DOI:10.16451/j.cnki.issn1003-6059.2011.02.010

基于 Widrow-Hoff 神经网络的多指标推荐算法*

张付志 常俊风 王 栋

(燕山大学 信息科学与工程学院 秦皇岛 066004)

摘 要 为解决传统的协同过滤推荐算法不能综合运用多个指标进行推荐的问题,通过引入多指标评分的概念对标准的协同过滤推荐算法进行扩展,提出一种基于 Widrow-Hoff 神经网络的多指标推荐算法. 利用 Widrow-Hoff 最小二乘法自适应算法在进行系统辨识时的高精度拟合特性,提出一种基于 Widrow-Hoff 最小二乘法算法的用户偏好特征向量计算方法. 利用用户偏好特征向量和空间距离矩阵度量用户相似度,以定位邻居集并为用户推荐最优项目. 实验结果表明,本文算法可提高推荐精度,改进推荐质量.

关键词 Widrow-Hoff 神经网络,推荐算法,多指标评分,相似度,用户偏好特征向量
中图法分类号 TP 393

Multi-Criteria Recommendation Algorithm Based on Widrow-Hoff Neural Network

ZHANG Fu-Zhi, CHANG Jun-Feng, WANG Dong
(*School of Information Science and Engineering, YanShan University, Qinhuangdao* 066004)

ABSTRACT

To solve the problem that the traditional collaborative filtering recommendation algorithm can not recommend with multiple criteria, a multi-criteria recommendation algorithm based on Widrow-Hoff neural network is proposed by introducing the concept of multi-criteria rating for extending the standard collaborative filtering algorithm. The Widrow-Hoff least mean square (LMS) adaptive algorithm has the characteristics of high accuracy fitting in the process of system identification. Based on that, an approach to compute user preferences eigenvector based on Widrow-Hoff LMS algorithm is proposed. The user preferences eigenvector and spatial distance are adopted to measure user similarity and then a neighbor set for the best recommendations is located. Experimental results show that the proposed algorithm improves the accuracy and the quality of recommendation.

Key Words Widrow-Hoff Neural Network, Recommendation Algorithm, Multiple Criteria Rating, Similarity, User Preference Eigenvector

* 国家重点基础研究发展计划(973 计划)项目(No. 2005CB321902)、河北省自然科学基金项目(No. F2008000877,F2011203219)、教育部科技发展中心网络时代的科技论文快速共享专项研究项目(No. 20091333110011,20101333110013)资助
收稿日期:2010-01-27;修回日期:2010-08-31
作者简介 张付志,男,1964 年生,教授、博士生导师,主要研究方向为智能网络信息处理、网络与信息安全、面向服务计算. E-mail: xjzfz@ysu.edu.cn. 常俊风,女,1986 年生,硕士研究生,主要研究方向为智能网络信息处理、个性化服务. 王栋,男,1986 年生,硕士研究生,主要研究方向为个性化服务、计算机网络.

1 引 言

传统的协同过滤推荐算法都是基于单一指标评分,用户通过对项目的单一指标评分反映对该项目的偏好程度. 然而在一些特定的应用环境下,往往需要综合考虑项目的多个指标评分才能知道用户的具体偏好[1]. 例如在 Zaget Survey[2] 中,为餐厅评分提供 4 个指标: 食物、价格、装饰、服务; 在 YahooMovie[3-4] 中,用户需要对电影的 Story、Direction、Visual、Acting 这 4 个方面评分,以确定对该电影的偏好. 显然,如果只单独考虑用户是否喜欢某个餐厅(或电影) 来为用户做推荐,无法反映用户兴趣,推荐质量不高. 只有将用户对项目的多个指标评分融入到推荐算法中,才能够为用户推荐真实地反映其偏好的项目. 因此传统的协同过滤推荐算法难以满足这种需求.

多指标推荐算法的目标与单指标推荐算法相同,都是寻找对用户最有用的项目[5]. 二者的区别在于多指标推荐将更多的项目信息和用户信息加入到推荐过程中,以产生更加精确的推荐. 这就需要有更多的新技术来实现多指标推荐,目前已有一些关于多指标推荐的相关工作. Wei-Guang Teng 等[2] 提出一种基于多维评分的协同推荐算法,该算法将用户的多个指标评分作为空间中的一个点,当用户输入所需偏好后,根据给定的偏好采用数据查询技术选择所有空间点中指标与该偏好指标最近的一些点作为用户推荐的项目. 但是该算法旨在为用户推荐所需项目,没有考虑用户之间的相关性及推荐的最优化问题. 因此,对于两个偏好差距较大的用户,算法的推荐结果基本接近,没有综合用户多个指标的评分来发现用户的根本偏好. Adomavicius 等[3] 提出一种综合考虑用户多指标评分的推荐算法,并将项目各个指标评分的平均值作为对该项目的综合评分. 虽然不再单一考虑项目评分,但是仍然没有很好地度量用户对各个指标的具体偏好程度,因此推荐质量和推荐效果仍不理想. Lakiotaki 等[4] 提出一种基于多指标分析的推荐算法,他们将多指标决策分析方法应用到推荐系统中,采用改进的效用函数(UTilites Additives, UTA) 计算用户各个指标的效用值,一定程度上提高推荐精度. 但是,随着系统中用户和项目数据的急剧增加,数据出现极端稀疏性[5],严重影响该算法的推荐质量.

本文针对目前的多指标推荐算法在推荐精度方面存在的不足,利用 Widrow-Hoff 最小二乘法(Least Mean Square,LMS) 自适应算法的高精度拟合性优点,提出一种用户偏好特征向量的计算方法,以挖掘更精确的用户偏好并给出一种基于用户偏好特征向量的用户相似度度量方法. 在此基础上,提出一种基于 Widrow-Hoff 神经网络的多指标推荐算法. 同现有的多指标推荐算法相比,本文提出的推荐算法在推荐精度方面有明显提高.

2 自适应算法的传统相似度度量分析

2.1 Widrow-Hoff 最小二乘法自适应算法

图 1 是具有 R 个输入的单层(有 S 个神经元) 线性神经网络结构,其权值矩阵为 w,阈值向量为 b,是一种连续取值的线性加权求和阈值网络,也称为 Madaline 网络.

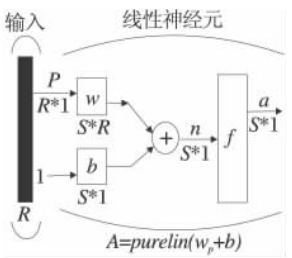


图 1 单层神经网络

Fig. 1 Single-layer neural network

使用 Widrow-Hoff 最小二乘法自适应算法训练网络的权值和阈值,使其线性逼近一个函数式. 首先定义一个线性网络的误差函数[6]:

$$e(w,b) = \frac{1}{2}(t-a)^2 = \frac{1}{2}(t-wp)^2.$$

通过上式可知,线性网络具有抛物面型的误差曲面,因此只有一个误差最小值. 由于该值取决于网络的权值和目标矢量,因此可通过调整权值使均方误差达到最小. Widrow-Hoff 学习规则是通过沿着相对于误差平方和的最速下降方向,连续调整网络的权值和阈值. 根据梯度下降法,权值矢量的修正正比于当前位置上的 $e(w,b)$ 的梯度,对于第 i 个输出节点,则有

$$\Delta w(i,j) = -\mu\frac{\partial e}{\partial w(i,j)} = \mu[t(i)-a(i)]p(i),$$

或

$$\Delta w(i,j) = -\mu\delta(i)p(i),$$
$$\Delta b(i) = \mu\delta(i),$$

其中,μ 是学习率,$\delta(i) = t(i) - a(i)$.

当μ较大时,学习过程加速,网络收敛较快,但μ过大时,学习过程变得不稳定,且误差增大.

Widrow-Hoff 最小二乘法自适应算法的权值变化量正比于网络的输出误差及网络的输入矢量. 该算法不仅简单,而且具有收敛速度快和精度高的优点.

2.2　传统相似度度量方法

度量用户间相似性的方法,目前主要有余弦(Cosine)相似性、相关(Correlation)相似性和修正的余弦(Adjusted Cosine)相似性 3 种[7-9].

1) 余弦相似性. 把用户评分看作 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定为 0,用户间的相似性通过向量间的余弦夹角来度量. 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 i 和向量 j,则用户 i 和用户 j 之间的相似性为

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

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

$$sim(i,j) = \frac{\sum_{C\in I_{ij}}(R_{i,c}-\overline{R}_i)(R_{j,c}-\overline{R}_j)}{\sqrt{\sum_{C\in I_{ij}}(R_{i,c}-\overline{R}_i)^2}\sqrt{\sum_{C\in I_{ij}}(R_{j,c}-\overline{R}_j)^2}},$$

其中,$R_{i,c}$ 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来改善上述缺陷,设经用户 i 和用户 j 共同评分的项目集合用 I_{ij} 表示,I_i 和 I_j 分别表示经用户 i 和用户 j 评分的项目集合,则用户 i 和用户 j 之间的相似性 $sim(i,j)$ 为

$$sim(i,j) = \frac{\sum_{C\in I_{ij}}(R_{i,c}-\overline{R}_i)(R_{j,c}-\overline{R}_j)}{\sqrt{\sum_{C\in I_i}(R_{i,c}-\overline{R}_i)^2}\sqrt{\sum_{C\in I_j}(R_{j,c}-\overline{R}_j)^2}},$$

其中,$R_{i,c}$ 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.

在余弦相似性方法中,将用户未评分的项目假定为 0,但用户对未评分的项目的偏好不一定完全相同,对未评分的项目的评分也不一定完全都为 0. 由于推荐系统中用户和项目数量的剧增,用户对项目的评分数量不超过项目总数量的 1%[9],因此在数据稀疏情况下,余弦相似度不能很好地度量用户的相似度,对于修正的余弦相似度存在同样的问题. 相关相似性根据用户间的共同评分项目度量用户相似度,在用户间共同评分项目充足的情况下,相似度计算能较好地评价用户间的相似度,但如果用户间的共同评分较少,相关相似性就不再适用.

3　基于 Widrow-Hoff 最小二乘法自适应算法的用户偏好特征向量与用户相似度度量

3.1　用户偏好特征向量

定义 1　向量

$$\boldsymbol{F}:\{\boldsymbol{F}\mid \boldsymbol{A}\rightarrow \boldsymbol{F}^*\ [\boldsymbol{B}_1\cdots\boldsymbol{B}_i\cdots\boldsymbol{B}_n]^{\mathrm{T}}+\boldsymbol{b}\}$$

称为用户偏好特征向量,其中 $\boldsymbol{B}_i$ 表示用户对项目 i 的评分向量,$\boldsymbol{A}$ 表示项目 i 的总评分向量,$\boldsymbol{b}$ 为阈值向量.

根据已知用户数据计算 Widrow-Hoff 神经网络(WHNN)中控制稳定性和算法收敛速率的学习率μ使 WHNN 达到优收敛性,其中 $0<\mu<1/x$(x 是 WHNN 输入矩阵的最大特征值). 训练 WHNN 以便得到各个用户关于4个指标story、actiong、direction、visual 的偏好特征向量. 设项目的指标数为 l,用户的输入为 $\boldsymbol{B}_i=[\boldsymbol{B}_{i1}\cdots\boldsymbol{B}_{ik}\cdots\boldsymbol{B}_{il}]^{\mathrm{T}}$,输出为 $\boldsymbol{A}_i$. 则根据 WHNN 训练用户数据使满足

$$\boldsymbol{A}_i\rightarrow \boldsymbol{F}_i^*\ [\boldsymbol{B}_{i1}\cdots\boldsymbol{B}_{ik}\cdots\boldsymbol{B}_{il}]^{\mathrm{T}}+b_i, \tag{1}$$

其中,$i=1,2,\cdots,n$; $k=1,2,\cdots,l$; b_i 为阈值.

给定用户输入向量 $\boldsymbol{B}_i=[\boldsymbol{B}_{i1}\cdots\boldsymbol{B}_{ik}\cdots\boldsymbol{B}_{il}]^{\mathrm{T}}$,采用 WHNN 沿着相对于误差平方和的最速下降方向,连续调整网络的权值和阈值以得到最适宜的 $\boldsymbol{F}$ 使均方误差达到最小. 将偏好特征向量 $\boldsymbol{F}$ 进行初始化为 $\boldsymbol{F}(0)$,在 $\bar{e}$ 的下降方向对 $\boldsymbol{F}(0)$ 作第一修改 $\Delta\boldsymbol{F}(0)$,得

$$\boldsymbol{F}(1)=\boldsymbol{F}(0)+\Delta\boldsymbol{F}(0),$$

到第 $k+1$ 步时为

$$\boldsymbol{F}(k+1)=\boldsymbol{F}(k)+\Delta\boldsymbol{F}(k).$$

在第 $k+1$ 步采用第 k 个训练输入 b_k,得到第 $k+1$ 步的偏好特征向量 $\boldsymbol{F}$ 修改算式为

$$\boldsymbol{F}(k+1)=\boldsymbol{F}(k)+2\mu\delta_k(k)\,b_k, \tag{2}$$

其中 μ 为学习率.

同样,对于阈值的修改算式为

$$b(k+1)=b(k)+2\mu\delta_k(k), \tag{3}$$

其中 μ 为学习率.

综上,计算用户偏好特征向量的步骤如下.

step 1　采用式(1),利用给定的初始偏好特征

向量计算总评分并与原总评分求差.

step 2 通过式(2)、(3)不断调整用户的偏好特征向量.

step 3 使用新调整的偏好特征向量计算下条数据的总评分与原有评分的差值.

下面通过一个例子来说明偏好特征向量的计算过程. 表 1 为用户 User1 对 5 个项目的 4 个指标的偏好和各项目的总偏好的评分情况,本文的评分值范围为 1 – 13 分.

设根据用户偏好得到的预测总评分为 a',实际总评分为 a,两者间的误差为 e. 初始化用户 User1 的偏好特征向量为 $\boldsymbol{F}(0)=(0,0,0,0)$,初始阈值向量为 $\boldsymbol{b}=0$,选择学习率为 $l=0.0004$,已知输入向量 $\boldsymbol{B}(0)=(9,9,9,9)$.

1) 计算 User1 对 Item1 的预测总评分:

$$a'=\boldsymbol{F}(0)*\boldsymbol{B}(0)^{\mathrm{T}}+\boldsymbol{b}(0)$$
$$=(0,0,0,0)(9,9,9,9)^{\mathrm{T}}+0=0,$$

误差 $e=a'-a=9$.

2) 根据式(2)和(3)调整 $\boldsymbol{F}$ 和 $\boldsymbol{b}$:

$$\boldsymbol{F}(1)=\boldsymbol{F}(0)+2*1*e*\boldsymbol{B}(0)$$
$$=(0,0,0,0)+2*0.0004*9*(9,9,9,9)$$
$$=(0.648,0.648,0.648,0.648).$$
$$\boldsymbol{b}(1)=\boldsymbol{b}(0)+2*1*e=0+2*0.0004*9$$
$$=0.072.$$

3) 根据 2) 调整出的 $\boldsymbol{F}$ 和 $\boldsymbol{b}$ 对 Item2 进行计算,即

$$a'=\boldsymbol{F}(1)*\boldsymbol{B}(1)+\boldsymbol{b}(1)$$
$$=(0.648,0.648,0.648,0.648)(9,13,11,12)^{\mathrm{T}}+0.072$$
$$=29.232,$$

误差 $e=a'-a=29.232-11=18.232$.

4) 重复 2)、3),直到误差 e 达到最小.

5) 得到误差最小时的用户偏好特征向量

$$\boldsymbol{F}=f_1,f_2,f_3,f_4.$$

表 1 用户 User1 评分表

Table 1 Rating table of User1

Item \ Rating	Overall	Story	Action	Direction	Visual
1	9	9	9	9	9
2	11	9	13	11	12
3	11	11	12	11	10
4	10	10	12	11	10
5	9	9	9	10	10

根据训练得到的用户对 4 个指标的偏好度可以得到各个用户的偏好特征向量,表 2 是训练得到的部分用户对电影各个指标的偏好度.

表 2 部分用户偏好特征向量中 4 个指标偏好度

Table2 4-criteria preference-value for some users

User \ Interest	Story	Action	Direction	Visual
1	0.200456	0.083257	0.449954	0.284212
2	0.230384	0.187641	0.220274	0.210404
3	0.195774	0.22527	0.218259	0.25609
4	0.280267	0.31436	0.108126	0.226064
5	0.022915	0.405687	0.161842	0.405687
6	0.181353	0.18449	0.188655	0.198964
7	0.274095	0.269314	0.270999	0.268751
8	0.300762	0.207323	0.316076	0.246033
9	0.715984	0.092455	0.131783	0.048616
10	0.409776	0.082984	0.395478	0.107813

采用 WHNN 对已知用户数据进行训练得到各个用户 4 个指标的偏好度,根据用户各自的偏好度所形成的偏好特征向量对用户进行推荐. 通过表 2 可以看出,不同的用户对项目各个指标有不同的偏好. 例如,用户 7 对电影的 4 个指标的偏好度分别为 0.274 095、0.269 314、0.270 999、0.268 751,而用户 9 的兴趣是 Story 指标. 只有明确用户具体的兴趣,才更有利于提高推荐精度和质量.

3.2 基于用户偏好特征向量的相似度度量方法

传统的用户相似度度量方法单纯从项目角度反映用户的偏好,针对具体项目计算具体相似度,而没有挖掘到用户的深层偏好,因此无法反映出用户间对项目的共性认识. 针对传统相似度度量方法的不足,本文采用基于用户对项目的偏好特征向量发现用户对多项目的共性认识,真正意义上根据用户偏好间关系计算用户相似度. 根据 2.1 节中得到的各个用户偏好特征向量,计算用户相似度的公式如下:

$$sim(i,j)=\cos(\boldsymbol{F}_i,\boldsymbol{F}_j)=\frac{\sum_{k=1}^{n}(F_{i,k},F_{j,k})}{\|F_i\|\ \|F_j\|},\quad(4)$$

其中,$F_{i,k}$ 和 $F_{j,k}$ 分别是用户 i 和用户 j 第 k 个指标的偏好度,$\boldsymbol{F}_i$ 是用户 i 的偏好特征向量,$\boldsymbol{F}_j$ 是用户 j 的偏好特征向量,n 值为评测的指标个数.

下面通过实例对比说明基于用户偏好特征向量的用户相似度计算方法的有效性.

已知两个用户 User1 和 User2,表 3 和表 4 分别为二者的评分情况,下面通过评分项目相同和评分项目不同两种情况进行具体分析.

1) 评分项目相同. 表 3 中用户对项目的 Overall

表 3　User1 和 User2 的评分项目相同

Table 3　Ratings of User1 and User2 with same items

Item \ User / Interest	User1					User2				
	Item1	Item2	Item3	Item4	Item5	Item1	Item2	Item3	Item4	Item5
Story	12	12	10	11	13	12	12	10	11	6
Action	13	13	12	12	13	13	13	12	12	6
Direction	13	12	11	11	6	13	12	11	11	13
Visual	11	13	10	10	6	11	13	10	10	13
Overall	12	12	11	11	6	12	12	11	11	6

表 4　User1 和 User2 的评分项目不完全相同

Table 4　Ratings of User1 and User2 with different items

Item \ User / Interest	User1					User2				
	Item1	Item2	Item3	Item4	Item5	Item5	Item6	Item7	Item8	Item9
Story	12	12	10	11	13	6	12	12	10	11
Action	13	13	12	12	13	6	13	13	12	12
Direction	13	12	11	11	6	13	13	12	11	11
Visual	11	13	10	10	6	13	11	13	10	10
Overall	12	12	11	11	6	6	12	12	11	11

评分完全相同,但在部分项目的多个指标中表现出不同的用户偏好,如 Item 5 中,虽然 Overall 评分相同为 6,但是其余 4 个指标的评分分别为 6,6,13,13 和 13,13,6,6 偏好完全不同,此时数据稀疏性很低. 针对这种情况,分别采用余弦相似性、相关相似性和修正的余弦相似性公式和本文提出的基于用户偏好特征向量度量用户相似性的计算方法,计算比较两用户间的相似性,所得结果为表 5 中的 similarity1.

从表 5 可以看出,利用相关相似性、余弦相似性和修正的余弦相似性计算的相似度值均为 1,即两个用户的偏好完全相同,但是从评分数据中各个指标可以看出,两个用户对项目 5 的各个指标偏好并不相同,因此两个用户的相似度不会完全相同. 由于本文提出的相似度度量方法融合用户对项目多个指标的偏好特征,从而挖掘出用户间的共同偏好,因而本文得到的相似度度量方法能够更准确地反映用户间对项目的共性认识.

2) 评分项目不相同. 表 4 中用户对项目的总评分不相同,此时数据的稀疏性很高,共同的评分项目为项目 5,而用户对非共同评分项目的多个指标的偏好趋于一致,因此需要从细节上掌握用户对项目的偏好情况.

针对这种情况,分别采用余弦相似性、相关相似性和修正的余弦相似性公式和本文提出的用户偏好相似性计算方法,计算两个用户间的相似性,所得结果为表 5 中的 similarity2.

表 5　4 种相似性度量方法计算结果的比较

Table 5　Comparison results of 4 similarity measurement methods

计算结果 / 方法	Similarity 1	Similarity 2
余弦相似性	1	0.124
相关相似性	1	−1
修正的余弦相似性	1	−0.631
本文提出的相似度计算方法	0.9989	0.998

从表 5 可以看出,利用相关相似性公式计算的相似度值为 −1,即两个用户无关,但用户间并不是完全无关,而是对很多项目存在共性认识,通过表 5 看出基于用户偏好特征向量的相似度度量方法能够提炼出用户对项目间的共识. 因此,基于用户偏好特征向量的相似度度量方法不仅能够很好的度量用户相似度,而且解决了用户间有共同的偏好但由于缺乏共同评分项目而无法度量相似度的问题,因此这种相似度度量方法能够从细粒度层面发现用户的共同偏好,得到更加精确的相似度值.

每个用户都有各自的偏好特征向量,既不用考虑是否要对未评分项目进行填充以提高用户间相似度,也不用考虑用户间共同评分项目的数量多少. 使用用户偏好特征向量度量用户相似度具有如下特点: 1) 根据用户的偏好特征向量计算相似度能够更准确地定位相似邻居; 2) 克服传统相似度计算方法中侧重相关性而非相似性的弱点; 3) 使得那些缺少

共同评分但有共同偏好的用户可比较;4) 可用于用户评分数据稀疏的情况.

4 基于 Widrow-Hoff 神经网络的多指标推荐算法

4.1 算法思想

基于 Widrow-Hoff 神经网络的多指标推荐算法思想是,首先根据用户数据计算收敛系数 μ,然后根据 WHNN 训练用户数据以得到用户的偏好特征向量,通过得到的偏好特征向量计算用户的相似邻居并进行推荐. 具体步骤如下.

step 1 计算收敛系数. 计算用户输入矩阵 $\boldsymbol{R}$ 的最大特征值 x,依据 $0 < \mu < 1/x$ 原则定位 WHNN 的学习率.

step 2 计算用户的偏好特征向量. 首先使用初始偏好特征向量 $\boldsymbol{F}(0)$ 和输入数据 B_{ik} 计算

$$A_i(0) = \boldsymbol{F}(0) \times B_{ik} + \mu_i(0),$$

将得到的 A_i 与已知项目总评分取差值得

$$\delta(0) = d(0) - A_i(0),$$

其中 d 表示已知用户数据. 根据式(2)、(3) 在 WHNN 中训练得到最优 $\boldsymbol{F}$ 作为用户的偏好特征向量.

step 3 计算用户的相似度. 根据得到的用户偏好特征向量通过式(4) 计算用户相似度.

step 4 产生推荐. 设用户 U 的最近邻居集合用 N_u 表示,则用户 U 对项目 i 的预测评分 $P_{u,i}$ 可以通过用户 U 对最近邻居集合 N_u 中项目的评分得到,计算公式如下:

$$P_{u,i} = \overline{R}_u + \frac{\sum_{n \in N_u} sim(u,n) * (R_{n,i} - \overline{R}_n)}{\sum_{n \in N_u} (|sim(u,n)|)},$$

其中,$\overline{R}_u$,$\overline{R}_n$ 分别表示用户 u 和用户 n 的平均评分,$sim(u,n)$ 表示用户 u 和用户 n 的相似度. $R_{n,i}$ 表示用户 n 对项目 i 的评分. 根据 $P_{u,i}$ 的大小判定为用户推荐哪些项目.

4.2 算法描述

根据上述算法思想,给出算法描述如下.

算法 基于 Widrow-Hoff 神经网络多指标推荐算法

输入 用户评分表 R

输出 用户 U 对测试集项目评分

$$r = \{r_1, r_2, \cdots, r_n\}$$

$R \leftarrow \emptyset$, $\delta := 0$

leanring coefficient $\chi \leftarrow get\ L\ \arg\ est$ Eigenvalue (R)

for each user $U_i \in U$ do

repeat

for each item belongs to user U_i do

$A_i(j) \leftarrow get$ Pr $edictRating\ (F_i(j), R_{ij}(1 \cdots n-1), \mu_{i_{ij}})$

// 计算用户预测值

$\delta_i(j) \leftarrow getSubsiractValue\ (A_i(j), R_{ij}(n))$

// 计算预测值与已知数据误差

if $\delta_i(j) > 0$ then

// 根据误差的不同情况调整偏好特征向量和阈值

$\boldsymbol{F}_i(j+1) \leftarrow AdijustFunction(\boldsymbol{F}_i(j), \chi, \delta_i(j), R_{ij}(1 \cdots n-1), -)$

$\mu_i(j+1) \leftarrow AdijustFunction(\mu_i(j), \chi, \delta_i(j), -)$

else if $\delta_i(j) < 0$ then

$\boldsymbol{F}_i(j+1) \leftarrow AdijustFunction(\boldsymbol{F}_i(j), \chi, \delta_i(j), R_{ij}(1 \cdots n-1), +)$

$\mu_i(j+1) \leftarrow AdijustFunction(\mu_i(j), \chi, \delta_i(j), +)$

else $A_i(j+1) \leftarrow get$ Pr $edictRating\ (\boldsymbol{F}_i(j+1), R_{i(j+1)}(1 \cdots n-1), \mu_{i_{i(j+1)}})$

$\delta_i(j+1)\ getSubsiractValue\ (A_i(j+1), R_{i(j+1)}(n))$

endif

endfor

until δ_i closing to 0

endfor

for each user $U_i, U_j \in U$ do

// 计算用户相似度

$sim(U_i, U_j) \leftarrow getSimilarity(F_i, F_j))$

$TopNeighbour(U_i) \leftarrow Sort(sim(U_i, U_j))$

endfor

for each user $U_i \in U$ do

// 计算相关项目评分

$getRating(U_i) \leftarrow TopNeighbour(U_i)$

endfor

return Rating

算法由 4 部分组成: 第一部分为整个算法计算网络学习率; 第二部分计算用户偏好特征向量,是算法的核心部分,首先将用户数据经过式(1) 计算出预测值与原始值的偏差,再根据式(2)、(3) 和计算出的偏差值,不断调整用户偏好度,使所有用户评分项目预测值的偏差均小于阈值 δ,最终形成各用户偏好特征向量,该部分可以离线计算; 第三部分计算用户相似度并产生相似邻居,根据第二部分计算出的用户偏好特征向量度量用户偏好特征向量间的距离得到用户相似邻居; 第四部分计算项目评分,为用户产生推荐.

5 实验结果及分析

5.1 数据集

本文采用的两类数据集分别来自 YahooMov-

ie[3-4] 站点(www. yahoo. movie. com) 和 MovieLens 站点(http: //MovieLens. umn. edu) . YahooMovie 是一个基于 Web 的推荐系统,用于接收用户对电影的评分并提供相应的电影推荐列表. 目前,该 Web 站点的用户已经超过 43 000 人,用户评分的电影超过 3 500 部. 该数据集中包括用户 id、项目 id、用户对项目的 story、action、direction 和 visual 4 个指标的评分以及对该项目的 Overall 评分数据. 评分值分为 13 个等级: 从 A + 到 F,本文将这 13 个等级转换成 13 -1 分. 为评估所提算法,在数据预处理阶段,要求每个用户至少对 10 部电影评分,每部电影至少被 10 个用户评分. 数据集包括 110 个用户对 45 部电影的共 2 056 个已知评分,数据的稀疏等级[11] 为

$$1-\frac{2056}{(110\times 45)}=0.5846,$$

每个用户的平均评分电影为 18. 7 部,每部电影平均被 45. 7 个用户评分. 每个指标的平均评分大约是 9.

MovieLens 数据集为 943 个用户对 1 682 个项目的 10 万条投票记录,用户评分数据集的稀疏等级为

$$1-\frac{100000}{(943\times 1682)}=0.9370.$$

下载的文件中包括: u. data,为未排序的全部 10 条投票记录; u. item, 为项目信息,包括项目的名称和分类; u. user,为用户信息,包括用户的年龄,性别和职业; u. genre,为项目的分类信息.

本文采用 YahooMovie 数据集的优势在于: 用户对各个项目的 4 个指标存在各自评分,我们可以根据用户对每个项目的 4 个指标的偏好发现用户的具体偏好; 而 MovieLens 数据集中缺乏用户对项目细节的偏好,因此对于挖掘用户对项目深层次的偏好力度不够. 但是为体现本文算法的有效性,将 MovieLens 数据集的 u. user 文件中的 age, sex, occupation 作为度量用户偏好的 3 个指标,根据需要我们将这 3 个指标按类别划分后进行数字量化,便于后期实验所用.

5.2　评价指标

推荐质量的评价标准主要有统计精度度量和决策支持精度度量两类[8,12] . 统计精度度量方法中常用平均绝对偏差(Mean Absolute Error, MAE) 指标; 决策支持精度度量中常用召回率(Recall) [12] 和准确率(Precision) [13] 等指标.

1) 平均绝对偏差 MAE 通过计算预测的用户评分与实际的用户评分之间的偏差度量预测的准确性,MAE 越小,推荐质量越高. 设预测的用户评分集合表示为 $\{p_1,p_2,\cdots,p_N\}$,对应的实际用户评分集合为 $\{q_1,q_2,\cdots,q_N\}$,则平均绝对偏差定义为[8,12]

$$MAE=\frac{\sum_{i=1}^{N}|p_i-q_i|}{N}.$$

2) 召回率(Recall) 反映待推荐项目被推荐的比率,准确率(Precision) 表示算法推荐成功的比率. 召回率和精确率是一对矛盾,当 top-N 个数增加时,召回率增加但推荐精度降低. 召回率和精确率对推荐结果同等重要,推荐质量越高,F-measure 值越高. 因此,通过使用相同的权重将两者结合起来 F-measure[11] :

$$F-measure=\frac{2*\ recall*\ precision}{recall+precision}.$$

5.3　实验及其分析

为在小型数据集下得到可信的实验结果,本实验采用 10 折交叉验证技术,随机将数据分为 10 组不相关的子集,将其中的 1 组作为测试集,剩余的 9 组作为训练集. 将实验重复进行 10 次,最终 MAE 值由 10 次 MAE 值算术平均得到,最终 Time(由网络训练到计算得到 MAE 值的总时间) 值由 10 次 Time 值算术平均得到. 为选取最优的目标用户邻居数目,将 MAE 值和 Time 作为参数进行比较,实验结果如表 6 所示.

表 6　选取最优邻居个数

Table 6　Selecting optimal number of neighbors

Neighbor \ Value	MAE	Time/s
3	0. 52476	23. 96
7	0. 21473	30. 15
11	0. 20816	97. 44
15	0. 19894	130. 76
19	0. 19732	182. 61

在该实验中,分别设定邻居个数为 3、7、11、15、19,对所提算法的的 MAE 值和 Time 值进行比较,以确定最优邻居个数. 表 6 中分别记录各种邻居个数下的 MAE 值和 Time 值,通过表 6 可以看出随着邻居个数的增加,实验中 MAE 值依次减小,但是从邻居个数 3 到邻居个数 7 变化时,MAE 值变化最大,而从邻居个数 15 开始 MAE 值变化不明显. 通过观察 Time 值发现,随着邻居个数的增加运行时间逐渐增大,而运行时间必须能够满足用户需求,期望运行时间越小越好. 因此,需要综合考虑 MAE 值和 Time 值,不仅要使 MAE 值较小而且要求 Time 值可接受. 根据表 6 可以发现邻居个数为 7 时达到最优. 下文

中在没有明确强调实验邻居个数的条件下，默认选择邻居个数为 7.

5.3.1 基于 F-measure 指标的算法评价

本文将算法所推荐项目按评分分为 4 个等级：等级 1：大于等于 9 分；等级 2：大于等于 10 分；等级 3：大于等于 11 分；等级 4：大于等于 12 分. 图 2 是各评分等级随邻居个数变化，F-measure 的变化情况. 其中，邻居个数从 3 增加到 19，间隔为 4.

从图 2 可以看出，随着邻居个数的不断增加，各个等级的 F-measure 值呈上升趋势，在默认邻居个数 7 时，等级 1 的 F-measure 值可达 0.93 648，具有较高的 F-measure 值；在邻居个数为 19 时，等级 1 的 F-measure 值可高达 0.94. 本文提出的算法在等级 1 的 F-measure 值平均为 0.936 613，等级 2 的 F-measure 值平均为 0.891 769，即使在数据要求很高的等级 4，其 F-measure 均值可达到 0.747 486，因此本文算法具有很高的 F-measure 值.

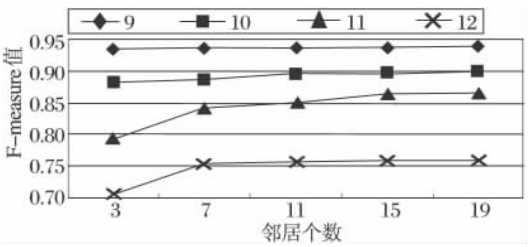


图 2 不同等级的 F-measure 比较

Fig. 2 Comparison of F-measure at different levels

5.3.2 基于 Kendall 相关系数的算法评价

Kendall [14] 相关系数通过度量预测数据与测试集之间排序的相关度来评价推荐算法的质量，即

$$\tau = \frac{2p}{\frac{1}{2}n(n-1)} - 1 = \frac{4p}{n(n-1)} - 1,$$

其中，n 表示项目数量，p 表示两排序一致对的数量. Kendall 相关系数在 -1 和 1 之间变化，如果预测数据与测试集排序完全相同则该值为 1，完全相反则为 -1.

我们对测试集中各用户的项目评分进行排序，然后将用户对项目的总评分的预测结果与其进行排序对比，根据 Kendall 相关系数的定义得到本文提出的基于 Widrow-Hoff 神经网络多指标推荐算法（Widrow-Hoff Neural Network，WHNN）、多指标评分的协同过滤推荐算法（Multi-Criteria Rating Collaborative Filtering，MRCF）[3] 的 Kendall_tau 值比较结果如图 3 所示.

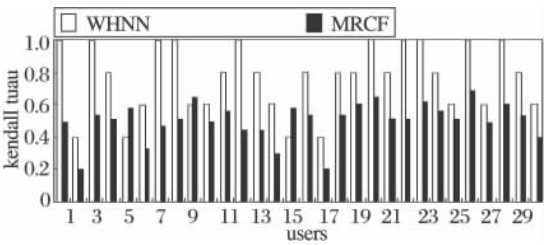


图 3 2 种算法 Kendall_tau 值对比

Fig. 3 Comparison of Kendall_tau between 2 algorithms

表 7 为算法 WHNN 和 MRCF 的 Kendall_tau 值比较结果. 从表 7 可以看出，提出的算法不仅在 Kendall_tau 的平均值上有很大优势，而且有 33.3% 的用户的 Kendall_tau 值为 1，所有用户的 Kendall_tau 值均大于 0.4. 因此，本文算法中用户评分预测排序与已知用户预测排序有很高的一致性.

表 7 2 种算法的 Kendall_tau 对比

Table 7 Comparison of Kendall_tau between 2 algorithms

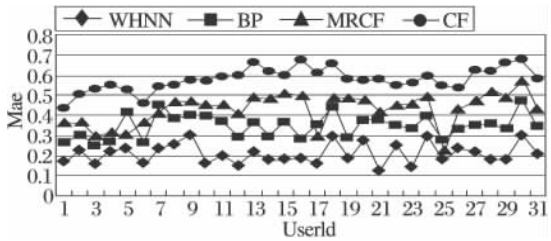
%

Value / Method	average	=1	≥0.8	≥0.6	≥0.4	≥0.2
WHNN	0.766667	33.33	64	86.67	100	100
MRCF	0.500259	0	0	13.33	86.67	100

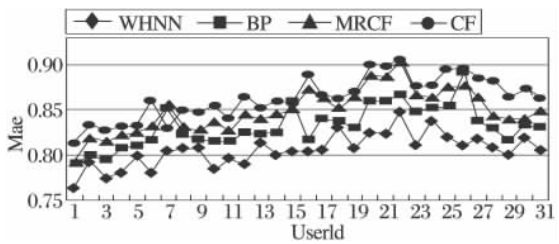
5.3.3 各算法 MAE 值比较

图 4 中（a）、（b）分别是在 YahooMovie 数据集和 MovieLens 数据集上的随机选取 30 个用户进行的基于本文提出的基于 Widrow-Hoff 神经网络多指标推荐算法（WHNN）、多指标评分的协同过滤推荐算法（MRCF）[3] 以及单指标协同过滤推荐算法（Collaborative Filtering，CF）[10] 以及基于 BP（Back Propagation）神经网络的协作过滤推荐算法 [15] 的 MAE 比较.

通过图 4（a）可以看出少量用户在采用 WHNN 算法、BP 算法或者 MRCF 算法时得到的 MAE 值较接近，例如 UserId 为 4 或者 25 的用户. 在采用 BP 算法和 MRCF 算法时大部分用户 MAE 值较接近，整体在 YahooMovie 数据集上的 MAE 值保持在 0.1 ~ 0.7 之间，WHNN 的 MAE 值波动范围为 0.12 ~ 0.3，BP 的 MAE 值波动范围为 0.247 ~ 0.47，而 MRCF 的 MAE 值波动范围为 0.227 ~ 0.57 和 CF 的波动范围为 0.431 ~ 0.67，因此从图 4 的（a）中可以发现对于随机选择的 30 个用户 WHNN 算法具有较优的推荐精度，各算法总体 MAE 值比较见表 8.



(a) YahooMovie



(b) MovieLens

图 4　2 数据集下 4 种算法 MAE 值比较

Fig. 4　Comparison of MAE values among 4 algorithms on 2 datasets

图 4 中(b)是基于 MovieLens 数据集进行的实验,MAE 值的整体波动范围为 0.75 – 0.93,比基于 YahooMovie 数据集上的波动范围大 0.45 左右,原因在于 YahooMovie 的数据稀疏等级小于 MovieLens 的稀疏等级,导致对于公共评分项较少的用户偏好准确度量度较低. 由于采用 MovieLens 数据集时我们将用户信息作为评价的多指标,因此用户在评价的多部电影中指标影响差距不大,导致各算法的精度差距不明显,部分用户的 MAE 值较接近,波动范围较小,各算法总体 MAE 值比较见表 8.

表 8　4 种算法在 2 种数据集上总体 MAE 值比较

Table 8　Comparison of overall MAE values among 4 algorithms on 2 datasets

Method / Data	WHNN	BP	MRCF	CF
Yahoomovie	0.206335	0.344537	0.429549	0.576582
MovieLens	0.805057	0.831433	0.848688	0.862771

表 8 为基于两种数据集上的各算法的总体 MAE 值比较,通过表 8 可以看出实验采用 YahooMovie 数据集获得的总体 MAE 值要小于采用 MovieLens 数据集获得的总体 MAE 值,并且在 YahooMovie 上各算法的 MAE 值差距精度大约为 0.1,而采用 MovieLens 时各算法的 MAE 值差距精度大约为 0.02,在两数据集上各算法的总体 MAE 值呈递增关系. 实验表明在两种不同的数据集上 WHNN 算法都能有效地提高推荐精度.

6　结 束 语

多指标推荐算法是推荐系统的一个新的研究热点,本文在这方面进行一些有益的探索,提出一种基于 Widrow-Hoff 神经网络的多指标推荐算法,给出了用户偏好特征向量的计算方法及用户相似度度量方法. 该算法融合用户对项目的多个指标的评分,通过度量用户间偏好特征向量的相似性,不仅可以更精确地发现用户的偏好,而且对那些共同评分项目较少的用户之间也可以进行比较,提高了推荐精度. 如何利用用户的上下文信息为用户提供更精确的推荐,将是下一步要研究工作.

参　考　文　献

[1] Adomavicius G, Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans on Knowledge and Data Engineering, 2005, 17(6): 734 – 749

[2] Teng Weiguang, Lee H H. Collaborative Recommendation with Multi-Criteria Ratings. Journal of Computers. 2007, 17(4): 69 – 78

[3] Adomavicius G, Kwon Y O. New Recommendation Techniques for Multicriteria Rating Systems. IEEE Intelligent Systems, 2007, 22(3): 48 – 55

[4] Lakiotaki K, Tsafarakis S, Matsatsinis N. UTA-Rec: A Recommender System Based on Multiple Criteria Analysis // Proc of the ACM Conference on Recommender Systems. Lausanne, Switzerland, 2008: 219 – 226

[5] Huang Zan, Chen H, Zeng D. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. ACM Trans on Information System, 2004, 22(1): 116 – 142

[6] Ge Lei, Huo Aiqing. Application Research of Widrow-Hoff Neural Network Learning Rule. Electronic Design Engineering, 2009, 17(6): 15 – 16,19 (in Chinese)
(葛 蕾,霍爱清. Widrow-Hoff 神经网络学习规则的应用研究. 电子设计工程, 2009, 17(6): 15 – 16,19)

[7] Zhang Guangwei, Li Deyi, Li Peng, *et al*. A Collaborative Filtering Recommendation Algorithm Based on Cloud Model. Journal of Software, 2007, 18(10): 2403 – 2411 (in Chinese)
(张光卫,李德毅,李 鹏,等. 基于云模型的协同过滤推荐算法. 软件学报, 2007, 18(10): 2403 – 2411)

[8] Deng Ailin, Zhu Yangyong, Shi Bole. A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction. Journal of Software, 2003, 14(9): 1621 – 1628 (in Chinese)

(邓爱林,朱扬勇,施伯乐. 基于项目评分预测的协同过滤推荐算法. 软件学报, 2003, 14(9): 1621 – 1628)

[9] Sarwar B, Karypis G, Konstan J, *et al*. Item-Based Collaborative Filtering Recommendation Algorithms // Proc of the 10th International Conference on World Wide Web. Hong Kong, China, 2001: 285 – 295

[10] Zhang Binqi. A Collaborative Filtering Recommendation Algorithm Based on Domain Knowledge. Computer Engineering. 2005, 31(21): 79 – 85 (in Chinese)
(张丙奇. 基于领域知识的个性化推荐算法研究. 计算机工程, 2005, 31(21): 79 – 85)

[11] Xu Hailing, Wu Xiao, Li Xiaodong, *et al*. Comparison Study of Internet Recommendation System. Journal of Software, 2009, 20(2): 350 – 362 (in Chinese)
(许海玲,吴 潇,李晓东,等. 互联网推荐系统比较研究. 软件学报, 2009, 20(2): 350 – 362)

[12] Ma Hongwei, Zhang Guangwei, Li Peng. Survey of Collaborative Filtering Algorithms. Journal of Chinese Computer Systems, 2009, 30(7): 1282 – 1288 (in Chinese)
(马宏伟,张光卫,李 鹏. 协同过滤推荐算法综述. 小型微型计算机系统, 2009, 30(7): 1282 – 1288)

[13] Goldbreg K, Roeder T, Gupta D, *et al*. Eigentaste a Constant Time Collaborative Filtering Algorithm. Information Retrieval, 2001, 4(2): 133 – 151

[14] Silvestri F, Baragla R, Palmerini P, *et al*. On-Line Generation of Suggestions for Web Users // Proc of the International Conference on Information Technology: Coding Computing. Las Vegas, USA, 2004, Ⅰ: 392 – 397

[15] Zhang Lei, Chen Junliang, Meng Xiangwu, *et al*. BP Neural Networks-Based Collaborative Filtering Recommendation Algorithm. Journal of Beijing University of Posts and Telecommunications, 2009, 32(6): 42 – 46 (in Chinese)
(张 磊,陈俊亮,孟祥武,等. 基于 BP 神经网络的协作过滤推荐算法. 北京邮电大学学报, 2009, 32(6): 42 – 46)

* *

2011 年全国开放式分布与并行计算学术年会
http://grid.hust.edu.cn/dpcs2011
征 文 通 知

由中国计算机学会开放系统专业委员会主办、华中科技大学计算机学院承办的"2011 全国开放式分布与并行计算学术年会(DPCS2011)"将于 2011 年 8 月 16 – 19 日在湖北恩施召开。本次大会接收中英文投稿。录用的英文文章将由 IEEE 出版,EI 检索,优秀论文推荐到 SCI 国际期刊;录用的中文论文将以正刊方式发表在《微电子学与计算机》,优秀论文推荐到一级学报发表。欢迎大家积极投稿。有关征文事宜通知如下。

一、征文范围(但不限于)

- **开放式分布与并行计算模型、体系结构、编程环境、算法及应用;**
- **开放式网络、数据通信、网络与信息安全、业务管理技术;**
- **开放式海量数据存储与 Internet 索引技术,分布与并行数据库及数据/Web 挖掘技术;**
- **开放式网格计算、云计算、Web 服务、P2P 网络及中间件技术;**
- **开放式无线网络、移动计算、传感器网络与自组网技术;**
- **分布式人工智能、多代理与决策支持技术;**
- **开放式虚拟现实技术与分布式仿真;**
- **开放式多媒体技术与流媒体服务,媒体压缩、内容分送、缓存代理、服务发现与管理技术。** (下转第 254 页)