《写给程序员的数据挖掘指南》notes

中文版链接：https://datamininggguide.books.yourtion.com/
源码链接： https://github.com/yourtion/DataminingGuideBook-Codes
一本很好的数据挖掘入门书和python入门书，如果代码写得差数学也不好也看得下去。

Part I 协同过滤
协同过滤是什么？
比如我想推荐一本书给你，我会在网站上搜索与你兴趣类似的其他用户，看看这个用户喜欢什么书然后把它们推荐给你。
如何寻找相似用户？
根据用户A和B有共同评价的书的评分计算距离：

曼哈顿距离和欧氏距离可以一般化为如下的明氏距离（Minkowski Distance）：

$$d(x,y)=(\sum_{k=1}^{n}|x_k-y_k|^r)^{\frac{1}{r}}$$

其中：

- r=1 时，上述公式计算的就是曼哈顿距离。
- r=2 时，上述公式计算的就是欧氏距离。
- r=∞时，上述公式计算的是上确界距离（Supermum Distance）。

Minkowski Distance

如何解决用户的评级差异问题？
皮尔森相关系数

$$r=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

Pearson Correlation Coefficient

但A和B有共同评价的书可能很少（数据稀疏），余弦相似度可以忽略这样的0-0匹配：

$$\cos(x,y)=\frac{x\cdot y}{||x||\times||y||}$$

余弦相似度

仅依赖于最相近的用户推荐有时候是片面的，基于多个相似用户的推荐可能更好。比如K近邻算法。

隐式评级
除了用户给出评级（显式评级）之外，观察用户行为也是获得结果的方法：比如跟踪用户在纽约时报在线的点击轨迹。
到目前为止都是基于用户的过滤，但有两个问题：1 、扩展性（用户数量会变多）；2、稀疏性。
因此最好采用基于物品的过滤，计算出最相近的两件物品。
同样用余弦相似度计算两个物品的相似度（为抵消分数夸大的情况，我们从每个评级结果中减掉平均评分）：

$$s(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_u)^2}}$$

调整后的余弦相似度

Slope one 算法
（1）计算所有物品对的偏差

物品 i 到物品 j 的平均偏差为：

$$dev_{i,j}=\sum_{u\in S_{i,j}(X)}\frac{u_i-u_j}{card(S_{i,j}(X))}$$

（2）利用加权偏差进行预测

$$P^{wS1}(u)_j=\frac{\sum_{i\in S(u)-\{j\}}(dev_{j,i}+u_i)c_{j,i}}{\sum_{i\in S(u)-\{j\}}c_{j,i}}$$

其中，

$$c_{j,i}=card(S_{j,i}(\chi))$$

$P^{wS1}(u)j$ 指的是利用加权 Slope One 算法给出的用户 u 对物品 j 的预测值。

Part II 内容过滤和分类
内容过滤不同于协同过滤，它是基于物品属性的过滤。
选定某些特征，给出物品各项特征得分，其他和第一部分计算相似。
Part III 算法评估和KNN
（1）算法评估
n折交叉验证（留一法）：测试集和训练集的划分可能会导致准确率无法反应实际分类器的精确程度。若把数据集分为N份，使用其中n-1份进行训练，剩下一份测试，重复10次做平均。（计算机开销比较大，适合小数据集）

混淆矩阵：可视化输出结果

actual MPG \ predicted MPG	10	15	20	25	30	35	40	45
10	3	10	0	0	0	0	0	0
15	3	68	14	1	0	0	0	0
20	0	14	66	9	5	1	1	0
25	0	1	14	35	21	6	1	1
30	0	1	3	17	21	14	5	2
35	0	0	2	8	9	14	4	1
40	0	0	1	0	5	5	0	0
45	0	0	0	2	1	1	0	2

53.316% accurate
total of 392 instances

混淆矩阵举例

Kappa统计量

$$\kappa = \frac{P(c) - P(r)}{1 - P(r)}$$

kappa统计量

一个最常被人引用的 Kappa 统计量区间对照解释

<0: 比随机方法的性能还差（less than chance performance）

0.01-0.20: 轻微一致（slightly good）

0.21-0.40: 一般一致（fair performance）

0.41-0.60: 中度一致（moderate performance）

0.61-0.80: 高度一致（substantially good performance）

0.81-1.00: 接近完美（near perfect performance）

① Landis, JR, Koch, GG. 1977. *The measurement of observeragree-ment for categorical data*. Biometrics 33:159-74

评估对照表

（2）kNN（选取最近的k个点投票）
Brill：更多训练数据比算法改进带来的准确率提高更多。
Part IV 概率和朴素贝叶斯
（1）朴素贝叶斯简介
近邻方法：lazy learner （每次都要遍历整个训练数据）
贝叶斯方法：eager learner（给定数据集立刻进行数据分析和模型构建）
P(h)即某个假设h为真的概率称为h的先验概率。
P(h|d)即观察到数据d之后h的概率称为后验概率。

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

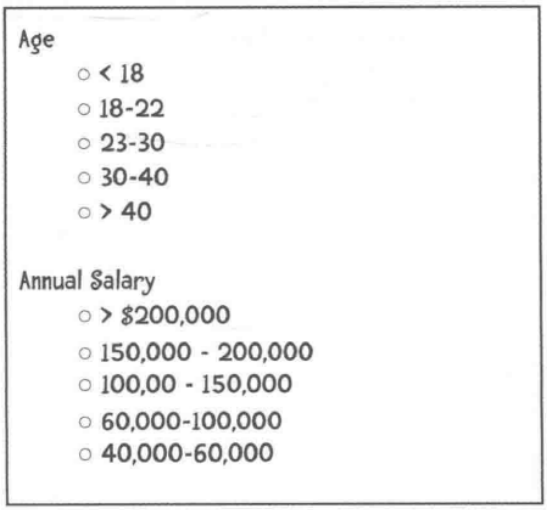
贝叶斯定理

朴素贝叶斯（假设相互独立）
P(A|B,C,D)=P(A)P(B|A)P(C|A)P(D|A)
概率估计（防止nc为0）

$$P(x \mid y) = \frac{n_c + mp}{n + m}$$

概率估计

数据类型
贝叶斯方法用categorical data，区间内的值计数问题可以用如下解决：
方法一：构建类别



构建类别

方法二：高斯分布

$$P(x_i \mid y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{\frac{-(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

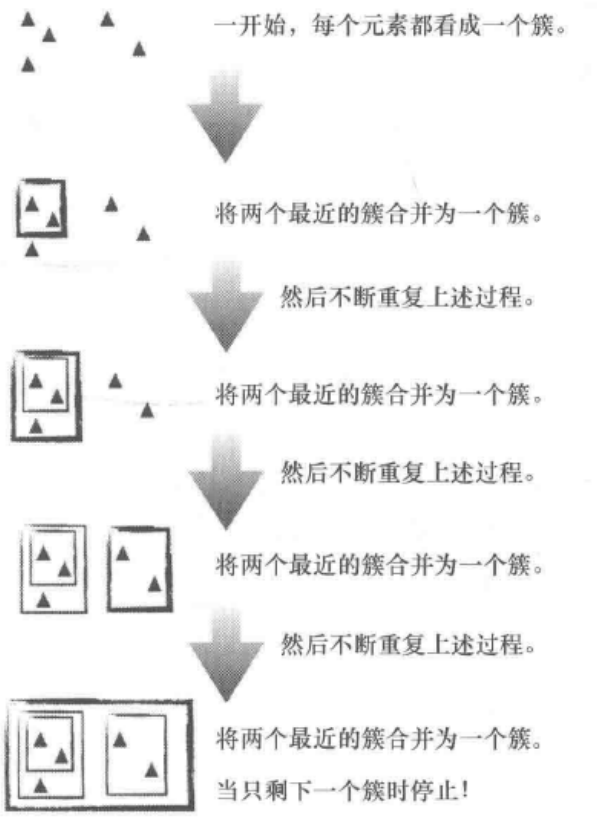
高斯分布

（2）朴素贝叶斯及文本-非结构化文本分类

Part V 聚类--群组发现

kmeans聚类

层次聚类



层次聚类步骤

其他：

- heapq模块 http://blog.csdn.net/marksinoberg/article/details/52734332
- *assert
- assert语句：用以检查某一条件是否为True，若该条件为False则会给出一个AssertionError。
- 用法：assert type(x)=int and x>=0
- 如果不满足后面的expression,则会弹出
- Traceback (most recent call last):
- File "<pyshell#7>", line 1, in <module>
- assert type(n)==int and n>0
- AssertionError