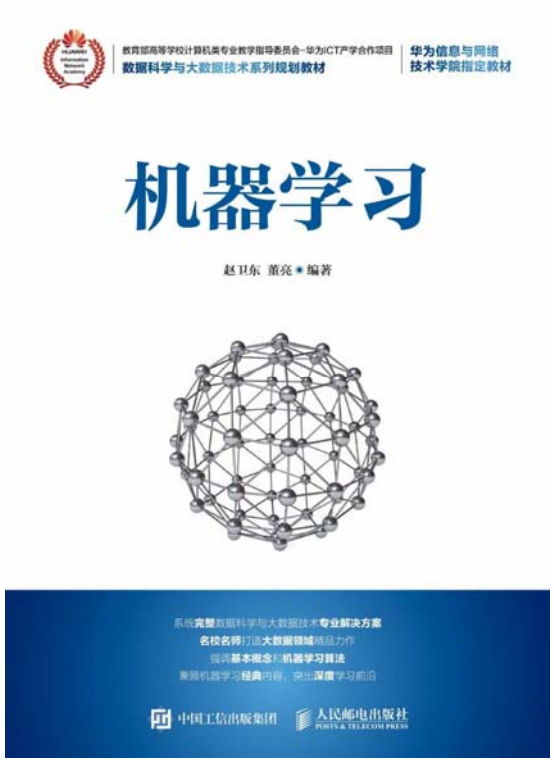

机器学习
进化计算

复旦大学 **赵卫东** 博士

wdzhao@fudan.edu.cn

章节介绍

- 进化计算包括遗传算法、进化策略和基因编程等。进化计算是受进化生物学启发而发展起来的计算模型，其实现过程基于达尔文的生物进化原理，将现实问题转化为基因染色体表示，通过染色体操作，逐步逼近最优解。本章主要是介绍遗传算法的概念、实现方法等基础知识，并结合实例对蚁群算法和蜂群算法做出介绍。

章节结构

- 遗传算法的基础
  - 基因重组与基因突变
  - 遗传算法实现技术
  - 遗传算法应用案例

遗传算法的基础

- Holland 在上世纪60年代提出了了遗传算法遗传算法是进化计算的一个分支，是一种模拟自然界生物进化过程的随机搜索算法。
- 遗传算法首先对问题进行编码，然后随机初始化种群，每个个体对应一个编码。通过适应度函数以及选择函数来进行对个体的淘汰，保留优良个体基因，产生新的子代。
- 遗传算法中有一些基本概念：

  选择算子：根据适应值把个体按比例进行淘汰，从而提高群体的适应值。

  交叉算子：种群中随机选择两个个体，交换染色体部分编码，产生两个新的子个体。

  变异算子：以一个很小的概率随机改变染色体上的某个基因来增加群体的多样性。

基因重组与基因突变

- 交叉运算可以被分为以下五种情况：
  - 单点交叉
  - 两点交叉和多点交叉
  - 均匀交叉
  - 算术交叉
  - 基因突变

单点交叉

- 单点交叉也叫简单交叉，只在个体编码中随机设置一个交叉点，在该点互换两个配对个体的部分染色体。在单点交叉情况下，个体两两配对，其中每一对配对的个体都依照设定的交叉概率在交叉点处相互交换后续的染色体编码串，从而产生两个新的个体。

双亲			后代		
X1	1000	10011110	x1*	1000	11000110
X2	0110	11000110	x2*	0110	10011110

两点交叉和多点交叉

- 两点交叉是指在个体编码中随机设置了两个交叉基因点，然后再进行部分基因片段的交换，交换的部分就是所设定的两个交叉点之间的部分染色体。将单点交叉和两点交叉的概念加以推广，扩展到多点交叉。就是在个体编码串中随机设置多个交叉点，然后进行基因片段的交换。但在实际的遗传算法中，一般不使用多点交叉算子。因为交叉点增多，个体结构被破坏的可能性就更大，个体基因的稳定性就难以保持，从而可能会影响到遗传算法的效率。

均匀交叉

- 均匀交叉可以看成是多点交叉的一种特殊形式，是指两个配对个体的每个基因位上的基因都以相同的概率进行交换，组合成两个新的个体。具体的运算可以设置一串规则来确定新个体每个位置的基因如何继承哪一个父类基因位。

算术交叉

- 算术交叉是指两个个体通过线性组合产生两个新的子代个体。采用这种交叉方式的遗传算法通常采用浮点编码染色体。例如A、B为父体。配对后两个子代为a和b，a=mA+(1-m)B，b=mB+(1-m)A。m可以取一个常数，也可以选择一个由参数决定的变量。

基因突变

- 基因突变是指染色体编码的某一位基因上的改变。基因突变使一个基因变成了它的等位基因，并且通常会引起一些表现型上的变化。
- 二进制编码中，基因突变是指按照一定概率将基因串上的0、1取反。
- 浮点型编码中，基因突变指的是将原来的浮点数增加或者减少一个小随机数。

例：100011000110 → 100011010110

遗传算法的步骤

- 随机产生一组初始个体构成初始种群，并评价每个个体的适应值；
- 判断算法收敛准则是否满足，满足输出搜索结果，否则执行下面的步骤；
- 根据适应值大小以一定方式进行选择操作；
- 按交叉概率p_c执行交叉操作
- 按变异概率p_m执行变异操作
- 返回第二步进行循环

遗传算法实现技术

- 遗传算法实现相关的技术有：
  - 编码
  - 群体的规模
  - 选择策略
  - 适应性度及选择函数
  - 变异算子

编码

- 二进制编码，采用二进制0，1表示染色体的基因信息。
- 格雷码方法，是二进制编码的一种变形，是指连续两个整数所对应的编码值之间只有一个码位是不同的。这一特点解决了二进制编码中的相邻数字的距离较远的问题。
- 浮点编码法，对于一些多维、高精度要求的连续函数优化问题，使用二进制编码会使编码冗长，不利于算法效率的提高。浮点数编码采用浮点数来表示个体的每个基因值，这种编码法需要限制基因值始终在给定区间内。
- 符号编码法，符号编码是指染色体编码中的基因值可能涉及符号集的字符，使用符号编码，便于编码有意义的基因值。这种编码方法需要认真设计交叉、变异等遗传运算，以满足问题的各种约束，从而提高算法的搜索性能。

群体的规模

- 规模较大的群体一般对应的个体多样性较高，可以避免算法陷入局部最优解。但增大群体规模也会增加复杂度，降低算法效率。
- 群体规模一般选在编码长度的一个倍数值，群体的规模是可变的，可以根据算法得到的解的结果进行调整。
- 初始群体的选取采用随机的方法产生，也可以采用其他优化方法或者启发方法选取更加优良的群体。

选择策略

- 选择函数用于选择优胜个体，淘汰不满足条件的个体。有以下三种策略：
- 基于适应值比例的策略，计算个体的相对适应度，用于评价个体的好坏。以相对适应度为选择概率用轮盘赌选择种群。

$$f_i / \sum_{j}^{P} f_j$$

- 基于排名的策略，根据个体适应度在群体中的排名来确定其选择概率，再用第一种方法进行选择，可以避开非线性加速可能产生的早熟现象。
- 基于局部竞争机制的策略，群体中随机选择若干个个体（一般是两个）进行比较，其中适应度最好的个体被确定为生成下一代的父体。

适应性度及选择函数

- 适应度函数用于判定群体中的个体是否满足条件，一般是一个实值函数对个体进行评价，适应度函数值越大，越满足条件。适应度函数的输出值需要是能够进行比较的非负结果。适应度评价是选择操作的依据，适应度函数设计直接影响到遗传算法的性能。
- 选择函数用选择运算来实现对群体中的个体进行优胜劣汰，适应度高的个体被遗传到下一代种群中的概率就大。选择算子是一种选择方法，从父代中选择满足条件的个体遗传到下一代，常用的选择方法有轮盘赌选择法、随机遍历抽样法、局部选择法、最佳个体保存方法、排序选择法、联赛选择方法。

变异算子

- 变异算子能使个体按一定概率发生变异，产生新的遗传基因，有助于增加种群多样性，是提高全局最优搜索能力的有效步骤，也是保持群体差异，防止过早出现收敛的重要手段。遗传算法中交叉和变异的操作使算法具备兼顾全局和局部的均衡搜索能力。
- 群体的替换率与交叉概率和变异概率相关，替换率较低的情况下每代种群更新较慢，使得搜索范围扩展较慢，但能够较大程度保留现有基因。过高的替换率可能会过滤掉当前的最优解，可以采用保留策略，使上一代的当前最优解能够流传到下一代。

遗传算法的优越性

- 能够普遍适用于数值求解问题，对目标函数要求低，总能以较大的概率找到全局最优解。
- 在求解很多组合优化问题时，不需要对问题有非常深入的了解，在确定问题的决策变量编码后，其计算过程是比较简单的，且可较快的得到一个满意解。
- 与其他启发式算法有较好的兼容性，容易结合形成性能更优的问题求解方法。

遗传算法算例（1）

使用遗传算法求出函数最大值: $f(x)=x^2,\quad x\in[0,31]$

(1) 设定种群规模, 编码染色体，将种群规模设定为4；用5位二进制数编码染色体；取下列个体组成初始种群S_1:

s_1= 13 (01101), s_2= 24 (11000)

s_3= 8 (01000), s_4= 19 (10011)

(2) 定义适应度函数：$f(x)=x^2$

(3) 计算各代种群中个体的适应度, 并对其染色体进行遗传操作,直到适应度最高的个体：31（11111）出现为止。

首先计算种群S_1中各个体s_1= 13(01101)，s_2= 24(11000)，s_3= 8(01000)， s_4= 19(10011)的适应度$f(s_i)$。

得到$f(s_1)=f(13)=13^2=169$，$f(s_2)=f(24)=24^2=576$，$f(s_3)=f(8)=8^2=64$，$f(s_4)=f(19)=19^2=361$

选择概率的计算公式为 $P(x_i)=\dfrac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$

得到$P(s_1)=P(13)=0.14$，$P(s_2)=P(24)=0.49$，$P(s_3)=P(8)=0.06$，$P(s_4)=P(19)=0.31$

赌轮选择法

遗传算法算例（2）

染色体	适应度	选择概率	积累概率	选中次数
s_1=01101	169	0.14	0.14	1
s_2=11000	576	0.49	0.63	2
s_3=01000	64	0.06	0.69	0
s_4=10011	361	0.31	1.00	1

复制得到群体：

s_1'=11000（24），s_2'=01101（13）

s_3'=11000（24），s_4'=10011（19）

交叉

设交叉率p_c=100%，即S_1中的全体染色体都参加交叉运算。

设s_1'与s_2'配对，s_3'与s_4'配对，分别交换后两位基因，得新染色体：

s_1''=11001（25），s_2''=01100（12）

s_3''=11011（27），s_4''=10000（16）

变异

设变异率p_m=0.001。群体S_1中共有5×4×0.001=0.02位基因可以变异。0.02位显然不足1位，所以本轮遗传操作不做变异。

第二代种群S_2：

s_1=11001（25），s_2=01100（12）

s_3=11011（27），s_4=10000（16）

染色体	适应度	选择概率	积累概率	估计的选中次数
s_1=11001	625	0.36	0.36	1
s_2=01100	144	0.08	0.44	0
s_3=11011	729	0.41	0.85	2
s_4=10000	256	0.15	1.00	1

……

第三代种群$S3$：

s_1=11100（28），s_2=01001（9）

s_3=11000（24），s_4=10011（19）

第四代种群S_4：

s_1=11111（31），s_2=11100（28）

s_3=11000（24），s_4=10000（16）

遗传算法应用案例

- 应用遗传算法解决的实际问题是旅行商问题。旅行商问题可以用于评价不同的遗传操作以及选择机制的性能。这是因为：
  - 旅行商问题是一个易于描述却难以处理的问题，在可计算理论中有重要的理论价值；
  - 旅行商问题是诸多领域内出现的多种复杂问题的集中概括和简化形式，有一定的实际应用价值；
- 这个问题的求解可以划分为三个步骤：
  - 编码
  - 适应度函数
  - 基于遗传算法求解

编码

- 路径编码：一串数字代表一条路径，其中每个数字代表一个城市
- 顺序编码：将所有城市按顺序构成一个顺序表，对于一个旅程，可以依据行程经过的顺序处理每个城市，每个城市在顺序表中的顺序就是一个遗传因子，每次处理完一个城市，从顺序表中去掉该城市。处理完所有城市后，将每个城市的遗传因子表示连接起来，即成为这个旅程的基因编码。
- 布尔矩阵编码：布尔矩阵编码采用非向量表示方法，一个旅程定义为一个优先权布尔矩阵M，当且仅当城市i排在城市j之前时矩阵元素m_{ij} = 1。

适应度函数

- 适应度函数为回路长度的倒数

基于遗传算法求解

- 如两个父个体 A:(1 2 3 4 5 6 7 8 9)和B:(4 1 2 8 7 6 9 3 5)，对两个父代矩阵中位进行交叉运算（A中的4 1 3与B中的5 3 3交换），得一新矩阵，产生无矛盾的部分序：

A 1 1 2 1 4 1 3 1 1 → 1 1 2 1 5 3 3 2 1 A’

B 5 1 5 5 5 3 3 2 1 → 5 1 5 5 4 1 3 1 1 B’

- 由交叉结果得：城市1优先于2,3,5,6,7,8,9；城市2优先于3,5,6,7,8,9；城市3优先于5；城市4优先于5,6,7,8,9；城市6,7,8优先于9。