个性化推荐算法设计

赵 亮 胡乃静 张守志

(复旦大学计算机科学系 上海 200433)

(zhaoliu1998@citiz.net)

摘 要 协同过滤技术(collaborative filtering)目前被成功地应用于个性化推荐系统中,但随着系统规模的扩大,它的效能逐渐降低,针对它的缺点,提出了一种高效的个性化推荐算法,它包括维数简化和项集相似性计算两个过程,这种算法在提高精确性的基础上减少了计算耗费,可以较好地解决应用协同过滤技术的推荐系统所存在的稀疏性、扩展性等问题,快速产生精确的个性化推荐结果.

关键词 推荐系统,协同过滤,向量空间,单值分解,相似性

中图法分类号 TP311

ALGORITHM DESIGN FOR PERSONALIZATION RECOMMENDATION SYSTEMS

ZHAO Liang, HU Nai-Jing, and ZHANG Shou-Zhi

(*Deparment of Computer Science*, *Fudan University*, *Shanghai* 200433)

Abstract Collaborative filtering is the most successful technology for building recommendation systems. Unfortunately, the efficiency of these methods decline linearly with the number of users and items . To address these limitations, a high efficient personalization recommendation algorithm is presented, which includes two phases: dimensionality reduction and item-based recommendation methods. This algorithm reduces the computation consumption based on enhancing the accuracy, *etc*. It may solve questions well such as sparsity, scalability. It can create accurate personalization recommendation quickly.

Key words recommendation system, collaborative filtering, vector space, singular value decomposition, similarity

1 推荐系统和协同过滤技术

个性化推荐系统被用来帮助用户在大量的信息中寻找感兴趣的内容,它体现的"个性化"服务目前越来越为商务网站、电子图书馆等众多领域所接受,成为了它们的一个重要的功能,一些论文对此已做了较深入地研究,如文献[1]研究了综合考虑服务器的应用逻辑设计、页面拓扑结构等多个数据源的用户访问模式的分析算法,并产生个性化的结果.迄今为止在个性化推荐系统中,协同过滤技术是应用最成功的技术.协同过滤技术也称为面向用户(user-based)的技术,即协同过滤技术通过分析历史数据,生成与当前用户行为兴趣最相近的用户集,将他

原稿收到日期:2001-06-06;修改稿收到日期:2002-01-08
本课题得到国家自然科学基金重点项目资助(69933010)

们最感兴趣的项作为当前用户的推荐结果,即 *top-N* 推荐. 基于协同过滤技术的推荐过程可分为 3 个阶段:数据表述;发现最近邻居;产生推荐数据集.

在一个典型的基于协同过滤技术的推荐系统中,输入数据通常可以表述为一个 $m \times n$ 的用户-项评估矩阵 $\boldsymbol{R}$,m 是用户数,n 是项数,r_{ij} 是第 i 个用户对第 j 项的评估数值,评估值与项的内容有关,如果项是电子商务中的货品,则表示用户订购与否,例如 1 表示订购,0 表示没有订购;如果项是 Web 文档,则表示浏览与否,用户对它的兴趣度有多高,这样的评估值可以有几个等级,如 1～5 等.

基于协同过滤技术的推荐系统的核心是为一个需要推荐服务的当前用户寻找其最相似的"最近邻居"集(nearest-neighbor),即:对一个用户 u,要产生一个依相似度大小排列的"邻居"集合 $N=\{N_1, N_2, \cdots, N_t\}$, u 不属于 N,从 N_1,到 N_t, $sim(u, N_k)$ 从大到小排列.

图 1 演示了协同过滤中邻居的一种形成过程:当前用户 0 和其它用户之间的相似性被计算,如计算欧几里得距离. 图 1 中与点 0 为中心的 $k=5$ 个最近用户被选择为邻居.

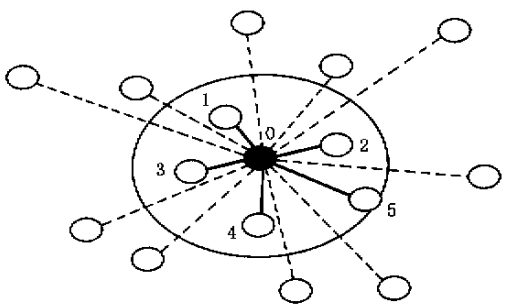

图 1　邻居的形成过程

用户之间的相似性计算有 Pearson 相关度(correlation)计算方法和目前常用的向量空间相似度计算方法.

$$\cos(u_1, u_2) = \frac{u_1 \cdot u_2}{\|u_1\| * \|u_2\|}, \tag{1}$$

式(1)是向量空间相似度计算公式,两个用户 u_1 和 u_2 被看做是向量空间中的两个向量,可以通过计算两个向量的夹角的余弦来衡量相互之间的相似度,夹角越小,相似度越高.

"最近邻居"集产生后,可计算两类推荐结果:用户对任意项的兴趣度和 *top-N* 推荐集.

(1) 用户对项的兴趣度计算

设用户 u 和相应的已选项集 $\boldsymbol{I}_u$,则其对任意项 $t(t \notin \boldsymbol{I}_u)$ 的兴趣度值如式(2)表示:

$$prediction = \bar{u} + \frac{\sum_{i=1}^{n} (corr_i) \times (rating_i - \bar{i})}{\sum_{i=1}^{n} (corr_i)}, \tag{2}$$

$\bar{u}$ 是用户 u 对项的平均评估值,i 是"最近邻居"集的用户. $corr_i$ 是用户 u 和用户 i 之间的 Pearson 系数[2],$rating_i$ 是用户 i 对项 t 的评估值. $\bar{i}$ 是用户 i 对项的平均评估值.

(2) *top-N* 推荐集的产生

为了得到 *top-N* 推荐集,分别统计"最近邻居"集中的用户 i 对不同项的兴趣度,可以用访问频率来衡量,取其中 N 个排在最前面、不属于 $\boldsymbol{I}_u$ 的项作为 *top-N* 推荐集.

尽管协同过滤技术在个性化推荐系统中获得了极大的成功,但随着站点结构、内容的复杂度和用户人数的不断增加,协同过滤技术的一些缺点逐渐暴露出来,主要有:

① 稀疏性(sparsity):在许多推荐系统中,每个用户涉及的信息量相当有限,在一些大的系统如亚马逊网站中,用户最多不过就评估了上百万本书的 1%～2%,造成评估矩阵数据相当稀疏,难以找到相似用户集,导致推荐效果大大降低.

② 扩展性(scalability):"最近邻居"算法的计算量随着用户和项的增加而大大增加,对于上百万之巨的数目,通常的算法将遭遇到严重的扩展性问题.

③ 精确性(accuracy):通过寻找相近用户来产生推荐集,在数量较大的情况下,推荐的可信度随之降低.

为了解决协同过滤技术存在的问题,目前常用聚类分析(clustering)的方法,它或者将"最近邻居"搜索对象限制在最相近的聚类中,或者用聚类的质心提取推荐结果,虽能提高推荐速度,但降低了推荐质量. 并没有从根本上解决问题[2,3]. 本文提出一种高效的个性化推荐算法,它能够快速产生精确的个性化推荐结果.

2　个性化推荐算法的设计

本文提出的个性化推荐算法由两部分组成:维数简化和项集相似性计算. 如第 1 节所述,协同过滤技术的用户-项矩阵的数据表述方法所带来的稀疏

性严重制约了推荐效果,在系统较大的情况下,它既不能精确地产生推荐集,又忽视了数据之间潜在的关系,有必要对这种矩阵表示方式做优化,维数简化就是其中较好的一种方法,本文应用单值分解(singular value decomposition,SVD)技术对用户-项矩阵进行维数简化.

2.1 基于单值分解的维数简化

单值分解是一种矩阵分解技术[4],它可将一个 $m\times n$ 的矩阵 R 分解为 3 个矩阵.

$$R = T_0 S_0 D_0', \quad S_0 = \mathrm{diag}(\sigma_1, \cdots, \sigma_r),$$

其中,$\sigma_1 \geqslant \cdots \geqslant \sigma_r \geqslant 0$,$T_0$ 和 D_0 分别是 $m\times r$ 和 $n\times r$ 的正交矩阵($T_0' T_0 = I$,$D_0' D_0 = I$),r 是矩阵 R 的秩($r \leqslant \min(m,n)$). S_0 是一个 $r\times r$ 的对角矩阵,所有的 σ_r 大于 0 并按照大小顺序排列,称为单值(singular value). 通常对于矩阵 $R = T_0 S_0 D_0'$,T_0,S_0,D_0 必须是满秩的,但单值分解有一个优点,它允许存在一个简化的近似矩阵. 对于 S_0,保留 k 个最大的单值,将其余的用 0 来替代,这样,我们就可以将 S_0 简化为仅有 k 个单值的矩阵($k<r$). 因为引入了 0,可以将 S_0 中的值为 0 的行和列删除,得到一个新的对角矩阵 S,如果矩阵 T_0,D_0 据此简化得到矩阵 T,D,那么有重构的矩阵 $R_k = TSD'$,$R_k \approx R$. 单值分解能够生成初始矩阵 R 的所有秩等于 k 的矩阵中与矩阵 R 最近似的一个.

本文将单值分解应用到推荐系统中,首先将矩阵 R 中评估值为 0 的稀疏项用相关列的平均值代替,即项的平均评估值. 接着将矩阵每行规范化为相同长度,用 $r_{ij} - r_i'$ 代替原来的 r_{ij}(r_i' 是相关列的项的平均评估值). 进行规范化的目的是因为选择不同数量项的用户对相似度计算结果的影响不同,容易造成偏差,规范化为相同长度后,选择项数目较多的用户对相似度计算结果的影响降低了. 经过这样的处理,我们得到矩阵 R',这是算法的输入矩阵,由此,我们提出算法 1.

算法 1. 基于单值分解的的推荐算法

输入:矩阵 R'、用户 U、与之对应的已选项集 I_u.

输出:相关矩阵 T,S,D.

过程:

① 用单值分解方法分解矩阵 R' 得到矩阵 T_0,S_0,D_0.

② 将 S_0 简化为维数为 k 的矩阵,得到 S($k<r$,r 是矩阵 R 的秩).

③ 相应简化矩阵 T_0,D_0 得到 T,D.

④ 计算 S 的平方根得到 $S^{1/2}$.

⑤ 计算两个相关矩阵 $TS^{1/2}$,$S^{1/2}D'$.

$TS^{1/2}$ 是 $m\times k$ 的矩阵,它描述的是用户在 k 维空间中的关系,即用户对 k 个元-项的评估值,可以理解为用户矩阵,矩阵 $S^{1/2}D'$ 大小为 $n\times k$,可以理解为相应的项矩阵. 接下来用这两个简化矩阵来产生推荐结果.

(1) 最近邻居集和 top-N 推荐集的产生

采用向量空间方法计算相似性,这里分析的对象是经过 SVD 分解后的 $m\times k$ 矩阵 $TS^{1/2}$,前面我们提到它描述的是用户在 k 维空间中的关系,因为经过单值分解,大大降低了它的数据稀疏性,可以产生更精确的最近邻居集和相应的 top-N 推荐集.

(2) 用户兴趣度的计算

除了 top-N 推荐集外,还可以计算用户 u 对任意项 t 的兴趣度,因为两个矩阵 $TS^{1/2}$,$S^{1/2}D'$ 的乘积就是规范化后的评估值,则对矩阵 $TS^{1/2}$ 的第 u 行和矩阵 $S^{1/2}D'$ 的第 t 列的内积反规范化,就得到实际的评估值,如式(3):

$$pred_{u,t} = \bar{u} + TS^{1/2}(u) \cdot S^{1/2}D'(t), \qquad (3)$$

$\bar{u}$ 是用户 u 的平均评估值. 与常规的兴趣度计算式(2)相比,式(3)在计算上有相当的简化.

基于维数简化的算法 1 较好地解决了数据稀疏性的问题,同时,因为 $k \ll n$,计算消耗有相应的降低,也有利于解决扩展性问题. 与协同过滤技术一样,基于维数简化的算法也是面向用户的算法,可以提供真正的带有个性化色彩的推荐结果,但随着系统的庞大,其推荐的精确性也逐渐降低,为此,接下来我们用一种基于项相似性的面向模型的技术来对算法 1 进行改进.

所谓面向模型技术(如贝叶斯网络(Bayesion network)[5]等)是相对于协同过滤这样的面向用户技术而言,这种技术首先从数据中抽取出关系描述模型,然后应用这个预先得出的模型进行推荐,这样就能够快速、准确地产生推荐结果. 考虑到用户对与他自己已经选择的项相似或相关的内容更感兴趣,即更能满足精确性的要求,本文的方法分析项之间的相似性,从中提取出关系模型,可将它称为基于项集相似性的推荐算法.

2.2 项集相似性的计算

基于项集相似性的算法分析用户-项矩阵以确定不同项之间的关系,接着用这些关系来产生 top-N 推荐结果,算法如下:

算法 2. 基于项集的推荐算法(产生 $top\text{-}N$ 推荐集)

输入:用户 U、与之对应的已选项集 I_u、推荐用户-项评估矩阵 R.

输出:与 I_u 最相似的 $top\text{-}N$ 推荐集.

过程:

① 对每个项 $j\in I_u$,计算它的 k 个最相似的项 $\{l_1,l_2,\cdots,l_k\}$,记录相应的相似度 $\{S_{l1},S_{l2},\cdots,S_{lk}\}$,合并这些最相似项,得到项集 C.

② 从 C 中删除 I_u 中已经存在的项,得到侯选推荐项集 C'.

③ 对任意项 $C\in C'$,相似度 $sim(C,I_u)=\sum sim(C,j_k),\ j_k\in I_u$.

④ 将 C' 中的项按相似性排列,其中最前的 N 个项作为推荐集.

这个算法关键是要计算不同项之间的相似性.根据推荐系统的特点,本文用条件概率来评估项之间的相似性,相对向量空间模型方法,条件概率方法在这里有更高的灵活性和实际意义.

对于项 u 和 v,用 $P(u/v)$ 表示选择了项 v 的同时也选择项 u 的条件概率,式(4)给出了条件概率的一般计算方式,它可以用来衡量项 u 和 v 之间的相似性.

$$P(u/v)=\frac{Freq(uv)}{Freq(v)},\tag{4}$$

其中,$Freq(X)$ 为选择集合 X 中的项的用户数.

这个公式容易产生一个缺点,往往 $P(u/v)$ 很高,但并不是因为项 u 和 v 有很高的相似性,而是因为项 u 被频繁选择的缘故.一般的解决方法是将 $P(u/v)$ 除上一个与项 u 出现频率有关的数值,如文献[6]直接将 $P(u/v)$ 除以 $Freq(u)$.但这样做存在两个缺陷:首先,因为这样的"缩放"将极大地影响推荐结果,考虑到这样的"缩放"对不同系统的影响也不同,有必要设置可调节的优化"缩放"系数;其次,在文献[6]中,不管用户已经选择的项数目是多是少,他们对项相似度计算的作用是相同的,但实际上,对于选择项数目较少的用户,他所选择的项往往更有参考意义,在计算时应突出它们的作用.为此,可以对文献[6]中的计算公式作如下修订:首先将矩阵 R 的每一行规范化为相同长度,然后得到式(5):

$$sim(v,u)=\frac{\sum_{\forall i,r_{i,v}>0} r_{i,u}}{Freq(v)\times(Freq(u))^{\alpha}},\tag{5}$$

式(5)用用户-项矩阵第 u 列中相应的非零项之和代替了文献[6]中的频率 $Freq(uv)$.矩阵中的行被规范化为相同大小后,选择项数目较多的用户对计算结果的影响被降低了.同时式(5)还设置了"缩放"系数 α,α 是一个 0～1 之间的值,根据不同的推荐系统的数据集要采用不同大小的优化 α 值.

基于项集相似性的推荐算法得到的结果其个性化效果较差(与面向用户的推荐算法相比),但通过分析项之间的相似性,可以产生精确的推荐结果,我们用它来对算法 1 进行改进,得到算法 3.

2.3　优化的个性化推荐算法

个性化推荐算法 3 将维数简化和基于条件概率的项相似性计算的优点结合在一起,首先用单值分解对项-评估矩阵进行简化,接着对以邻居集为基础的子集计算项相似性.

算法 3. 个性化推荐算法.

输入:用户 U、与之对应的已选项集 I_u、推荐用户-项评估矩阵 R.

输出:与 I_u 最相似的 $top\text{-}N$ 推荐集.

过程:

① 去掉矩阵 R 中的稀疏值后规范化,将每行规范化为相同长度,得到矩阵 R'.

② 对矩阵 R' 进行单值分解,计算相应简化的用户矩阵和项矩阵.

③ 分析用户矩阵,采用向量空间计算方法得到邻居集 P.

④ 以邻居集 P 中的用户为基础,得到矩阵 R 的大小为 $m'\times n'$ 的子集 R_s,m' 是邻居集 P 中的用户数.n' 是邻居集 P 中的项数.

⑤ 对 I_u,在 R_s 中计算它的最相似项集 C'_s(用条件概率方法),$C'_s=C_s-I_u$,C_s 是 I_u 中每个项的相似项(k 个)的总和.

⑥ 将 C' 中任意项 C 按 $sim(C,I_u)$ 排列,最前的 N 个项作为 $top\text{-}N$ 推荐集.

3　实验评估

我们用 EachMovie 数据集[7]作为测试数据集来对本文提出的算法 3 与面向用户的协同过滤技术做比较.EachMovie 数据集收集了 1996～1997 年之间 72916 个用户对 1628 个电影的 2456676 个评估值,其矩阵是非常稀疏的.

可以用信息检索领域中评估系统效果的两个标准:召回率($Recall$)和精确率($Precision$),作为对比两种算法执行效率的尺度,首先将数据分为训练集($training\ set$)和测试集($testing\ set$)两部分,训练集用来生成用户-项评估矩阵并得到 $top\text{-}N$ 推荐集,我们将既在 $top\text{-}N$ 推荐集又在测试集中的项的比例作

为 *Recall* 的标准[6]，有

$$Recall = (testing \cap top\text{-}N)/|testing|.$$

为简便起见，我们取前 10000 个记录，引入一个划分系数 x，x 分为 9 个刻度，从 0.1～0.9，相应的训练集是 1000～9000，测试集则是 9000～1000，然后分别计算两种算法的 *Recall* 值(9 次)，经过计算得到数据表 1 和图 2(这里 N 取 10，P 取 300，k 取 20，α 取 0.5，SVD 用 mathlib 计算)：

表 1 本文的优化算法与协同过滤方法计算召回率的数据表

次数	本文提出的优化算法	协同过滤方法
1	0.69	0.50
2	0.66	0.47
3	0.67	0.49
4	0.58	0.43
5	0.62	0.47
6	0.71	0.55
7	0.61	0.45
8	0.62	0.46
9	0.59	0.44

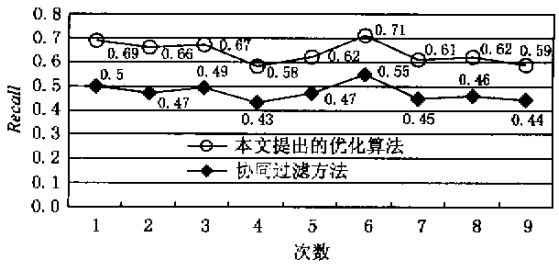

图 2 本文的优化算法与协同过滤方法计算召回率的曲线图

计算其平均值得：

$$\overline{Recall}_{\text{协同过滤}} = 0.473,$$

$$\overline{Recall}_{\text{算法3}} = 0.638$$

可见算法 3 的效果比协同过滤的计算结果性能有 34%的提高. 为了减少计算量，还可以进一步对要测试的用户和项设置一些人为条件，如只取选择项数大于某个阈值的用户和已被一定数量的用户评估过的项等.

算法 3 中存在一些动态的参数，如邻居集 P 的大小、k 和 α 的取值等，它们的取值关系到系统的执行效果，如为了尽可能使潜在的关系不至被遗漏，取的邻居集 P 应适当大些，但又不能过分影响计算效率；不同的系统，最优化的“缩放”系数 α 值也不同，等等，这些都需要动态地确定，我们仍用召回率和精确率，在一个较小的测试集(原始数据集的子集)中动态测试这些参数的优化值，使其尽量达到最优.

仍以上面的数据集为例，图 3 显示 k 值与 *Recall* 的关系：显示推荐的效果在 k 达到一定的大小后，并没有明显提高(在 $k>20$ 后，*Recall* 值几乎走平)，说明对于图例中的数据集，$k=20$ 就可以取为优化值，然后将它应用到原始数据集中去，这对减少计算量有重要意义. 上面计算的提高值根据不同的动态参数有一定的差别，应取最优化值作为计算参数.

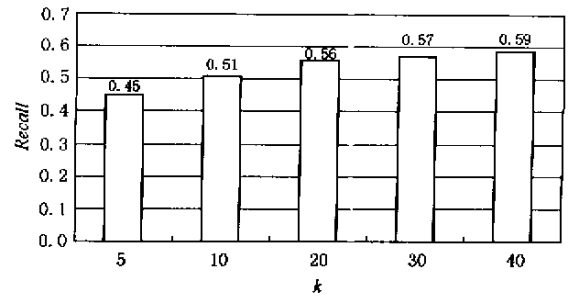

图 3 k 值优化的分析

算法 3 结合了算法 1 和算法 2 二者的优点，通过单值分解，在减少了计算量的同时保持了个性化的优点；通过项相似性计算提高了精确度，还在计算结果中突出了选择项较少的用户的参考意义. 算法 3 能够产生优化的个性化结果，提高推荐质量.

4 结束语

个性化推荐目前被广泛地应用于电子商务、电子图书馆等众多领域，随着系统的不断庞大，原有的推荐算法暴露出许多缺点，本文提出的结合维数简化和项相似性的个性化推荐方法在提高精确性的基础上减少了计算耗费，同时保持了原有的面向用户的方法在个性化方面的优点. 但目前采用测试集作为分析样本的方法不是很优化，下一步将探讨更为有效的方法；同时对算法在不同的测试标准及不同系统中的执行效果作进一步的实验.

参 考 文 献

1 周斌等. 用户访问模式数据挖掘的模型与算法研究. 计算机研究与发展，1999，36(10)：870～875
(Zhou Bin *et al*. On model and algorithms for mining user access patterns. Journal of Computer Research and Development (in Chinese), 1999, 36(10): 870～875

2 J Breese, D Heckerman, C Kadie. Empirical analysis of predictive algorpithms for collaborative filtering. In: Proc of the 14th Conf on Uncertainty in Artificial Intelligence. Madison, WI: Morgan Kaufmann Publisher, 1998. 43～52

3 B Mobasher, H Dai. Integrating web usage and content mining for more effective personalization. In: Proc of the Int'l Conf on E-Commerce and Web Technologies (ECWeb2000). Greenwich, UK, 2000

4 Deerwester S Dumais. Indexing by Latent Semantic Analysis. New York: John Wiley Press, 1990

5 Yung-Hsin Chen, Edward I George. Bayesian model for collative filtering. In: Proc of the IEEE RIDE'97 Workshop. Birmingham, England, 1997

6 Brendan Kitts, David Freed. Cross-sell: A fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities. In: Proc of ACM SIGKDD Int'l Conf. Boston, MA, USA. 2000. 437～446

7 MeJones Paul. EachMovie collaborative filtering data. Princeton, USA, DEC system Research Center. 1997. http://www.research.digital.com/SRC/eachmovie/

赵　亮　男,1971 年生,博士研究生,主要研究方向为数据库与知识库.

胡乃静　男,1971 年生,博士研究生,主要研究方向为数据库与知识库.

张守志　男,1966 年生,博士研究生,主要研究方向为数据库与知识库.