

一种优化的组合协同过滤算法

谢翔宙, 张延园

(西北工业大学 计算机学院, 陕西 西安 710029)

摘 要: 改进了传统协同过滤算法中最近邻搜索这一关键步骤, 提出了一种结合概念层次和用户局部兴趣相似的协同过滤算法, 使推荐系统在用户矩阵整体稀疏局部密集时依然能产生较好的推荐. 该算法应用于基于 iPhone 平台开发的 EatMe 菜肴推荐系统, 实验证明改进算法比传统协同过滤算法有更高的查全率.

关键词: 在线推荐系统; 协同过滤; 概念层次; 局部相似性; 查全率

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2010)12-0102-03

An Improved Combining Collaboration Filtering Algorithm

XIE Xiang-zhou, ZHANG Yan-yuan

(Department of Computer Science, Northwestern Polytechnic University, Xi'an 710029, China)

Abstract: This paper improved on computing neighbor users for active users, which was the crucial step of traditional collaboration filtering algorithm. It proposed a collaboration filtering algorithm combining concept hierarchy and users' partial similarity so that it could still work on the sparse but partial dense user matrix. This algorithm was applied in EatMe dish recommendation system based on iPhone, and the experimental results showed that the improved algorithm can provide better recall than traditional collaboration filtering algorithm.

Key words: online recommendation system; collaboration filtering; concept hierarchy; partial similarity; recall

1 引言

目前协同过滤算法是个性化推荐系统中应用较为广泛的推荐技术. 它利用个性化推荐系统的用户项偏好矩阵, 搜索与当前用户行为兴趣最相近的用户集, 再将用户集最感兴趣的项作为当前用户的推荐结果. 当用户项偏好矩阵密集时, 它通常表现出良好的性能, 然而当遇到矩阵整体稀疏局部密集的情况时, 由于用户评分项的交集很小, 协同过滤技术计算用户相似度的运算法则较难发现最近邻.

文中提出了一种优化的组合协同过滤算法. 首先利用概念层次的思想引入用户一项目簇矩阵以改进用户相似度计算公式, 接着利用用户局部兴趣相似原理, 根据项目属性划分矩阵并在局部数据集上寻找用户最近邻向量, 再合并为最终的用户最近邻向量以改进最近邻搜索, 最后使用传统协同过滤预测评分公式, 生成在线 TOP-N 推荐结果. 该算法

应用于基于 iPhone 平台开发的 EatMe 菜肴推荐系统, 此系统利用移动 GIS 确定用户当前所在的餐馆并向用户推荐该餐馆的菜肴.

2 协同过滤算法

协同过滤算法的主要思想是先找与目标用户兴趣相似的用户, 再将这些用户感兴趣的内容推荐给此用户. 基于用户的协同过滤方法主要分为三个步骤^[1]:

(1) 形成用户一项目矩阵

用户一项目矩阵可用一个 $M \times N$ 矩阵 $R(M, N)$ 来表示, 其中 M 行代表 M 个用户, N 列代表 N 个项目, 元素 $R(i, j)$ 表示用户 i 对项目 j 的评分. 根据项的类型和系统的实现方式, 有不同的用户评分方法. 比如用 1 表示用户对某项目感兴趣, 0 表示不感兴趣; 也可用 1, 2, 3, 4, 5 等离散值表示对该商品的偏爱度.

(2) 最近邻搜索

先计算目标用户 i 与各用户之间的相似程度,

可以使用皮尔森相关系数或余弦相似性等度量方法.其定义如下:

余弦相似系数:

$$\text{sim}(i, j) = \frac{\sum_{u \in I_{i,j}} R_{i,u} * R_{j,u}}{\sqrt{\sum_{u \in I_{i,j}} R_{i,u}^2} \sqrt{\sum_{u \in I_{i,j}} R_{j,u}^2}} \quad (1)$$

Person 相似系数:

$$\text{sim}(i, j) = \frac{\sum_{u \in I_{i,j}} (R_{i,u} * \bar{R}_i)(R_{j,u} * \bar{R}_j)}{\sqrt{\sum_{u \in I_{i,j}} (R_{i,u} * \bar{R}_i)^2} \sqrt{\sum_{u \in I_{i,j}} (R_{j,u} * \bar{R}_j)^2}} \quad (2)$$

其中, $\text{sim}(i, j)$ 为用户 i 和用户 j 之间的相似度, $I_{i,j}$ 表示用户 i, j 的共同评分项目, $R_{i,u}$ 和 $R_{j,u}$ 分别为用户 i 和用户 j 对项目 u 的评分项目. 再根据各用户与目标用户 i 之间的相似度大小确定该目标用户 i 的邻居 $N = \{N_1, N_2, \dots, N_n\}$.

(3) 生成 Top-N 推荐结果

根据用户最近邻相似度对目标用户 i 未评分项目进行预测, 选出预测分数最高的 N 项作为 Top-N 推荐结果. 预测公式如下:

$$P_{i,k} = \bar{R}_i + \frac{\sum_{m=1}^n \text{sim}(i, m) * (R_{m,k} - \bar{R}_m)}{\sum_{m=1}^n \text{sim}(i, m)} \quad (3)$$

用户最近邻发现的质量将直接影响着最终的推荐结果. 当用户-项目矩阵整体稀疏而局部密集时, 传统协同过滤算法在计算用户相似度 $\text{sim}(i, j)$ 时精确度不够, 且由于整体数据稀疏, 传统协同过滤算法较难发现最近邻. 针对这种情况, 文中提出了一种优化的组合协同过滤算法.

3 优化算法

3.1 用户-项目矩阵预处理

由于用户和项目之间存在着地域关联, 因此用户-项目偏好矩阵呈现出局部密集整体稀疏的特点, 用户-项目矩阵如下:

$$\begin{pmatrix} 3 & 2 & 1 & 5 & \dots & & & & & & \\ & & & 2 & & & & 3 & 1 & 4 & \\ 4 & 2 & 1 & 5 & & & & & & & \\ \dots & & & & 3 & 1 & \dots & & & & \\ \dots & & & & 2 & 5 & \dots & & & & \\ \dots & & & & 5 & 3 & & 2 & & & \\ 5 & 1 & 4 & & 2 & 4 & & & & & \\ & & & & & & & 3 & 4 & & \\ 1 & 4 & 3 & 2 & & 3 & & 2 & 5 & 2 & \\ & & & 3 & & & & 5 & 3 & & \end{pmatrix}$$

用户-项目矩阵预处理的目的在于针对矩阵结构特点, 为计算用户相似性作数据准备, 既能尽量避免不必要的系统花销, 又能获取合理的用户最近邻向量. 预处理过程如下:

3.1.1 形成子用户-项目矩阵

由于个体的差异性, 两个人的兴趣完全一致几乎是不可能的. 但是, 两个人在某一方面或者几方面有相似兴趣的可能性还是很大的^[2]. 根据用户兴趣局部相似性的思想, 把整个用户-项目矩阵空间划分为若干类别, 每项(或用户)可能属于多个类别^[3]. 文中使用项目的地域属性, 最终形成 50 个子用户-项目矩阵.

对划分后的子用户-项目矩阵, 为了减少不必要的系统花销且降低矩阵稀疏度, 将矩阵进行以下优化:

(1) 利用用户点菜行为填充子用户-项目矩阵. 推荐系统中记录了用户的点菜历史, 有些菜品用户可能点过了, 但并没有对菜品进行评分. 这部分数据也反应了用户的饮食兴趣. 因此可按用户 i 平均分再减 1 来预测.

(2) 对于填充后的子用户-项目矩阵, 我们设定一个阈值 λ 进行过滤, 即用户评分数小于 λ 则视为无参考意义的评分, 文中取 $\lambda = 2$.

3.1.2 生成子用户-项目簇矩阵

概念层次结构(concept hierarchy)表示把一组较低概念映射到与它们相对应的较高级概念的次序, 这种映射可以按偏序关系($<$)来组织概念集^[4]. 根据层次关系 $C_i < C_j$, 可以形成用户-项目簇矩阵.

传统协同过滤算法通常只关注用户和项目之间的相似度, 事实上, 用户和项目簇之间的相似度也在一定程度上反应了用户的兴趣. 比如两个喜欢同一些菜品的用户之间存在相似性, 两个都喜欢同一家餐厅的用户其实也存在着相似性. 同时考虑用户-项目簇的相似性也降低了矩阵的稀疏度, 即使子层因为矩阵稀疏找不到最近邻, 父层也可以为最近邻搜索提供参考. 文中存在层次关联关系: 菜品餐馆, 具体例子如图 1 所示.

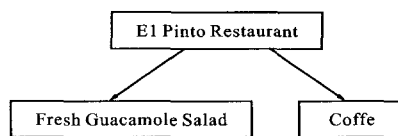


图 1 层次关联关系

用户-项目簇矩阵生成过程为:

(1) $A = P_0 * C_0$, 其中 P_0 为评分型用户—项目矩阵, C_0 为项目类关系矩阵, 描述父项目类和子项目类的内含关系, 如果父项目类包含子项目, 则 C_i 为 1, 否则为 0.

(2) $B = P_0' * C_0$, 其中 P_0 为布尔型的用户—项目矩阵, C_0 为项目类关系矩阵, 描述父项目类和子项目类的内含关系, 如果父项目类包含子项目, 则 C_i 为 1, 否则为 0.

(3) $C = \{c \mid c_{ij} = a_{ij}/b_{ij}, a_{ij} \in A, b_{ij} \in B\}$

由此可得, 生成的用户—项目簇矩阵的评分为对应用户—项目矩阵的平均评分值.

3.2 最近邻搜索

根据 3.1 节用户兴趣局部相似性的思想^[2], 先对各子用户—项目簇矩阵和子用户—项目矩阵进行相似性计算获取用户最近邻向量, 然后合并目标用户 i 的所有用户最近邻向量集.

(1) 计算局部用户相似度, 得到用户最近邻向量

对于区域用户, 结合该区域的子用户—项目簇矩阵和子用户—项目矩阵, 使用加权余弦公式计算用户之间的相似度, 公式如下:

$$\begin{aligned} \text{sim}(i, j) = & \alpha * \frac{\sum_{u' \in I'_{ij}} T_{i,u'} * T_{j,u'}}{\sqrt{\sum_{u' \in I'_{ij}} T_{i,u'}^2} \sqrt{\sum_{u' \in I'_{ij}} T_{j,u'}^2}} \\ & + \beta * \frac{\sum_{u \in I_{ij}} R_{i,u} * R_{j,u}}{\sqrt{\sum_{u \in I_{ij}} R_{i,u}^2} \sqrt{\sum_{u \in I_{ij}} R_{j,u}^2}} \quad (4) \end{aligned}$$

其中 $T_{i,u'}, T_{j,u'}$ 分别表示用户 i, j 对项目簇 u' 的评分, I'_{ij} 表示用户 i, j 共同评分的项目簇. $R_{i,u}, R_{j,u}$ 分别表示用户 i, j 对项目 u 的评分, I_{ij} 表示用户 i, j 共同评分的项目. α 和 β 分别表示两种相似度的权重, 文中使用 $\alpha = 0.3, \beta = 0.7$.

(2) 合并用户最近邻向量集

对目标用户 i 的用户最近邻向量集取并集, 生成最终的用户最近邻向量. 即

$$\text{neighbor}(i) = \{u \mid \exists u \in \bigcup_{j=1}^n U_j, U_j \notin \emptyset\} \quad (5)$$

其中, U_i 表示每个区域的用户最近邻向量, n 为区域数, 文中 $n = 50$, 即为矩阵划分数.

为了不影响实时性, 以上的计算离线实现, 且基于用户兴趣的相对稳定性, 无需过于频繁计算用户最近邻向量.

上述最近邻算法的优势在于:

(1) 使用加权的余弦公式计算用户之间的相似度, 综合考虑到了项目和项目簇的评分数据, 更全面

地反应了用户的兴趣. 同样的数据得到了更深层次的挖掘, 提高了推荐算法的质量.

(2) 利用用户兴趣局部相似原理, 对区域内用户使用相似度算法寻找最近邻, 再合并最近邻向量, 得到的最终用户最近邻向量更加合理和丰富.

3.3 项目预测评分, 在线 TOP—N 推荐

和传统 TOP—N 推荐不同, 本算法仅对用户目前所需要的项目集合 D 内的项目进行预测. 比如用户在某一餐馆内, 算法只对该餐馆的菜品进行预测. 推荐候选项目集为该项目集合 D 内, 被目标用户 i 的邻居所喜好, 且未被 i 访问过的项目. 预测公式和传统协同过滤方法相同:

$$P_{i,k} = \bar{R}_i + \frac{\sum_{m \in \text{neighbor}(i)} \text{sim}(i, m) * (R_{m,k} - \bar{R}_m)}{\sum_{m \in \text{neighbor}(i)} \text{sim}(i, m)} \quad (6)$$

最后按 $P_{i,k}$ 大小降序排列, 选出预测分数最高的 N 项作为 Top—N 推荐结果.

4 实验评估

4.1 实验数据集

采用基于 iPhone 平台开发的 EatMe 菜肴推荐系统提供的数据集, 对文中算法与传统协同过滤算法进行比较. 数据集记录了 186 679 的餐馆数据, 11 254 579 的菜品数据和 3 184 个用户的 137 250 条评分数据.

4.2 评估标准

文中使用 Sarwar 提出的^[5]查全率(recall)作为算法在精确性方面的评估标准, 即

$$\text{Recall} = 100 * \frac{|P_i \cap TE_i|}{|TE_i|} \quad (7)$$

其中, P_i 为用户 i 的最终 TOP—N 推荐; TE_i 为用户的测试集.

4.3 测试方法与实验结果

取其中 250 个用户对 3 000 个菜品的 13 234 条评分记录, 将数据集的 80% 分为训练集, 20% 分为测试集, 用 5 折交叉法得出 5 组不同的训练集和测试集, 将 5 次实验结果平均. 将训练集的数据分别作不同程度的稀疏化, 即每 N 个数据抽取一个作为输入, $1/N$ 表示抽取率^[5], 测试传统协同过滤算法与优化协同过滤算法的查全率. 如图 2 所示.

4.4 结果分析

从图 2 中可以看到, 由于在局部较为密集的矩阵中, 使用优化算法得到的最近邻较为合理. 因此使

(下转第 108 页)

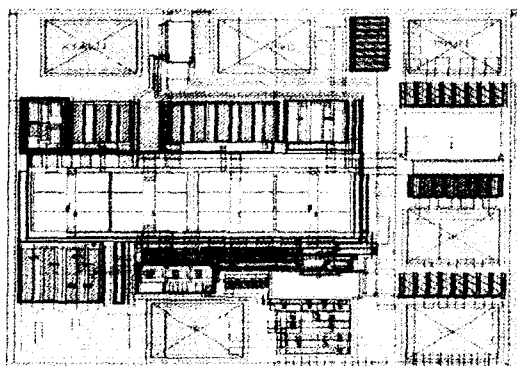


图7 芯片版图

5 结束语

文中采用 SMIC 0.35 μm CMOS 工艺,成功地设计了一款低功耗、输出高稳定性的晶体振荡器芯片。仿真结果表明:正常工作,电源电压为 5 V 时,晶体振荡器芯片消耗总电流低于 5 mA,振荡器模块消耗电流仅为 0.6 mA;引入低功耗模式后,振荡模块消耗电流降低至 3 μA 以下;各种工艺条件下,负载电容为 15 pF 时,输出占空比为 $50 \pm 0.8\%$ 的方波信号,其频率随电源电压的变化率仅为 0.5×10^{-6} 。与传统结构相比,功耗降低了 60%,频率随电压稳定性提高了 10 倍,非常适合应用于高精度时钟系统。

(上接第 104 页)

用在线 TOP-10 推荐在抽取率 1/5 时,查全率比传统协同过滤算法高出了 26.45%。但随着整体稀疏度的增大,传统协同过滤算法和优化的协同过滤算法能预测出来的项目急剧减少,因此查全率都随之降低,但整体来说优化的协同过滤算法比传统协同过滤算法有较高的查全率。

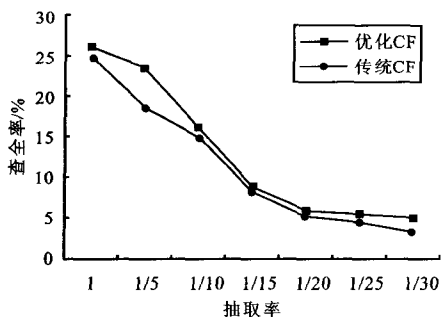


图2 算法查全率比较

5 结束语

文中结合基于概念层次的协同过滤算法和基于用户兴趣局部相似的协同过滤算法的思想,提出了一种优化的组合协同过滤算法,并在实际个性化推

参考文献:

- [1] 赵声衡, 赵英. 晶体振荡器[M]. 北京: 科学出版社, 2008.
- [2] 曾健平, 王阅, 何先良. 石英晶体振荡器的集成化设计[J]. 微电子学与计算机, 2009, 26(2): 30-32.
- [3] 徐赏林, 李威, 王继安, 等. 高线性度低频压控振荡器的设计[J]. 微电子学与计算机, 2006, 23(8): 42-45.
- [4] Huang Qiuting. Design considerations for high-frequency crystal oscillators digitally trimmable to sub-ppm accuracy[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1997, 5(4): 410-416.
- [5] Tan J Y J, Ng K A, Xu Y P. An integrated low power one-pin crystal oscillator[C]//IEEE Asia Pacific Conference on Circuits and Systems. Macao: IEEE Press, 2008: 1352-1355.
- [6] 高明伦, 胡永华, 李丽. 深亚微米芯片设计中相位噪声的讨论[J]. 微电子学与计算机, 2002, 19(9): 57-60.
- [7] Razavi B. 模拟 CMOS 集成电路设计[M]. 西安: 西安交通大学出版社, 2000.

作者简介:

陈红梅 女, (1986-), 硕士研究生. 研究方向为模拟 CMOS 集成电路设计.

徐静平 男, (1958-), 教授, 博士生导师. 研究方向为集成电路设计与器件工艺.

荐系统中得到了较好的应用. 同时, 实验结果表明对于局部较为密集而整体稀疏的用户项目偏好矩阵, 改进后的算法提高了算法精确性.

参考文献:

- [1] Manos Papagelis, Dimitris Plexousakis. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents[J]. Engineering Applications of Artificial Intelligence, 2005(18): 782-784.
- [2] 吴发青, 贺木梁, 夏薇薇, 等. 一种基于用户兴趣局部相似性的推荐算法[J]. 计算机应用, 2008, 28(8): 1981-1984.
- [3] 高凤荣, 杜小勇, 王珊. 一种基于稀疏矩阵划分的个性化推荐算法[J]. 微电子学与计算机, 2004, 21(2): 59-60.
- [4] 熊馨, 王卫平, 叶跃祥. 基于概念分层的个性化推荐算法[J]. 计算机应用, 2005, 25(5): 1006.
- [5] 张晓敏, 王茜. 基于概念层次树的个性化推荐算法[J]. 计算机工程, 2007, 33(24): 58-59.

作者简介:

谢翔宙 女, (1986-), 硕士研究生. 研究方向为软件工程专业与网络软件.

张延园 男, (1956-), 教授, 硕士生导师. 研究方向为网络存储、软件工程与网络软件.