一种新颖的个性化视频搜索排名算法

李 慧, 李存华, 王 霞

(淮海工学院 计算机工程学院, 江苏 连云港 222002)

[摘要] 视频搜索是目前信息检索领域研究的热点. 提出一种利用协同过滤技术来实现个性化的视频搜索. 该算法根据用户项目兴趣相似度来计算目标项目的得分, 从而为每个用户产生一个推荐列表. 实验结果表明该排名算法较 MDB搜索和 Google搜索的结果在用户满意度上有很明显的提高.

[关键词] 协同过滤, 推荐, 视频搜索, 个性化

[中图分类号] TP 313 [文献标识码] A [文章编号] 1672-1292(2008)04-0182-04

A Novel Individualized Video Search Ranking Algorithm

Li Hui, Li Cunhua, Wang Xia

(Department of Computer Science, Huaihai Institute of Technology, Lianyungang 222005, China)

Abstract Video search is a hotspot in current information searching field. The paper presents a new video search by using collaborative filtering technique to realize individuality. The algorithm computes the scores of the target items according to the similarity degree of the items in which the users are interested, and thus generates a recommendation list for every user. The experimental results show that compared with MDB search and Google search, our algorithm can obtain better rank results and improve user's satisfaction highly.

Key words collaborative filtering; recommender; Video search; individuation

为了更好地为在线用户提供服务, 个性化推荐系统成为网络信息检索领域的一项重要研究内容. 它通过判断用户兴趣, 为其选择并推荐适当的信息来解决用户的信息过载和迷失问题. 协同过滤技术[1, 2]是应用最成功的技术. 本文提出了一种基于项目的协同过滤技术来改变视频排名的算法. 其基本思想就是基于兴趣相似的用户评分数据向目标用户产生推荐, 以此提高用户对视频搜索结果的满意度.

1 构建用户兴趣模型

1.1 用户兴趣模型的描述

个性化推荐主要通过用户需求分析、信息搜索和结果表示等步骤, 为用户主动地提供符合其偏好的信息. 其中, 识别用户需求是良好推荐的基础, 其关键技术是用户模型的构造. 为了跟踪和学习用户的兴趣行为, 有必要为每个用户建立一个模型. 用户模型是用于存储用户的兴趣, 存储和管理用户的行为历史, 存储学习用户行为的知识和进行相关推导的知识集合.

Blog作为一种新出现的信息发布模式以其独特的魅力吸引着越来越多的网络用户. Blog的开放性是自身固有的, 用户的每次发布信息都被一个惟一的 URL定位, 任何人可随意浏览. 由于 Blog上发布的信息具有公开性与个性化等特性, 因此, 本文提出一种新的方法: 基于 Blog中的内容来构建用户兴趣模型. 该方法根据用户在 Blog中发布的信息来建立一个兴趣模型, 从而为实现个性化的推荐奠定了基础.

1.2 用户兴趣模型的建立

用户兴趣是通过分析用户在 Blog中发布的信息而得到的. 用发布的关键词 (keyword) 来表示用户的兴趣. 即: 用向量来描述一个用户 u, 在用户 u 发布的信息中抽取 m 个关键词 $k1, k2, \cdots, km$ 来构建一个用

收稿日期: 2008-06-18

通讯联系人: 李 慧, 硕士研究生, 研究方向: 数据挖掘、智能信息系统. E-mial: shufanzs@126.com

户, 即用一个 m 维的权值 $\boldsymbol{P} = (W_{k1}, W_{k2}, \cdots, W_{km})$ 来表示一个用户模型. 最终构建出的用户兴趣模型如图 1所示.

为了表示每个词 k 在描述每个用户兴趣时所起的作用的大小, 需要为每个词分配一个权值. 目前有多种加权方案, 其中应用最广泛的自动产生权值的方案是 tf* idf加权方案. 本文也采用 tf* idf加权方案.

	$k1$	…	km	..	kn
$\boldsymbol{P}_1$	$W_{1,1}$	..	$W_{1,m}$	…	/
…	…	…	…	…	…
$\boldsymbol{P}_j$	W_{j1}	…	/	…	W_{jn}
..	…	…	…	…	…
$\boldsymbol{P}_k$	/	…	$W_{k,m}$	…	$W_{k,n}$

图 1　用户兴趣模型矩阵

Fig. 1　The matrix of user interest model

2　项目预测得分

2.1　协同推荐算法

我们使用基于项目的协同过滤算法[3] (CF)来计算用户对返回结果的预测得分. 算法基于这样一个假设: 如果大部分用户对一个项目的评分比较相似, 则当前用户对这些项目的评分也比较相似. 基于项目的协同过滤推荐系统使用统计技术找到目标项目的若干最近邻居, 可以根据当前用户对最近邻居的评分预测当前用户对目标项目的评分, 产生对应的推荐列表. 将此推荐算法应用于本系统, 就是将与目标用户具有相似兴趣爱好的项目推荐给目标用户, 以此实现视频的个性化推荐.

在使用 CF查找最近邻居的过程中, 都需要量度项目之间的相似性. 度量项目 i 和项目 j 之间相似性的方法如下: 首先得到对项目 i 和项目 j 都感兴趣 (在 Blog中如果用户 u 发布的信息包含关键词 k, 则定义用户 u 对项目 k 感兴趣.)的所有用户, 然后通过不同的相似度量方法计算项目 i 和项目 j 之间的相似性, 记为 $\mathrm{sim}(i, j)$. 度量项目相似性的方法有许多种, 主要包括以下三种: 余弦相似性, 相关相似性和修正的余弦相似性.

(1) 余弦相似性 (Cosine): 用户评分看作为 n 维项目空间上的向量, 如果用户对项目没有进行评分, 则将用户对该项目的评分设为 0. 用户间的相似性通过向量间的余弦夹角度量. 设用户 i 和用户 j 在 n 维项目空间上的评分分别表示为向量, 则用户 i 和用户 j 之间的相似性 $\mathrm{sim}(i, j)$ 为

$$\mathrm{sim}(i, j) = \cos(\boldsymbol{i} \cdot \boldsymbol{j}) = \frac{\boldsymbol{i} \cdot \boldsymbol{j}}{\|\boldsymbol{i}\| \times \|\boldsymbol{j}\|},$$

其中分子为两个用户评分向量的内积, 分母为两个用户向量模的乘积.

(2) 相关相似性: 设用户 i 和用户 j 共同评分过的项目集合用 I_{ij} 表示, 则用户 i 和用户 j 之间的相似性 $\mathrm{sim}(i, j)$ 通过 Pearson相关系数度量:

$$\mathrm{sim}(i, j) = \frac{\sum_{c \in I_{ij}} (R_{ic} - \overline{R_i})(R_{ic} - \overline{R_j})}{\sqrt{\sum_{c \in I_{ij}} (R_{ic} - \overline{R_i})^2} \sqrt{\sum_{c \in I_{ij}} (R_{jc} - \overline{R_j})^2}},$$

其中 R_{ic} 表示用户 i 对项目 c 的评分, $\overline{R_i}$ 和 $\overline{R_j}$ 分别表示用户 i 和用户 j 对项目的平均评分.

(3) 修正的余弦相似性: 在余弦相似性度量方法中没有考虑不同用户的评分尺度问题, 修正的余弦相似性度量方法通过减去用户对项目的平均评分改善上述缺陷. 设用户 i 和用户 j 共同评分过的项目集合用 I_{ij} 表示, I_i 和 I_j 分别表示用户 i 和用户 j 评分过的项目集合, 则用户 i 和用户 j 之间的相似性 $\mathrm{sim}(i, j)$ 为:

$$\mathrm{sim}(i, j) = \frac{\sum_{c \in I_{ij}} (R_{ic} - \overline{R_i})(R_{ic} - \overline{R_j})}{\sqrt{\sum_{c \in I_i} (R_{ic} - \overline{R_i})^2} \sqrt{\sum_{c \in I_j} (R_{jc} - \overline{R_j})^2}}.$$

2.2　基于项目的最近邻查询

预测用户 i 对项目集合 U_{ij} 中未评分项目的评分是基于项目相似性的协同过滤推荐算法的关键. 设用户 i 在项目空间 U_{ij} 中未评分的项目集合用 N_i 表示, 即:

$$N_i = U_{ij} - I_i.$$

对任意项目 $p \in N_i$, 使用如下方法预测用户 i 对项目 p 的评分 P_{ip}:

(1) 计算项目 p 与其他项目之间的相似性, 与计算用户间相似性类似, 首先需要得到对项目 i 和项目 j 评分的所有用户评分, 然后通过 2.1节中介绍的各种相似性度量方法计算项目 i 和项目 j 之间的相似性.

(2) 将相似性最高的若干项目作为项目 p 的邻居项目集合, 即在整个项目空间中查找项目集合 $\boldsymbol{M}_p$ =

$\{I_1, I_2, \cdots, I_v\}$, 使得 $p \notin M_p$, 并且项目 I_1与项目 p的相似性 $\mathrm{sim}(p, I_1)$最高, 项目 I_2与项目 p的相似性 $\mathrm{sim}(p, I_2)$次之, 依此类推.

(3)得到M_p后, 采用文[4]中提出的方法预测用户 i对项目 p的评分 $P_{i,p}$:

$$P_{i,p} = \frac{\sum_{n \in M_p} \mathrm{sim}_{p,n} \times R_{i,n}}{\sum_{n \in M_p} (|\mathrm{sim}_{p,n}|)}.$$

3 产生推荐

通过本文提出的相似性度量方法得到目标用户的最近邻居, 下一步需要产生相应的推荐[5,6]. 设目标项目 TI的最近邻居集合用 $NN_T = \{NN_1, NN_2, \cdots, NN_K\}$表示, 则用户 u对项目 T的预测评分 $P_{u,T}$可以通过用户 u对最近邻居集合 NN_T中项目的评分得到, 计算方法如下:

$$P_{u,T} = R_T + \frac{\sum_{n \in NN_T} \mathrm{sim}(T, n) \times (R_{u,n} - \overline{R_n})}{\sum_{n \in NN_T} (|\mathrm{sim}(T, n)|)},$$

其中 $\mathrm{sim}(T, n)$表示目标项目 T与最近邻居 n之间的相似性, $R_{u,n}$表示用户 u对项目 n的评分. R_T和 R_n分别表示对项目 T和项目 n的平均分值.

在求得用户对项目的预测评分后, 再结合视频页面本身的 PageRank值, 得到对于用户 u输入查询词 q后, 视频 i最终的得分 Rank值为:

$$\mathrm{Rank}(i, q, u) = PR(i) + P(u, i).$$

通过上述方法预测当前用户对未评分项目的评分, 然后选择预测评分最高的若干个视频作为推荐结果反馈给当前用户.

3.1 数据准备

Yahod movies站点包含 3 500多部电影, 并对电影视频做好了分类, 这极大地方便了数据的准备工作. 为了不失一般性, 将该网站所包含的 19种视频类型, 每种取排名前 20的视频. 除去重复的, 共计得到 337个返回结果, 即 337个视频页面.

3.2 实验结果及用户评价

用户对搜索结果的评价至关重要, 本文以用户对返回结果中排名前 5项的满意度作为评价标准. 具体公式为:

$$\text{满意度} = \left(\frac{t}{5}\right) \times 100\%,$$

其中, t为用户认为对自己有用的返回结果.

对同样实验数据, 分别用 IBDM (InternetMovieDatabase)、Google Search和本文提出的基于协同过滤的推荐算法进行实验. 为了测试排名结果的满意度, 我们请来了 40个参与者, 其中 20个为专业人士, 其他为普通用户. 每个人投入 10个查询词, 然后分别在 IBDM、Google Search的前 5个查询结果和本文算法的查询结果中挑出适合自己的查询结果(即用户感到满意的查询结果), 则 40个人共有 400个查询词. 如有相同的, 算 1个. 具体的查询结果如表 1所示, 由于篇幅有限, 我们取前 5个返回结果. 最终的用户满意度如表 2所示.

表 1 具体的返回结果

Table 1 The concrete outcome

IMDB	Web	Proposed Algorithm
1. Einleitung zu Arnold Schoenbergs (1973)	1. Arnold Schwarzenneger DVD 2- Pack	1. Terminator 2- Judgment Day (1991)
2. Benedict Arnold A Question of Honor (2003)	2. Last Action Hero (1993)	2. Commando (1985)
3. Love and Action in Chicago (1999) (V)	3. Commando (1985)	3. True Lies (1994)
4. Mary-Kate and Ashley in Action! (2001)	4. End of Days (1999)	4. Last Action Hero (1993)
5. Demonstrating the Action of the… (1900)	5. Eraser (1996)	5. The Terminator (1984)

表 2　用户平均满意度

Table 2　User's average satisfaction

用户人数	查询关键字数	MDB	Web	Proposed Algorithm
40	400	65%	75%	81%

实验结果表明, 应用本文提出的推荐算法对搜索引擎的返回结果重新排序后, 能得到很高的用户满意度, 为 81%. 因此, 在对视频页面的排名处理上, 该算法是行之有效的.

4　结论

随着推荐系统的系统实时性要求越来越高, 本文提出了一种基于项目的协同过滤推荐算法. 该算法结合了推荐系统和搜索工具进行视频搜索, 并将排名结果与传统的 PageRank 算法进行了对比, 实验结果表明, 本文提出的推荐算法可以大幅度提高用户对搜索结果的满意度, 从而有效解决推荐系统处理大规模数据面临的问题. 本文提出的方法还可应用于音乐、旅行、购物和 Web 搜索等其它领域.

[参考文献] (References)

[1] Aggarwal C C, Wolf J L, Wu K L, et al. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering [C] // Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 1999: 201-212.

[2] Balabanovic M, Shoham Y. Fab: content-based, collaborative recommendation[J]. Communications of the ACM, 1997, 40(3): 66-72.

[3] Basilico J, Hofmann T. Unifying collaborative and content-based filtering[C] // Proceedings of the 21st International Conference on Machine Learning. New York: ACM, 2004: 9.

[4] Claypool M, Gokhale A, Miranda T, et al. Combining content-based and collaborative filters in an online newspaper[J]. ACM SIGIR Workshop on Recommender Systems, 1999, 30(3): 128-136.

[5] DeCoste D. Collaborative prediction using ensembles of maximum margin matrix factorization[C] // Proceedings of the 21st International Conference on Machine Learning. New York: ACM, 2006: 249-256.

[6] Deshpande M. Item-based top-n recommendation algorithms[J]. ACM Transactions on Information Systems, 2004, 22(1): 143-177.

[责任编辑: 丁蓉]