一种基于时间序列性的推荐算法

A Recommender Arithmetic Based on The Order of Time

徐义峰　徐云青　刘晓平　（衢州学院 浙江衢州 324000）

摘要：聚类分析是 Web 个性化应用的一种重要技术手段，本文分析了 K－means、MCA、Buildclassification 三种典型的聚类算法。考虑到学习知识的循序渐进性，引入一个时间分量，提出一种基于时间序列性的推荐算法。在这个算法中，综合利用了三种聚类算法，将资源的后继资源在最准确的时间推荐给用户，使用户能够及时的获取最需要的资源。

关键词：聚类分析 时间序列性 推荐算法

1 引言

通常，学习知识是一个循序渐进的过程，每一方面的知识都有前驱知识和后继知识，每个用户查找需要的资源时往往也依据这个规律，因此我们可以依据权威用户的浏览记录找出知识循序渐进的规律，然后推荐给相关用户，让用户按照知识的递进性更好的学习掌握知识。为此在 K－means、MCA、Buildclassification 三种典型的聚类算法的基础上，考虑到知识学习的循序渐进性，提出了一种基于时间序列性的 W eb 信息推荐算法，它将资源的后继资源在最准确的时间推荐给用户，以更好地满足用户的需求。

2 相关的研究

2.1 K－means 聚类算法

K－means 聚类算法[2]是最常用的基于划分的方法。它以 k 为参数，把 n 个对象分为 k 个簇，以使簇内具有较高的相似度，而簇间的相似度最低。相似度的计算根据一个簇中所有对象的平均值（被看作簇的重心）来进行。算法过程：

（1）从 n 个对象中随机选择 K 个对象，每个对象初始的代表了一个簇中心。

（2）对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇。设 $C=(X_1,X_2,\cdots,X_p)$ 为已存在的簇的中心，$O=\{O_1,O_2,\cdots,O_p\}$ 为一个待分配的对象，C 与 O 的距离最常用的距离度量方法是欧几里得距离，其它还有曼哈顿距离、明考斯基距离等。欧几里得距离公式：$d(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\cdots+|x_{ip}-x_{jp}|^2}$

（3）为变化后的每个簇重新计算平均值。

（4）直到准则函数收敛则结束，否则转到第二步。其中，比较有代表性的是平均误差准则，其定义如下：$E=\sum_{p=1}^{K}\sum_{j\in O_p}|j-Avg_p|^2$，其中，E 是数据库中所有对象的平方误差的总和，j 是空间中的点，表示给定的数据对象，Avg_p 是簇 O_p 的平均值（j 和 Avg_p 都是多维的）。如果 E 值小于一个阀值，则聚类过程终止。

2.2 矩阵聚类算法（MCA）

矩阵聚类算法 MCA[3]是最常用的基于密度的方法。它从稀疏矩阵中提取密度较大的区域来进行聚类，把用户与资源的关系变换为 1 或 0 的关系。聚类的结果代表某个用户群体对某类资源集合感兴趣。通过 MAC 算法，数据对象（用户或资源项）可能分布在多个簇中。

假定数据对象的所有属性变量都是二元变量（取值为 0 或 1），一个矩阵的面积定义为矩阵的行数乘以列数，一个矩阵的密度定义为矩阵中 1 的个数除以矩阵的面积。

m 个数据对象在 n 个属性变量上的取值形成矩阵 D，它具有 m 行 n 列，假定数据对象的所有属性变量都是二元变量矩阵单元格中的值 $c_{ij}\{i=1,\cdots,m;j=1,\cdots,n\}$ 为 1 或 0。给出下列定义：

矩阵 D 的面积 S_D 等于矩阵的行数乘以列数，即 $S_D=m*n$。

矩阵 D 的密度 d_D 等于矩阵中 c_{ij} 为 1 的单元格数目除以矩阵面积。即：

$d_D = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}}{S_D}$。另外，$d_D$ 也等于矩阵中 c_{ij} 为 1 的单元格数目除以矩阵单元格总数。

给定密度阈值 ω，抽取整个矩阵中密度大于 ω 的子矩阵。

2.3 类层次结构算法（Buildclassification）

Buidlclassification 算法[4]能根据各个类之间的相似程度形成类层次结构，从而能有效反映类之间的相似性。因为考虑到在类层次结构中的类不是非常多，所以 Buildclassification 算法采用自底向上的分层聚类方法。Buildclassification 算法避免了一些系统必须按照某些拓扑结构进行分层聚类的方法，此外当相似度小于某一指定的阈值就不进行聚类分析，而不必像一些分层聚类方法那样直接合并到只剩下一个类为止的情况，这样做加快了聚类速度。

2.4 用户权威性

不同用户对资源的评价可靠程度不同，有经验的用户或权威用户能够给资源更准确、更客观的评价。可见，用户对资源评价的可靠程度（即用户的权威性[5]），反映了用户评价的稳定性。因此权威用户对资源的评价更值得其他用户参考。权威性的确定：

（1）用户评价过的资源数量。如果一个用户在某一方面评价过很多资源，那么他已经具备一定的经验，并在评价时表现出自己的品味，因此用户权威性的第一个特征可以利用用户对资源的评价数目来反映。

$W_1 = \begin{cases} 1, & n_u \geq A \\ \frac{n_u}{A}, & n_u < A \end{cases}$，其中，$n_u$ 表示用户 u 所评价过的资源数目。A 为一个常数，称作惩罚数目，一般取 50。当 $n_u < A$ 时，削弱该用户的权威性，若 $n_u >= A$ 时权重为 1。

（2）用户学术背景。如果一个用户评价的资源不多，用户的学术背景也将影响用户权威性。例如，教授和学生同时从事某个领域的研究，一般情况下教授的评价更具有客观性。所以，用户权威性的第二个特征可以利用用户的学术背景来定义：W_2 为 1（教授）；为 K_1（副教授）；为 K_2（讲师）；为 K_3（学生）等。

以上两个方面都是从用户自身角度来衡量的，因此有：$AU_1(u) = \alpha_1 W_1 + \alpha_2 W_2$，其中 $\alpha_1 + \alpha_2 = 1$。

（3）从资源的角度来衡量。当一个资源获得足够多的评价时，这个资源评价的平均值可以用来衡量该资源真正的质量或品质，而一个权威用户一般能正确评价事物的本质特征，因此一个用户的评价值接近资源品质的程度，也就是接近资源评价平均值的程度可以用来刻画用户权威性的第三个特征，定义：

$AU_2(u) = \frac{\sum_{i \in V_u}\left(1 - \frac{|v_{u,i} - \bar{v}_i|}{Max - Min}\right)}{n_u}$，其中 v_u 表示用户 u 评价过的资源集合，$v_{u,i}$ 表示用户 u 对资源 i 的评价值，$\bar{v}_i$ 表示资源 i 评价的平均值，Max 和 Min 表示评价的最大值和最小值。

综合用户权威性的三个特征，那么一个用户的权威值可以表示成：$AU(u) = AU_1(u) \times AU_2(u)$

3 基于时间序列性的推荐算法

考虑到学习知识的循序渐进性，提出一种基于时间序列性的推荐算法。在这个算法中，综合利用了三种聚类算法，将资源的后继资源在最准确的时间推荐给用户，使用户能够及时的获取最需要的资源。这种算法的思想是：找到某个主题的权威用户，将其评价过的资源进行内容聚类，并结合评价的时间找到关于这个主题的前驱知识和后继知识，然后将当前用户研究知识的后继知识按先后顺序推荐给用户。

3.1 算法过程

（1）利用 MCA 算法计算出资源的种类。一般情况下，一个用户在很短的一个时间段内研究的资源属于同一领域或反映同一个主题，我们借鉴 MCA 算法用一个矩阵代表一个用户在不同时间访问过的所有资源。行表示时间，列表示用户访问过的资源的关键词。这里列为关键词而不是资源，是因为资源是孤立的，大多数的资源仅在一个时间被访问，关键词则不同，同一时间可以访问不同的关键词，同一关键词也可以在不同时间被访问。因此聚类后的子矩阵表示在同一时间内访问过的相似关键词。利用 MCA 算法得到的是某个时间段内感兴趣的某些关键词的集合。

依据公式计算 S_D、d_D；给定阈值 ω，从稀疏矩阵中抽取密度大于 ω 的区域进行聚类，另外，在具体实施时，要附加限制条件，如最小的行数、最小的列数等参

数。最后输出形成聚类的所有子矩阵。每个子矩阵为一个类，依照时间次序得到每类中的资源。

（2）利用 K－Means 算法找到每种资源类别中包含的资源。依据第一步中集合的总数，我们可以得出用户访问过的资源划分的数量，即 K－Means 算法中的类别数。依据 K－Means 算法计算出每类中包含的资源。

（3）利用 Builderclassification 算法找到由资源类别形成的资源大类。用户的兴趣是多样的，在每个大的资源类别中都有许多知识的分支，因此我们要将找到的小的资源类别归属到上一级资源类别。如在 A 资源大类中，依据时间次序组织资源小类别 $a_1, a_2, \cdots, a_n$。根据人们认识知识的特征，a_1 是 a_2 的前驱知识，a_3 是 a_2 的后继知识。

（4）依据知识的连贯性进行推荐。找到用户在这一大类资源中相似用户，比较目前用户访问的资源小类 b_n 和相似用户的资源序列表中哪类（如 a_l）资源相似，即 a_l, b_n 为最相近的知识类，所以 a_{l+1} 类的资源是用户 B 最需要的资源。

（5）推荐资源

① 在相似用户中，根据权威性的计算方法找到权威用户。

② 用户 A$\{a_1, a_2, \cdots, a_m\}$ 向用户 B$\{b_1, b_2, \cdots, b_n\}$ 推荐资源。按次序取得用户 B 的最后一个关键词类 b_n（即用户目前感兴趣的资源类），在权威用户的资源类中找到和 b_n 资源类中最相似的资源类 a_l。

③ 用余弦相似度计算 b_n 和 a_l 最相似的类，找到 max（Sim（b_n, a_l））。其中：

$$sim(b_n, a_l) = \frac{b_n \cdot a_l}{\|b_n\| \|a_l\|} = \frac{\sum_{i=1}^{k} a_{ji} \cdot b_{nj}}{\sqrt{\sum_{i=1}^{k} a_{jl}} \cdot \sqrt{\sum_{i=1}^{k} b_{nl}}}$$

④ 权威用户中越是相似类的近邻后继知识推荐的可能性越大，因此推荐列表中增加一个后继知识度分量。若资源 j 属于关键词集合 k，则

$P_{a,j} = W_t * P_{a,j}$，其中 $W_t = \frac{|k-i|}{m-i}$ 推荐值排名在前 N 位的，成为用户的最终推荐列表，推荐给用户。

3.2 推荐结果的整理和合并

在推荐系统中每次推荐固定数目的资源给用户，但是满足用户兴趣的资源的数量也许远远超过了这个数量，而且某个资源可能被系统按照多种方式推荐。因此，需要对这些结果进行合并、整理并筛选出最符合用户兴趣但是又没有推荐给用户过的一定数目的资源。这里利用一些启发式规则来合并多种推荐方式的推荐结果，收到良好效果。一些启发式规则有：（1）如果一项资源被多种方式推荐则它最优先被推荐；（2）预测一个用户对一项资源感兴趣程度，预测值高的资源优先被推荐；（3）协同过滤推荐优于基于内容过滤推荐；（4）后继资源优于被推荐；（5）最新资源优于被推荐。

3.3 算法描述

输入：矩阵

输出：推荐值

过程：

（1）利用 MCA 聚类算法计算出用户感兴趣的资源小类，s＝s＋k；

（2）将 s 代入 K－means 聚类算法，找到用户感兴趣的资源子矩阵集；

（3）将子矩阵集利用 Buildclassification 算法形成层次结构；

（4）按照时间将每个大类中的子类排序；

（5）计算推荐值

For l＝1 to 权威用户 a 集合数

计算 a_l 和 b_n 的 sim（b_n, a_l）

end－for

取 Max（sim（b_n, a_l））

将 a_{l+1} 推荐给 b

（6）产生推荐列表。

4 结束语

针对用户学习知识的循序渐进性的过程，本文提出了基于时间序列性的推荐算法，在这个算法中，综合利用了三种聚类算法，并依据权威用户的浏览记录找出知识循序渐进的规律，将资源的后继资源在最准确的时间推荐给用户，让用户按照知识的递进性更好地掌握知识。研究表明，基于时间序列性的推荐算法明显地提高了推荐质量，具有一定的有效性和可行性。

（下转第 29 页）

(上接第 25 页)

参考文献

1 Cecilia M. Procopiuc. Clustering Problems and their Applications(a survey). Department of Computer Science,Duke University,1997.

2 J. Han and Micheline Kamber. Data Mining: Concepts and Techniques[M]. Higher Education Press, Morgan Kaufmann Publishers,2001,5.

3 艾旭生，基于数据挖掘的 web 个性化服务研究[D]，江苏 苏州大学,2003。

4 韩立新、陈贵海、谢立，一个面向 Internet 的个性化信息检索系统模型[J]，电子学报,2002(2):241-243。

5 曾春，信息过滤的概念表示与算法研究[D]，北京清华大学,2003。