

导航

博客园  
首 页  
新随笔  
联 系  
订 阅  
管 理

< 2017年12月 >

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

公告

昵称: kool\_zhang  
园龄: 5年5个月  
粉丝: 2  
关注: 1  
+加关注

搜索

找我看

谷歌搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

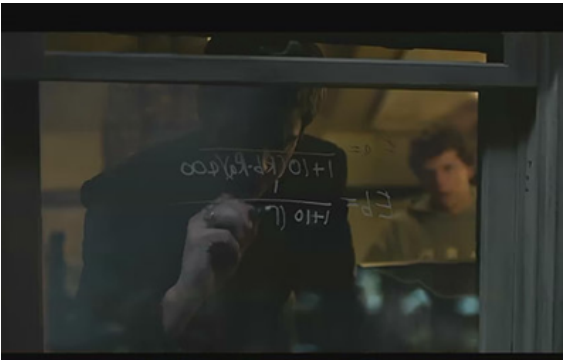
我的标签

Matplotlib(2)  
Python(2)  
数据挖掘 数据分析(2)  
读书笔记(1)  
可视化(1)  
随笔分类(4)  
essaies  
编程学习(4)  
读书随笔  
商业分析  
数据挖掘  
随笔档案(7)  
2017年4月 (1)  
2017年3月 (1)  
2015年10月 (1)  
2015年6月 (1)  
2015年4月 (1)  
2015年3月 (2)  
最新评论  
1. Re:Python  
html.parser库学习小结  
@.net爱上python好的，感谢关注。...  
--kool\_zhang  
2. Re:Python  
html.parser库学习小结  
可以关注一下pyquery  
--.net爱上python  
阅读排行榜  
1. Numpy 学习之路 (1)  
——数组的创建(3421)  
2. Python html.parser库学习小结(2917)

从Elo Rating System谈到层次分析法

1. Elo Rating System

Elo Rating System对于很多人来说比较陌生，根据wikipedia上的解释：Elo评分系统是一种用于计算对抗比赛（例如象棋对弈）中对手双方技能水平的方法，它由Arpad Elo创建。[\(The Elo rating system is a method for calculating the relative skill levels of players in competitor-versus-competitor games such as chess. It is named after its creator Arpad Elo, a Hungarian-born American physics professor.\)](#)。大家如果看过电影《社交网络》（The Social Network）的话，一定记得爱德华在玻璃窗上写给扎克伯格的，用以评价学校女生的算法——那就是Elo评分系统。Elo目前在许多国际比赛中用以进行选手（或者团队）的世界排名（电影中用作对女生进行评分，可以理解两个女性在比拼谁更漂亮，尽管她们不一定愿意参加这个“比赛”）。



(1) Elo Rating System的基本概念

对于一次有结果的双人（也可以是两个团队）对抗性比赛，对某方而言，结果无非三个：赢、平局、输。将比赛结果量化成积分的话，需要为每个结果赋予对应的分数值：赢+1，平+0.5，输+0。如果直接按照这种比赛结果进行积分的累计计算，显然是不公平的。如果某人参加比赛的次数较多，或者本身的积分较高，如果得分较低的选手赢得了得分较高的选手，那么他获取的积分应该更高才对。这就是EOL Rating System提出的初衷。

Elo Rating System以两个选手（rated player）在存量比分上的差异作为比赛结果的预测因子（Predictor），它是一个可以实现自我更正的系统（self-correcting）。Elo Rating System基于以下几个方面的假设：

- H1：选手的最终比分成绩是一个服从正太分布的随机变量（Normal Distribution Random Variable）
- H2：一位选手比赛成绩的均值是他真正实力（True Skill）的表现；
- H3：选手的实力评价完全由比赛的结果来判定；在象棋比赛中，我们不能够因为其中一参赛者走了一步很精彩的而认为这个选手的实力高，因为这是完全不可量化的，我们只能够根据比赛的结果来判定。换句话说，选手之间实力的差异程度可以通过比分的差异来判定。

对于两位选手A和B，假设他们的积分分别是Ra和Rb（电影中给每个女孩的基础得分是1400）。那么A和B进行比赛的得分期望是：

$$E_A = \frac{1}{1 + 10^{(Ra - Rb) / 400}}$$
$$E_B = \frac{1}{1 + 10^{(Rb - Ra) / 400}}$$

3. Python可视化学习 (2) : Matplotlib快速绘图基础(507)
4. Python可视化学习 (1) : Matplotlib的配置(459)
5. 从Elo Rating System谈到层次分析法(315)

评论排行榜

1. Python html.parser库学习小结(2)

推荐排行榜

1. Python html.parser库学习小结(2)
2. Python可视化学习 (1) : Matplotlib的配置(1)

其中有： $E_A + E_B = 1$

，因此，对于上面的两条公式，我们也可以理解为是对双方实力的比例进行概率化，分界点是0.5，上述的完整公式也可以在连续下表述成：

$$E_A = \frac{1}{2} + \int_0^{\text{Diff}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx$$

(PS：越看这条公式越是觉得它漂亮，充分体现了以前研究生一位老师说过的话：跟跑得慢的人一起跑，跑得快也是慢，跟跑得快的人一起跑，跑得慢也是快。)

在实际操作当中，由于选手的期望实力是未知的，我们通常采用选手的现有积分作计算其期望积分。如果一个选手在一次比赛中超越了它的期望成绩，那么Elo系统就认为这个选手的实力被低估了，应该往上调整，同理，如果低于期望值，那么Elo系统就会降低这位选手的评分。假设，对于一位选手A，他的期望低分是 $E_a$ ，而实际得分是 $S_a$ ，则根据Elo的线性调整建议，两位选手得分的更新策略是：

$$R'_A = R_A + K(S_A - E_A)$$

$$R'_B = R_B + K(S_B - E_B)$$

其中，K是一个调整步长，可根据选手的分级水平进行设定。Elo在国际象棋中，建议Master级的K=16，一般级别的K=32。举个例子，现在两个选手，A和B，他们的积分分别是 $R_a=1600$ ， $R_b=1530$ ，他们进行比赛，结果B赢了，则比赛后，A和B的积分计算步骤如下（假设K=32，win+1，lose0 draw 0.5）：

```

elo.py - C:\Users\Haohong.Zhang\Desktop\elo.py
File Edit Format Run Options Windows Help

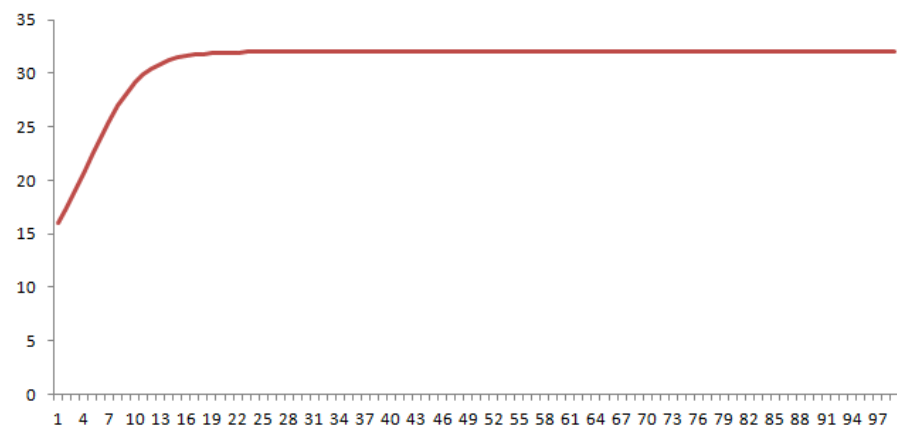
def eloRating(r_a,r_b, winner):
    result={}
    K=32
    #calculate the expected value
    E_a=1/(1+10**((r_a-r_b)/400))
    E_b=1/(1+10**((r_b-r_a)/400))
    if winner=='a':
        result['a']=r_a+32*(1-E_a)
        result['b']=r_b+32*(0-E_b)
    if winner=='b':
        result['a']=r_a+32*(0-E_a)
        result['b']=r_b+32*(1-E_b)
    return result

if __name__=="__main__":
    re=eloRating(1600,1530,'a')
    print(str(re))

Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
Traceback (most recent call last):
  File "C:\Users\Haohong.Zhang\Desktop\elo.py", line 18, in <module>
    print(str(re))
NameError: name 'prin' is not defined
>>>
>>>
({ 'a': 1619.1806974946962, 'b': 1510.8193025053038})
>>>

```

结果是： $R_a=1619.18$ ， $R_b=1510.82$ 。由于这次A和B的得分是比较接近的，所以输赢的分配是较为均等，我们如果将A的本来领先优势扩大的话，如果跟B连续进行比赛，那么他赢得比赛得到的加分趋势如下——



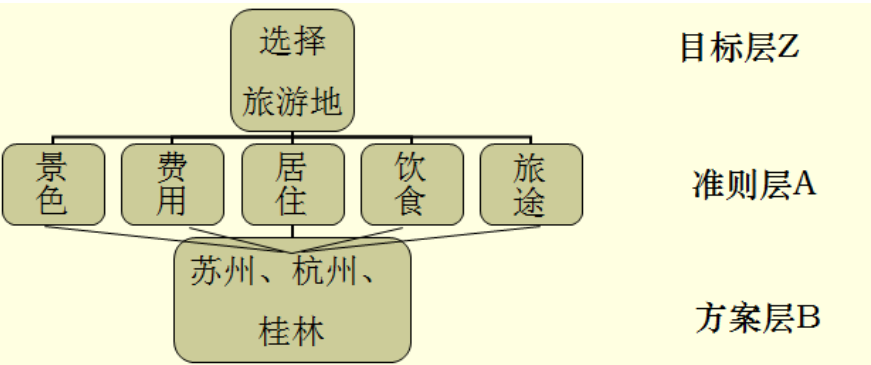
2. 层次分析法 ( Analytic Hierarchy Process , AHP )

人在决策时，通常都需要考虑多个因素。层次分析法 ( Analytic Hierarchy Process, AHP),这是一种定性和定量相结合的、系统化的、层次化的分析方法，他根据决策因素的影响结果进行决策，与人们对某一复杂决策问题的思维、判断过程大体一致。

层次分析法的基本步骤：

( 1 ) 建立层次结构模型

—— 一般需要三个层，最上面为目标层，最下面为方案层，中间为准则层/指标层



( 旅游决策案例，下面会用这个图作为举例 )

有时候，某一个方案的决策并不是需要考虑所有的准则，诸如此类，如果上层的每个因素都支配着下一层的所有因素，或被下一层所有因素影响，称为完全层次结构，否则称为不完全层次结构。

( 2 ) 构建 成对比较矩阵

构建成对比较矩阵的目的在于将某一层因素对上一层某一个因素相对影响程度进行量化处理。例如在上述案例中，准则层A对于目标层Z的影响就需要构建准则层对目标层的成对对比矩阵——构建成对比较矩阵的目的在于确定下层所有元素对上层的归一化权重w。

设某一层有n个元素，用A表示，对比该层所有元素的相对重要性程度，得到：

$$A = (a_{ij})_{n \times n} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

A就被称为成对比较矩阵。其中a<sub>ij</sub>的值是两两比较的结果，即该层某个元素对应上一层目标因素的相对重要性，其中，满足：

$$a_{ij} = \frac{1}{a_{ji}}$$

( 即该矩阵为一个正互反阵，满足a>0 )

矩阵中的值是两两比较的结果，比较时取的尺度通常是1-9，对比的标准如下：

尺度	含义
1	第 <i>i</i> 个因素与第 <i>j</i> 个因素的影响相同
3	第 <i>i</i> 个因素比第 <i>j</i> 个因素的影响稍强
5	第 <i>i</i> 个因素比第 <i>j</i> 个因素的影响强
7	第 <i>i</i> 个因素比第 <i>j</i> 个因素的影响明显强
9	第 <i>i</i> 个因素比第 <i>j</i> 个因素的影响绝对地强

2,4,6,8表示第*i*个因素相对于第*j*个因素的影响介于上述两个相邻等级之间。旅游地点选择问题的两两对比结果如下：

Z	A1	A2	A3	A4	A5
A1	1	1/2	4	3	3
A2	2	1	7	5	5
A3	1/4	1/7	1	1/2	1/3
A4	1/3	1/5	2	1	1
A5	1/3	1/5	3	1	1

成对对比矩阵：

$$A = \begin{bmatrix} 1 & 1/2 & 4 & 3 & 3 \\ 2 & 1 & 7 & 5 & 5 \\ 1/4 & 1/7 & 1 & 1/2 & 1/3 \\ 1/3 & 1/5 & 2 & 1 & 1 \\ 1/3 & 1/5 & 3 & 1 & 1 \end{bmatrix}$$

根据旅游问题的层次结构模型，该决策问题共有6个成对对比矩阵：

- 1个5阶的：准则层对目标层；
- 5个3阶的：决策层对准则层；

上述得出的成对比较矩阵只是一个层对上层某一因素相对重要性在内部结果，**下面进行的步骤关键是将这个对内的矩阵转化成对外的权重**

- 一致阵与非一致阵

所谓的一致阵，是指对于上述的成对比较矩阵中，满足： $a_{ik} \bullet a_{kj} = a_{ij}$ ，则称这个成对比较矩阵为一致阵，一致阵有以下的性质：

- 1 ).  $a_{ij}=1/a_{ji}$ ；
- 2 ). A的转置也是一致阵；
- 3 ). A的各行成比例，则 $\text{rank}(A)=1$

- 4 ). A的最大特征根（值）为n，其余n-1个特征根为0
- 5 ). A的任一列（行）都是对应于特征根n的特征向量

**定理：**n阶互反矩阵A的最大特征值 $\lambda \geq n$ ，当且仅当 $\lambda = n$ 时，互反矩阵A才是一致性矩阵。

在将成对比较矩阵转化为权重之前，需要做一致性检验：

- 如果一个成对比较矩阵式一致性阵的话，则用特征根n对应的归一化特征向量W={w1,w2....wn}，其中wi表示下层第i个因素对上层某个因素影响程度的权值。
- 成对比较矩阵是非一致性矩阵的时候，那么就采用最大特征值对应的特征向量的归一化结果来作为权重向量W,则：  
这样确定权重的方法称为特征根法
- 从特征值的变化来看，当比n大得多的时候，矩阵A的一致性程度就会越来越低（对于成对比较矩阵而言，就是各个要素之间相互评比的不一致程度较高），此时用使用最大特征根值对应的特征向量作为被比较因素对上层某因素影响程度的权重向量，其一致性程度越大引起的误判就越大。因而，我们需要用 $\lambda$  为一致性矩阵时的，对比 $\lambda - n$  数值的大小来衡量A的不一致程度，定义一致性指标如下：

$$CI = \frac{\lambda - n}{n - 1}$$

（其中n为对角线元素的和，也称为A的特征根之和）上面定义的CI（一致性指标）是一个绝对数指标，我们有一个客观标准来判定CI究竟是否可以接受。采用的方法是随机一致性指标RI来进行比较，通常随机一致性指标的值如下（随机构造500个成对比较矩阵，计算其平均CI）：

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51

一般，当一致性比率

$$CR = \frac{CI}{RI} < 0.1$$

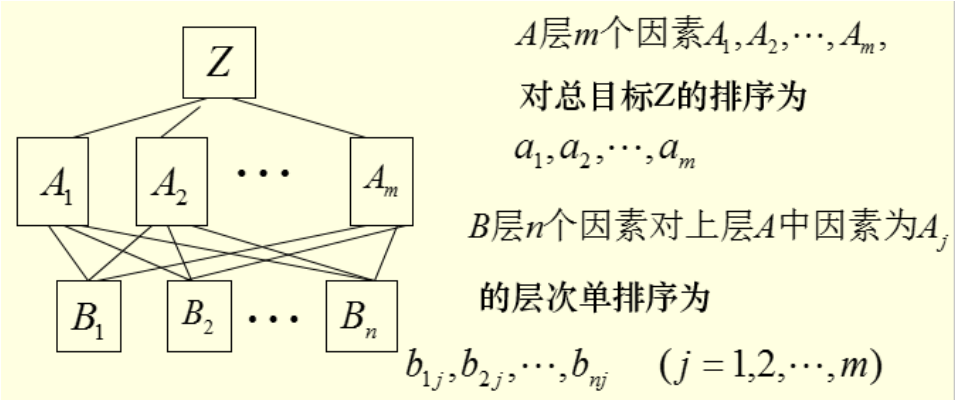
则认为成对比较矩阵的不一致性程度在容许范围之内，可以采用其归一化特征向量作为权重向量，否则要重新构造成对比较矩阵，对原始的矩阵加以调整。

**（3）层次总排序及一致性检验**

确定某一层所有因素对于总目标相对重要性的排序权重过程，称为层次总排序。

其实也就是确定每个决策方案对于总目标的影响程度，当已经完成某个下层对上层单一因素的影响后，应对总体某层的作用作总体评判。

从最高层到最低层。设：



计算B层的层次总排序：实质就是从纵向计算一个决策方案的总目标权重，然后衡量那个方案对总目标的权重结果较大。

A \ B	$A_1, A_2, \cdots, A_m$			B层的层次总排序
B	$a_1, a_2, \cdots, a_m$			
$B_1$	$b_{11}$	$b_{12}$	$b_{1m}$	$\sum_{j=1}^m a_j b_{1j} = b_1$
$B_2$	$b_{21}$	$b_{22}$	$b_{2m}$	$\sum_{j=1}^m a_j b_{2j} = b_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
$B_n$	$b_{n1}$	$b_{n2}$	$b_{nm}$	$\sum_{j=1}^m a_j b_{nj} = b_n$

横向：就是一个决策方案在所有准则上的权重加权求和

纵向：下层因素B对上层某一个因素A1的归一化权重

层次总排序的一致性检验：设B层 $B_1, B_2, \dots, B_n$ 对上层（A层）因素 $A_j$ 的一致性指标为 $CI_j$ ，则层次总排序的一致性比率为：

$$CR = \frac{a_1 CI_1 + a_2 CI_2 + \dots + a_n CI_n}{a_1 RI_1 + a_2 RI_2 + \dots + a_n RI_n} = \frac{\sum_{i=1}^n a_i CI_i}{\sum_{i=1}^n a_i RI_i}$$

当 $CR < 0.1$ 时，我们认为层次总排序通过一致性检验。

到此，根据最下层方案层的层次总排序结果做出最后决策。

层次分析法的优点和局限性：

优点：

- 1. 系统性：将研究的对象作为一个系统，按照分解、比较、综合判断的方法进行决策，符合人的基本思维习惯。
- 2. 实用性：将定性和定量方法结合起来，这种方法也让决策者和决策者能够相互沟通；
- 3. 简洁性：整个方法的逻辑易于理解，计算也十分方便。

不足：

- 1. 只能在备选方案中选择一个方案，无法拟合出最优方案
- 2. 从建立层次模型到成对比较矩阵，整个过程人为因素影响很大，这就使得结果难以让所有决



策者接受

- 3. 该方法中的比较、判断以及结果的计算过程较为粗糙，不适合精度较高的问题。
- 4.

3. 总结

将这两个方法放在一起谈，主要原因是两个方法都涉及到对一个因素进行对比的情况：在Elo系统中，我们是根据每个胜利者的期望进行对比计算，而在AHP中，则是通过定性的方法对比同一个层内各因素的相对重要性，各有优缺点。而从AHP中，学到的一个很重要的思维就是将一个系统内部的关系转化为对外的权重表现时，可以采用特征根的方法来解决。看来理解线性代数的物理含义和实际含义非常重要。

标签: [数据挖掘](#) [数据分析](#)

好文要顶

关注我

收藏该文

kool\_zhang

关注 - 1

粉丝 - 2

+加关注

0

0

« 上一篇: [尽信“数”不如无“数”——《决战大数据》读书笔记](#)

» 下一篇: [Python html.parser库学习小结](#)

posted on 2015-04-01 10:13 kool\_zhang 阅读(314) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

- 【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【促销】腾讯云技术升级10大核心产品年终让利
- 【推荐】高性能云服务器2折起，0.73元/日节省80%运维成本
- 【新闻】H3 BPM体验平台全面上线

ar

ActiveReports 报表控件  
V12 全新发布!

全面满足  
.NET 报表开发需求

立即了解

- 最新IT新闻：
- 外媒：阿里巴巴与俄罗斯最大银行放弃合作计划
  - 腾讯云和蔚来合作，要将汽车打造成移动会客厅
  - ofo投资人徐小平、张颖：希望ofo和摩拜合并
  - 2700平米全球最大星巴克店落户上海，背后释放出了这些信号
  - Twitter新功能方便用户同时发连串推文 绕开字数限制
- » 更多新闻...

告别高昂运维费用 云计算全面助力  
40+款核心产品免费半年 再+8000津贴任意采购

立即申请

- 最新知识库文章：
- 以操作系统的角度述说线程与进程
  - 软件测试转型之路
  - 门内门外看招聘
  - 大道至简，职场上做人做事做管理
  - 关于编程，你的练习是不是有效的？
- » 更多知识库文章...

Powered by:  
[博客园](#)  
Copyright © kool\_zhang