

1. TOPSIS
2. 熵值法
3. PROMETHEE
4. VAR向量自回归
5. 遗传算法

1. TOPSIS

1. 适用范围：综合多个评价指标，选出最优/最劣对象
2. 主要思想：综合考虑当前对象与最优对象、最劣对象之间的距离
3. 使用过程：

Algorithm 1: TOPSIS 算法流程

Input 原始数据集 $X = \{x_1, x_2, \dots, x_n\}$
各指标权重 $w = (w_1, w_2, \dots, w_m)$

Process

1. 对原始数据集中的指标属性同向化 X'
2. 构造向量归一化后的标准化矩阵 $Z = \{z_1, z_2, \dots, z_n\}$
3. **for** Z 的每一列 Z_i **do**
4. 最劣方案 Z^- 的第 i 维度 $\leftarrow Z_i$ 元素最小值
5. 最优方案 Z^+ 的第 i 维度 $\leftarrow Z_i$ 元素最大值
6. **end for**
7. **for** $z_i \in Z$ **do**
8. z_i 与最优方案的接近程度 $D_i^+ \leftarrow$ 式(7.1)
9. z_i 与最劣方案的接近程度 $D_i^- \leftarrow$ 式(7.2)
10. z_i 与最优方案的贴近程度 $C_i \leftarrow$ 式(8)
11. **end for**
12. 根据 C_i 大小进行排序

Output 各数据样本 TOPSIS 评价结果

知乎 @Suranyi

2. 熵值法

1. 适用范围：从原始数据集（不同行表示不同对象，不同列表示不同属性）推算出各属性的权重
2. 主要思想：某属性的离散程度越大，则越混乱，熵值越大，权重越大
3. 适用过程：
 - 归一化原始数据集（不用进行同向化处理）：

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}}$$

- 计算各属性的熵值：

$$e_j = -k \sum_{i=1}^n p_{ij} \ln p_{ij}, \quad (j = 1, 2, \dots, m)$$

其中 k 与样本数量有关，常取 $k = 1/\ln n$ ，此外，补充定义：若 $p_{ij} = 0$ ，则令 $p_{ij} \ln p_{ij} = 0$

- 计算各属性的权重系数：

$$h_j = \frac{1 - e_j}{\sum_{k=1}^m (1 - e_k)}, \quad (j = 1, 2, \dots, m)$$

熵权系数 h_j 越大，则该指标代表的信息量越大，表示其对综合评价的作用越大。

3. PROMETHEE

1. 适用范围：综合多个评价指标，选出最优/最劣对象
2. 主要思想：当前对象在当前指标超过一个对象，则加上一个当前指标的权重
3. 使用过程（有 n 个对象， t 个属性）：

- 定义偏好函数：一般直接取差即可，大于0置1，小于0置0即可

$$P_k(a_i, a_j) = \phi(a_{ik} - a_{jk}) = \begin{cases} 1, & a_{ik} - a_{jk} \geq 0 \\ 0, & a_{ik} - a_{jk} < 0 \end{cases}$$

- 定义偏好优先指数函数：

$$\Pi(a_i, a_j) = \sum_{k=1}^t \omega_k P_k(a_i, a_j)$$

- 定义对象 a_i 的偏好优序级别的正负方向（流入流出）：

$$\Phi^+(a_i) = \sum_{j=1}^n \Pi(a_i, a_j) \quad (2)$$

$$\Phi^-(a_i) = \sum_{j=1}^n \Pi(a_j, a_i) \quad (3)$$

式中： $\Phi^+(a_i)$ —— 方案 a_i 超过另外方案的程度， $\Phi^-(a_i)$ —— 方案 a_i 被超过的程度。

4. VAR向量自回归

传统的时间序列模型只考虑了纵向比较： $y_t = \beta_1 \cdot y_{t-1} + \beta_2 \cdot y_{t-2} + \dots$

传统的回归模型只考虑了横向比较： $y_t = \beta_1 x_t + \beta_2 z_t + \dots$

向量自回归既考虑了纵向比较，又考虑了横向比较：

$$y_t = \beta_1 \cdot y_{t-1} + \alpha_1 \cdot x_{t-1} + \beta_2 \cdot y_{t-2} + \alpha_2 \cdot x_{t-2} + \dots$$

5. 遗传算法

1. 适用范围：求解基于带约束的单目标优化问题（多目标优化也可以转换成单目标优化）
2. 主要思想：模拟自然过程
3. 算法流程（确定编码方式，适应函数，群体规模，变异概率和终止条件）：

$$\begin{aligned} \max \quad & f(x_1, x_2) = x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1 \in \{1, 2, 3, 4, 5, 6, 7\} \\ & x_2 \in \{1, 2, 3, 4, 5, 6, 7\} \end{aligned}$$

- 1) 确定个体编码方式：每个解对应一个编码，此题将十进制数编码成无符号二进制数，例如基因型 $X=101\ 110$ ，对应解为 $X=[5, 6]=[x_1, x_2]$
- 2) 确定初始群体规模：
 本例中，群体规模的大小取为4，即群体由4个个体组成，每个个体可通过随机方法产生。
 如：011101, 101011, 011100, 111001
- 3) 适应度计算：将目标函数的值当作个体适应度即可
- 4) 选择：采取与适应度成正比的概率确定繁衍个体

个体编号	初始群体 $p(0)$	x_1	x_2	适值	占总数的百分比	选择次数	选择结果
1	011101	3	5	34	0.24	1	011101
2	101011	5	3	34	0.24	1	111001
3	011100	3	4	25	0.17	0	101011
4	111001	7	1	50	0.35	2	111001
总和				143	1		

- 5) 交叉：

个体编号	选择结果	配对情况	交叉点位置	交叉结果
1	01 1101	1-2 3-4	1-2: 2 3-4: 4	011001
2	11 1001			111101
3	1010 11			101001
4	1110 01			111011

可以看出，其中新产生的个体“111101”、“111011”的适应度较原来两个个体的适应度都要高。

- 6) 变异:

个体编号	交叉结果	变异点	变异结果	子代群体p(1)
1	011001	4	011101	011101
2	111101	5	111111	111111
3	101001	2	111001	111001
4	111011	6	111010	111010

至此完成了一轮循环，可以看出经过一代进化，种群适应度的最大值和平均值都提升了

4. 实现：遗传算法、模拟退火、蚁群算法等可以直接调用[scikit-opt](#)库