

# 演化学习

高 阳, 李文斌

<http://cs.nju.edu.cn/rl>, 2021.4.8

# 大纲

动机

遗传算法

模式理论

其他议题

# 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

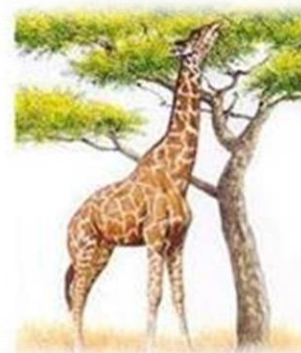
遗传算法：

- ✓ 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：一般假设模型到特殊假设模型？简单假设模型到复杂假设模型？ ✗
- ✓ 通过对当前最好的假设模型重组来产生后续假设模型。 ✓
- ✓ 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
- 

四大  
问题

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?

# 假设模型的表示

□ If-then规则：二进制位串的编码

□ 例：

✓ Outlook: Sunny/Overcast/Rain

✓ Wind: Strong/Weak

✓ Play Tennis: Yes/No

方法一：每个<属性-值>用二进制的1位表示



# 假设模型的表示

## □ 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 Play Tennis
- ✓ 111 10 10



PlayTennis = '11'?

语义上没有解释

# 假设模型的表示

## □ 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
- ✓ 所有假设模型的合法性
- ✓ 潜在假设模型的语义丰富性

如果用实数值表示，将使GA中的部分算子失效

# 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

✓  $\text{fit}(11100000) = 9300$

✓  $\text{fit}(01101000) = 0$

对学习的  
目标函数定量表达

# 种群

## □ 种群

- ✓ 一组染色体及其所计算的适应度
- ✓ GA工作在每一代的种群上。第0代种群中的染色体随机生成

## □ 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) |

# 产生后代：选择父母

## □ 基于适应度函数的选择

- ✓ 锦标赛选择(tournament selection)
  - ✓ 每次从种群中取出一定(N)数量个体(放回抽样)，然后选择其中最好的一个(M)进入子代种群。
  - ✓ 重复该操作，直到新的种群规模达到原来的种群规模。
- ✓ 截断选择(Truncation selection)
  - ✓ 根据适应度排序, 前f个染色体进入下一代种群
  - ✓ 对染色体进行复制，填充至种群规模达到原来的种群规模。

选择的作用：优胜劣汰，适者生存

# 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

$$p^{\alpha} = \frac{F^{\alpha}}{\sum_{\alpha'} F^{\alpha'}} \quad p^{\alpha} = \frac{\exp(sF^{\alpha})}{\sum_{\alpha'} \exp(sF^{\alpha'})}$$



## □ 假设选择的重要性

- ✓ Exploration(搜索) 与 Exploitation(探索/利用) 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



交叉的作用：保证种群的稳定性，  
朝着最优解的方向进化



# 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
- ✓ 常以小概率 $p \approx 1/L$ ( $L$ 是染色体长度) 发生变异

变异前:

000001110000000010000

变异点

变异后:

000001110001000010000

变异的作用：保证种群的多样性，  
避免交叉可能产生的局部收敛

# 后代种群的演化

## □ 简易方案

- ✓ 后代染色体直接替代父代染色体 (与自然吻合)
- ✓ 易丢失优秀解

## □ 精英法(Elitism)

- ✓ 每一代保留上一代最优的染色体，丢弃掉最差个体
- ✓ 往往与选择算子混用

## □ 锦标赛法(Tournaments)

- ✓ 让父母染色体与后代染色体参与竞争，胜者放入下一代种群

# 后代种群的演化

小生境的作用：

门当户对，并行演化

## □ 小生境法(Niching)

- ✓ 将每一代个体划分为若干类
- ✓ 每个类中选出若干适应度较大的个体作为一个类的优秀代表组成一个群
- ✓ 再在种群中，以及不同种群中之间，杂交，变异产生新一代个体群。

## □ 小生境中的选择机制

- ✓ 预选择机制：只用高适应度子代替换父代
- ✓ 排挤机制：预定义原型。所产生的子代要保持和原型的模式不一致，模式相似的个体被替换
- ✓ 共享机制：计算适应度和模式的关联关系，然后共享这种模式

# 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)

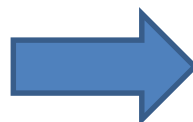
10011000 (fit = 8200)

✓ 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

其他议题

# 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

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

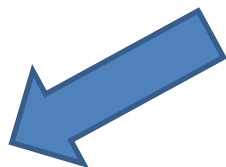
## □ 如果选择n次，得到s的实例的期望值是

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

# 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

其他议题

# 自然计算

## □ 自然计算(Nature Inspired Computation)

- ✓ 模仿自然界特点
- ✓ 一类具有自适应、自组织、自学习能力的模型与算法
- ✓ 遗传算法，蚁群算法，粒子群算法，免疫算法.....
- ✓ 往往用来解决非凸优化问题

# 粒子群优化与蚁群优化

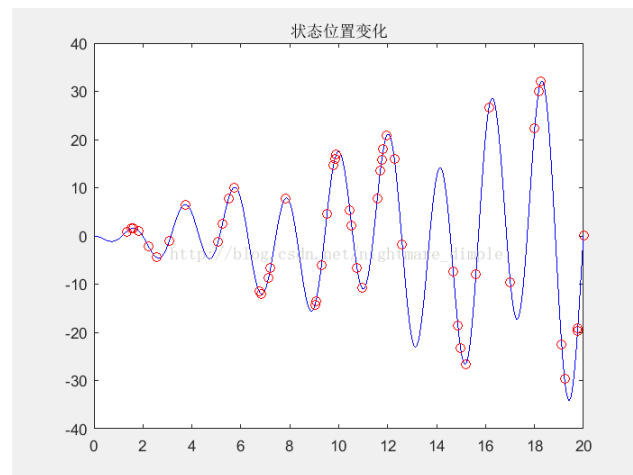
## □ 粒子群优化 (Particle Swarm Optimization, PSO)

✓ 模拟鸟群觅食, 更新移动位置

$$v_{id} = w \times v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})$$

## □ 蚁群优化 (Ant Colony Optimization, ACO)

✓ 模拟蚂蚁觅食, 更新信息素




# 学习分类器系统

## □ 学习分类器系统 (Learning Classifier Systems, LCS)

- ✓ 约翰.霍兰德(John Holland), 复杂系统先驱, 遗传算法之父
- ✓ 一类复杂的认知系统

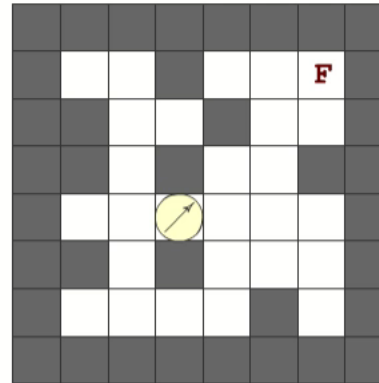
1978, Holland and Reitman, *Cognitive system/Learning classifier system*, “Cognitive Systems Based on Adaptive Algorithms”

### Local perception

0	1	0
0		0
0	1	0

### Current situation

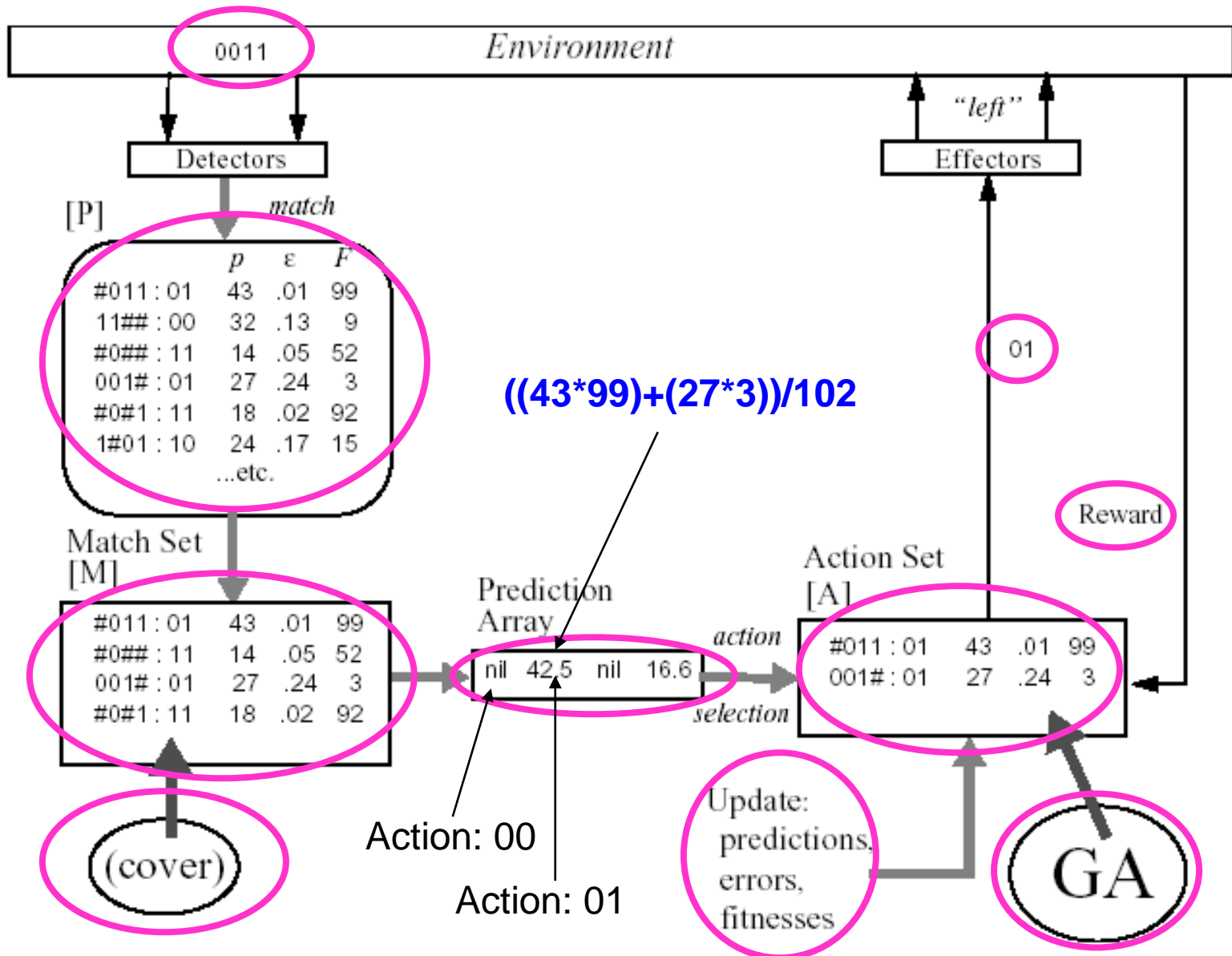
[10001000]



### Action

10001000

	Condition	Action	Payoff	Best classifier
Classifier list	[0010#010]	[nw]	0.7	
	[1##0100#]	[ w]	0.5	
	[000#0101]	[nw]	0.8	
	[0#000#01]	[ew]	0.5	
	[0#100001]	[ n]	0.9	
	[0010#0#0]	[sw]	0.3	
	[#1#01###]	[ w]	0.9	



# 思考和讨论

1. 遗传算法的表示和算子。
2. 小生境等先进遗传算法。
3. 其他自然计算的方法(往往又称为智能计算)。
4. 遗传编程。
5. 学习分类系统(Learning Classifier Systems)。

谢谢！