The lower the cardinality of the alphabet, the higher of schemata and the higher the degree of implicit parallelism.

机器学习中的基本问题

■ Rule representation

■ Credit assignment

- ✓ Structure credit assignment
 - ✓ Inter-rule credit assignment
 - ✓ Intra-rule credit assignment
- ✓ Temporal credit assignment

■ Rule discovery

*In realistic situation, exhaustive **exploration** of all possible action paths or rules is not feasible.*

机器学习中的基本问题

■ Rule representation

■ Credit assignment

■ Rule discovery

- ✓ Random generation of new rules can only work for the smallest problems.
- ✓ Use of past experience to generate plausible hypotheses about situations.

How does a system improve its performance in a perpetually novel environment where overt ratings of performance are only rarely available?

Such improvement is not possible unless the environment has repeating (sub-)patterns.

学习分类系统

■ Rule representation

- Answer: Production rule-based

■ Credit assignment

- Answer: is done by heuristics appropriate to the system. Especially GA acts in the population.

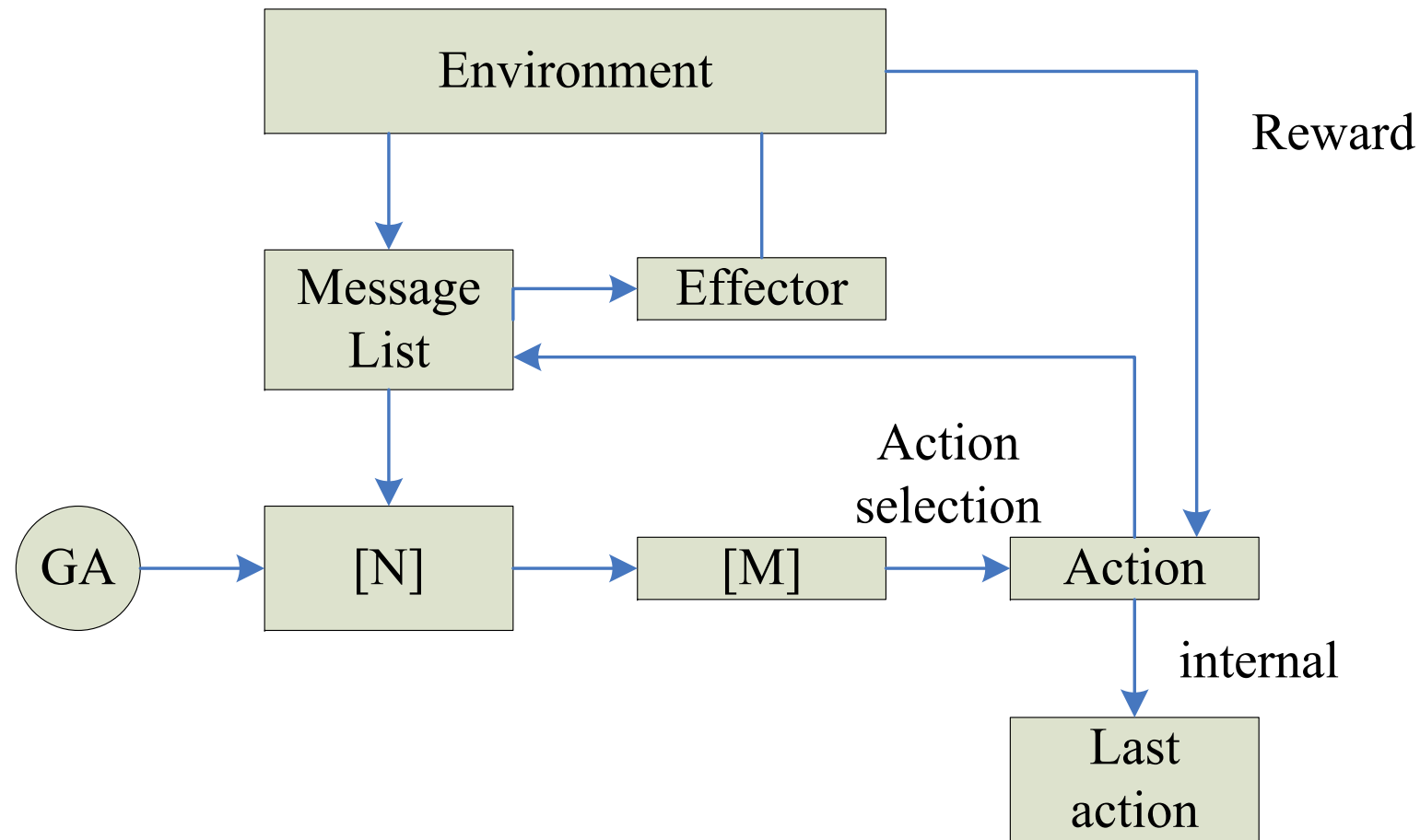
■ Rule discovery

- Answer: is done by temporal difference methods or reinforcement learning-like algorithm.

■ Other

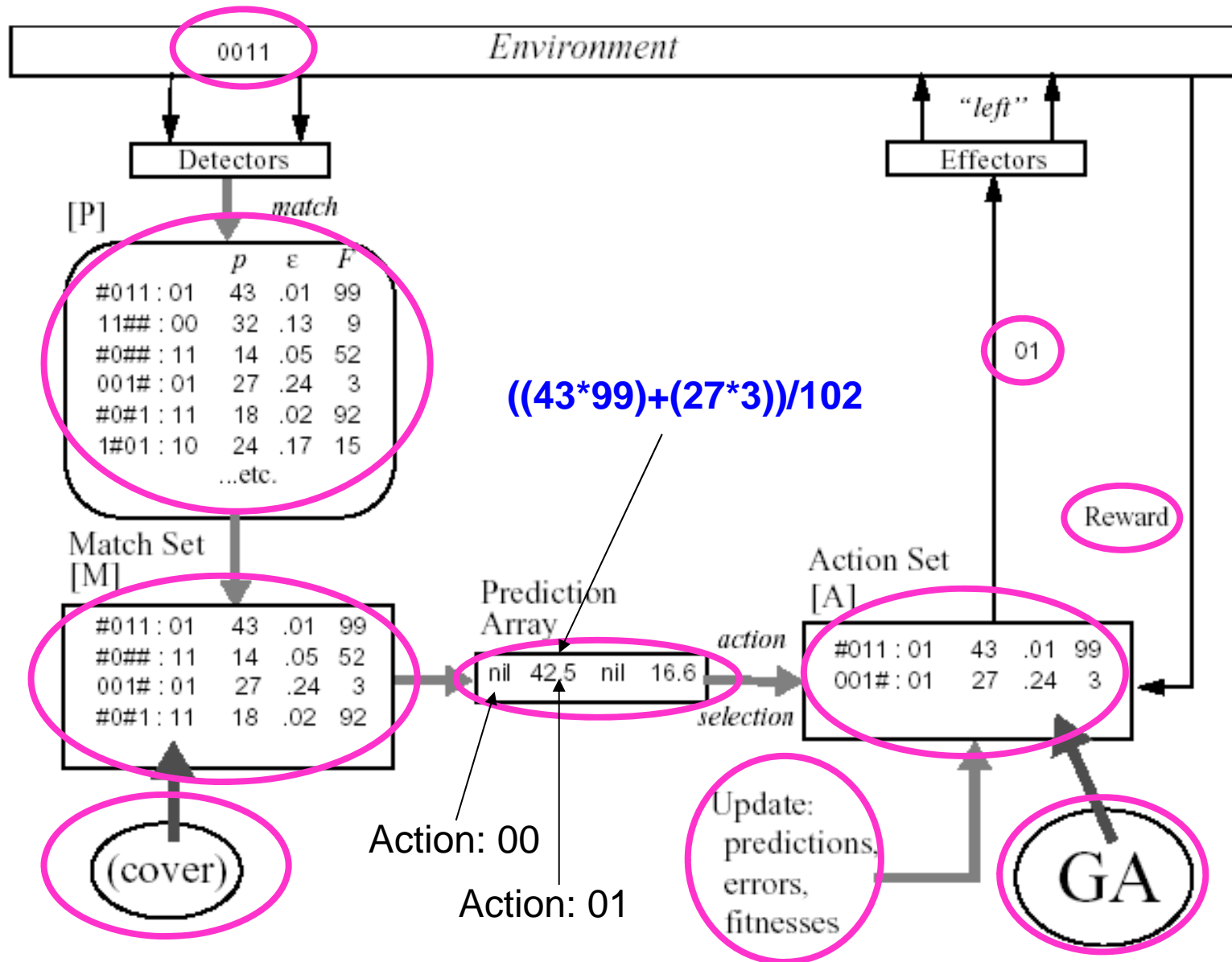
- A message board for communication.
- A competition for rules to become active.
- Parallel firing of rules.

实现(Holland's LCS)

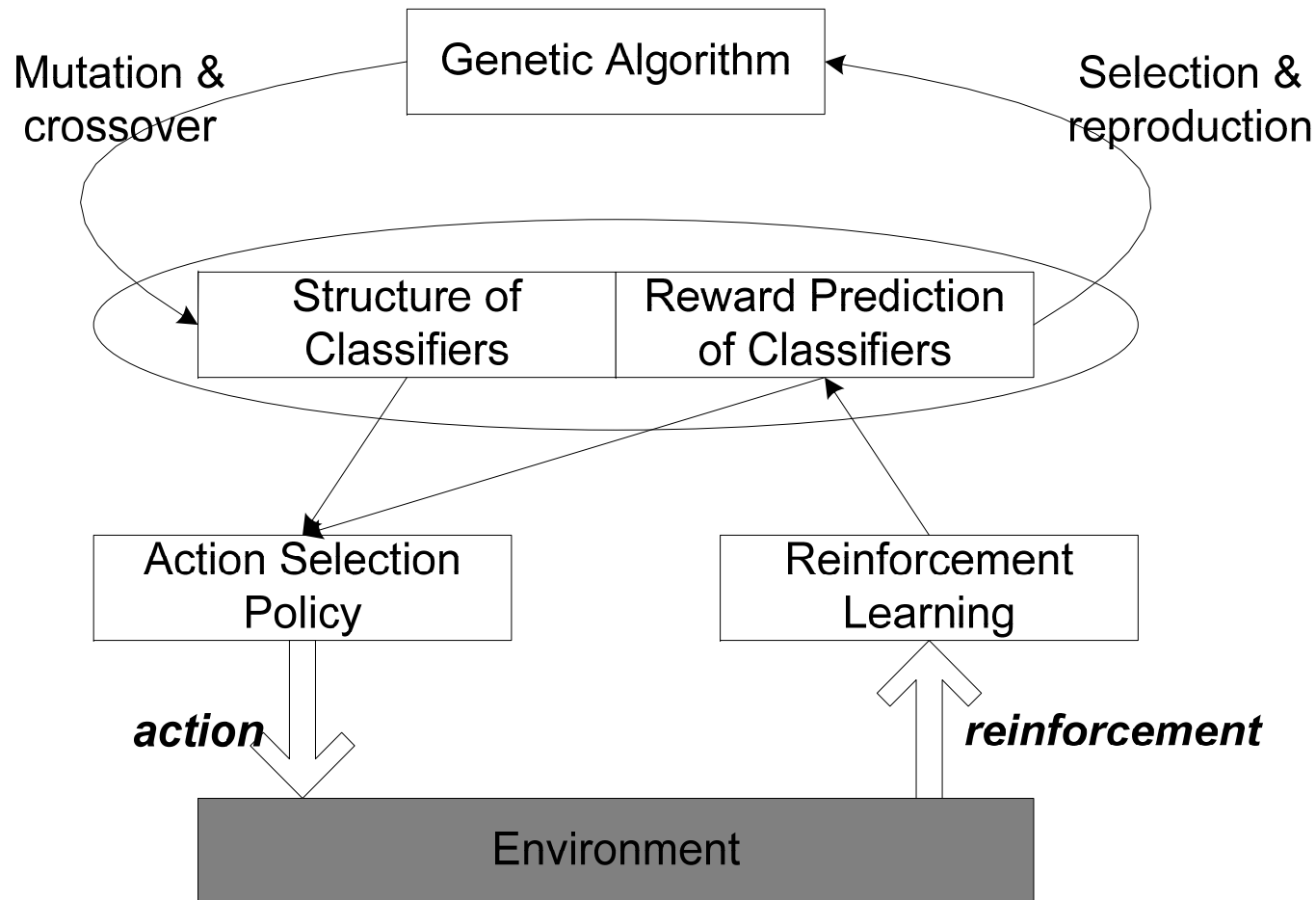


DEMO

Grid maze



学习组件间的交互



传统LCS存在的问题

- The most sever problems are
 - The unrestricted GA application;

The combination of such classifiers often leads to meaningless classifiers.

- The navie fitness approach based on the reward prediction;

The fitness approach based on the reward predictions directly often failed due to reward fluctuation in classifiers.

- The reinforcement learning allocation.

The reward prediction can be misguided by an unsuitable reward backpropagation.

思考和讨论

1. 遗传算法与其他优化算法的区别？
2. 遗传算法的潜在并行性？
3. 有其他理论来证明GA的有效性吗？
4. ZCS, ACS等新的LCS架构.....

实验

题目：实现GA算法。在UCI数据集中分别选择2个数据集（Audiology (Standardized), Credit Approval），进行学习和分类。

评判预测性能的指标：准确率（precision）、召回率(recall)、真阴性（true negative）、真阳性（true positive）（[请查阅资料了解这四个指标的概念](#)）。

谢谢！