计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
43(4): 667~672, 2006

使用 BP 神经网络缓解协同过滤推荐算法的稀疏性问题

张 锋 常会友
(中山大学信息科学与技术学院 广州 510275)
(mrzf@163.com)

Employing BP Neural Networks to Alleviate the Sparsity Issue in Collaborative Filtering Recommendation Algorithms

Zhang Feng and Chang Huiyou
(*School of Information Science and Technology, Sun Yat-sen University, Guangzhou* 510275)

Abstract Poor recommendation quality is one major challenge in collaborative filtering recommender systems. Sparsity of source data sets is one major reason causing the poor quality. The popular singular value decomposition techniques and the agent-based methods to a certain extent are able to alleviate this issue. But at the same time they also introduce new problems. To reduce sparsity, a novel collaborative filtering algorithm is designed, which firstly selects users whose non-null ratings intersect the most as candidates of nearest neighbors, and then builds up backpropagation neural networks to predict values of the null ratings in the candidates. Experiments are conducted based on standard dataset. The results show that this methodology is able to increase the accuracy of the predicted values, resulting in improving recommendation quality of the collaborative filtering recommendation algorithm.

Key words electronic commerce; data mining; recommender system; collaborative filtering; backpropagation neural network; algorithm

摘 要 推荐质量低是协同过滤推荐技术面临的主要难题之一. 数据集的极端稀疏是造成推荐质量低的主要原因之一. 常见的降维法和智能 Agent 法虽然某种程度上能缓解这个问题, 但会导致信息损失和适应性等问题. 设计了一个新的协同过滤算法, 根据用户评分向量交集大小选择候选最近邻居集, 采用 BP 神经网络预测用户对项的评分, 减小候选最近邻数据集的稀疏性. 该算法避免了降维法和智能 Agent 法的缺点, 而且实验结果表明, 该方法能提高预测值的准确度, 从而提高协同过滤推荐系统的推荐质量.

关键词 电子商务; 数据挖掘; 推荐系统; 协同过滤; BP 神经网络; 算法
中图法分类号 TP18; TP311.13

1 引 言

随着电子商务的蓬勃发展, 推荐系统被越来越广泛地应用于电子商务网站中, 扮演传统商业中销售人员的角色: 向顾客提供商品信息和建议, 模拟销售人员帮助顾客完成购买过程.

推荐系统使用的技术, 大致上可以分成 3 类: 基于内容的、基于协同过滤技术的和基于两种技术混合型的[1]. 第 1 类通过比较项(商品)之间的相似性实现推荐; 第 2 类根据目标顾客和历史顾客的购买行为特征相似性进行推荐; 第 3 类则是两类技术的

收稿日期: 2005-04-24; 修回日期: 2005-11-28
基金项目: 广东省自然科学基金重点项目(05100302)

综合应用.

协同过滤技术通常使用的是用户-项目评分数据集, 它根据一定的量度标准在评分数据集中找出目标顾客的"最近邻居", 然后参考这些"最近邻居"的"意见", 采用某些技术产生推荐.

协同过滤技术在取得了一定程度成功的同时, 也面临着两个瓶颈问题: 其一, 系统的扩展性差; 其二, 推荐质量不高. 其中数据的稀疏性[2]是导致推荐质量不高的一个主要原因[2~4].

2 相关工作

在解决推荐系统的稀疏性问题上, 一种常见的方法是使用缺省值 0、均值或者其他来填充空评分项[3]. 但这种方法, 把问题过分简单化了, 因为用户对未评分商品的评价不可能完全一样. 另一种比较成功的技术是使用矩阵因数分解技术(matrix factorization technique)——奇异值分解(singular value decomposition). 这种方法通过减少用户-项目评分矩阵的维数达到降低矩阵稀疏性的目的, 从而提高最近邻居质量; 同时 SVD 还能捕捉用户和项之间的隐藏关系从而解决所谓的同义词问题[5]. 但降维会导致信息损失, 因此这种方法对推荐质量的影响不一定都是正面的[6]. 另外一类解决稀疏性问题的研究成果是在系统中嵌入一种半智能化(semi-intelligent filtering)的智能体(agent)或类似的自动关联评分系统, 使用句法特征来预测用户对未评分项目的评分[7~9]. 但同时这会增加系统负担, 应用领域会受到限制[10].

针对推荐系统扩展性问题, 学界也有广泛的研究成果, 像聚类[4], 维数简化和项集相似性计算[11], RecTree 等[10].

本文着眼于推荐系统的推荐质量问题. 针对协同过滤数据集极端稀疏的情况, 设计了一个新的算法, 使用 BP 神经网络对缺失评分进行预测, 提高寻找最近邻居的准确度, 从而提高推荐质量.

3 算法描述

3.1 相关定义

定义 **1**. 用户-项目评分矩阵

$$\boldsymbol{UI}=\begin{bmatrix} ui_{11} & ui_{12} & ui_{13} & \cdots & ui_{1m} \\ ui_{21} & ui_{22} & ui_{23} & \cdots & ui_{2m} \\ ui_{31} & ui_{32} & ui_{33} & \cdots & ui_{3m} \\ \vdots & \vdots & \vdots & & \vdots \\ ui_{n1} & ui_{n2} & ui_{n3} & \cdots & ui_{nm} \end{bmatrix}$$

其中, 每个 u_{ij}($1\leqslant i\leqslant n$, $1\leqslant j\leqslant m$)对应用户 i 对项目 j 的评分.

$$\boldsymbol{UI}=[\boldsymbol{U}_1\ \boldsymbol{U}_2\ \boldsymbol{U}_3\ \cdots\ \boldsymbol{U}_n]^{\mathrm{T}}$$

其中, $\boldsymbol{U}_i$($1\leqslant i\leqslant n$)是用户 i 的评分向量(行向量) $\boldsymbol{U}_i=(ui_{i1}, ui_{i2}, ui_{i3}, \cdots, ui_{im})$.

$$\boldsymbol{UI}=[\boldsymbol{I}_1\ \boldsymbol{I}_2\ \boldsymbol{I}_3\ \cdots\ \boldsymbol{I}_m],$$

其中, $\boldsymbol{I}_j$($1\leqslant j\leqslant m$)是项目评分向量(列向量) $\boldsymbol{I}_j=(ui_{1j}, ui_{2j}, ui_{3j}, \cdots, ui_{nj})^{\mathrm{T}}$.

定义 **2**. ε 算子:

$$\varepsilon(\boldsymbol{V}, u)=\begin{cases}\text{true}, & u\in\boldsymbol{V}, u\neq\varnothing,\\ \text{false}, & u\in\boldsymbol{V}, u=\varnothing,\end{cases}$$

$\boldsymbol{V}$ 是任意一类评分向量(行向量或列向量或其子集). 下同.

定义 **3**. θ 算子:

$$\theta(\boldsymbol{V})=\{u\mid\varepsilon(\boldsymbol{V}, u)=\text{ture}\},$$

显然 $\theta(\boldsymbol{V})$是 $\boldsymbol{V}$的一个子集.

定义 **4**. 用户-项目评分矩阵的稀疏度[2]:

$S=1-$评分矩阵中值非空元素数/评分矩阵中元素总数.

定义 **5**. ω 算子和 ϕ 算子:

$$\omega(\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_n)=\theta(\boldsymbol{V}_1)\cup\theta(\boldsymbol{V}_2)\cup\cdots\cup\theta(\boldsymbol{V}_n),$$

$$\phi(\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_n)=\theta(\boldsymbol{V}_1)\cap\theta(\boldsymbol{V}_2)\cap\cdots\cap\theta(\boldsymbol{V}_n).$$

定义 **6**. 向量空间对齐操作算子 σ. $\boldsymbol{V}=[\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_n]$, 其中任意两向量 $\boldsymbol{V}_i$ 和 $\boldsymbol{V}_j$ 长度可能不相等($1\leqslant i, j\leqslant m$).

$\sigma(\boldsymbol{V})=\sigma([\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_n])=[\boldsymbol{V}_1', \boldsymbol{V}_2', \cdots, \boldsymbol{V}_n']$, 其中, $\boldsymbol{V}_i'=\{v\mid v\in\theta(\boldsymbol{V}_i)$, 其值为 v 的值$\}\cup\{v\mid v\notin\theta(\boldsymbol{V}_i)$ and $v\in\omega(\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_{i-1}, \boldsymbol{V}_{i+1}, \cdots, \boldsymbol{V}_n)$, 值为空$\}$.

如定义 6 所示, 经过 σ 运算后, 向量空间$[\boldsymbol{V}_1', \boldsymbol{V}_2', \cdots, \boldsymbol{V}_n']$ 中的任意两向量长度都相等, 向量元素值为空或者保留对应向量空间 $\boldsymbol{V}$ 中向量元素值.

以上所有运算, 除了 σ 运算外, 都不保留向量或向量空间中元素的值.

3.2 相似性度量

协同过滤技术中一个重要的步骤是计算目标用户和候选邻居之间的相似度, 生成最近邻居集, 进而产生推荐. 用户相似性的度量标准(metrics)主要有余弦法、修正余弦法和基于相关的相似性度量等[3].

余弦法计算相似度:

$$\cos(u, v) = \frac{\sum_{i \in \phi(u, v)} (R_{u,i} \times R_{v,i})}{\sqrt{\sum_{i \in \theta(u)} (R_{u,i})^2} \sqrt{\sum_{i \in \theta(v)} (R_{v,i})^2}}, \tag{1}$$

由 ϕ 算子和 θ 算子的定义, 知 $\phi(u, v)$是用户 u, v 共同评分项目集, $\theta(u)$, $\theta(v)$分别是用户 u, v 的评分项目集. $R_{u,i}$, $R_{v,i}$分别是用户 u、用户 v 对项目 i 的评分.

修正余弦法计算相似度:

$$sim(u, v) = \frac{\sum_{i \in \phi(u, v)} (R_{u,i} - R_u)(R_{v,i} - R_v)}{\sqrt{\sum_{i \in \theta(u)} (R_{u,i} - R_u)^2} \sqrt{\sum_{i \in \theta(v)} (R_{v,i} - R_v)^2}}, \tag{2}$$

R_u 和 R_v 分别表示用户 u 和用户 v 对已评分项目评分的算术平均值, 其他符号的意义同式(1).

文献[3] 的研究表明, 修正余弦法性能表现较好, 所以我们采用式(2)计算用户 u, v 的相似性.

3.3　用户-项评分预测

计算得到目标用户 u 的最近邻居集 NBS_u 后, 则用户 u 对项 i 的预测评分 $P_{u,i}$可通过如下公式计算[4-7]:

$$P_{u,i} = R_u + \frac{\sum_{v \in NBS_u} sim(u, v) \times (R_{v,i} - R_v)}{\sum_{v \in NBS_u} sim(u, v)}, \tag{3}$$

其中符号的含义同式(1)(2).

3.4　算　法

我们的算法和普通的协同过滤算法之间的差异主要在于寻找目标用户的最近邻居这一环节的不同. 所以只给出这一环节的算法.

计算 $\boldsymbol{U}_i$ 最近邻居的方法:

输入: $\boldsymbol{U}_i$ 为用户 i 的评分向量, $\boldsymbol{U}_i \subset \boldsymbol{UI}$;

α 为最近邻居候选集元素个数;

β 为候选最近邻居评分集最小稀疏度;

γ 为 $\boldsymbol{U}_i$ 的最近邻居个数;

输出: $\boldsymbol{U}_i$ 最近邻居集 $NBS_{\boldsymbol{u}_i}$.

① $\boldsymbol{U}_i' = \theta(\boldsymbol{U}_i)$, $\boldsymbol{S} = [\,]$, $\boldsymbol{S}' = [\,]$, $\boldsymbol{S} = \boldsymbol{S} \cup \boldsymbol{U}_i$ $\boldsymbol{S}' = \boldsymbol{S}' \cup \boldsymbol{U}_i'$, $\boldsymbol{V} = \boldsymbol{U}_i$;

② 从 $\boldsymbol{UI} - \boldsymbol{S}$ 中选取 $\boldsymbol{U}_j$, 使得$\frac{|\phi(\boldsymbol{V}, \boldsymbol{U}_j)|}{|\omega(\boldsymbol{V}, \boldsymbol{U}_j)|}$最大, $\boldsymbol{U}_j' = \theta(\boldsymbol{U}_j)$, $\boldsymbol{S} = \boldsymbol{S} \cup \boldsymbol{U}_j$, $\boldsymbol{S}' = \boldsymbol{S}' \cup \boldsymbol{U}_j'$;

③ $|\boldsymbol{S}| > \alpha$? 是, 跳到步骤⑤;

④ 取 $\boldsymbol{S}$ 中新加入(按照加入顺序, 先入先选取)的元素 $\boldsymbol{S}_i \Rightarrow \boldsymbol{V}$, 跳到步骤②;

⑤ $\boldsymbol{S}' = \sigma(\boldsymbol{S}')$,

1) 选择将要评分的空项;

2) 建立神经网络;

3) 预测空项值;

4) $\boldsymbol{S}'$的稀疏度$<\beta$, 则向下执行步骤⑥, 否则跳到步骤 1);

⑥ 在 $\boldsymbol{S}'$中按照修正余弦法求 $\boldsymbol{U}_i$ 的 γ 个最近邻居, 得到最近邻居集 NBS_{u_i}.

对于步骤②和步骤⑤, 解释如下:

步骤②基于这样的常识: 进行协同过滤计算, 在评分矩阵极端稀疏的情况下, 用户-项目评分向量交集很小(甚至为空集)的用户, 可以认为是不相似的. 因此找出评分交集相对比较大的用户组成最近邻居候选集是合理的. 这时, 构成候选集的评分矩阵还是稀疏的, 为了更准确地找到目标用户的最近邻居, 在步骤⑤中, 对这个矩阵的未评分项目值建立 BP 神经网络进行预测. 理由主要有两点: BP 神经网络对复杂的输入输出关系有比较强大的学习和建模能力, 能够有效地处理非完整信息, 后者对候选用户评分矩阵中仍有比较多的未评分项是非常重要的.

下面我们通过一个简化了的具体例子来说明 BP 神经网络预测空值项的过程.

设经过 σ 运算后的$\boldsymbol{S}'$用图 1 所示, 其中 u_i 表示第 i 个用户, i_j 表示第 j 个项, 相应表格单元内的值就是用户对项的评分.

User	Item					
	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	3		4		1
u_2	1		2	1	1	3
u_3	4	5	1	5	3	1
u_4	1	2	3		4	1
u_5		2	4	1	5	

Fig. 1　Format of the rating data after σ operation.

图 1　σ 运算后的评分数据集格式

图 1 中仍有若干空值, 我们要建立神经网络预测它们的值. 比如预测 $u_1 i_5$ 单元格的值: 图 1 稍作处理后得到图 2, 表格上半部分相当于训练集, 其中 In 部分是输入向量, Out 部分是理想输出.

对图 3 所示 BP 神经网络进行训练, 得到稳定的网络结构后, 输入[2, 3, 0, 4, 1], 可得到 u_1 对 i_5 的评分.

可以重复类似步骤, 对图 2 其他空值进行上面

所述神经网络训练-预测的过程. 直到整个评分矩阵的稀疏度小于指定的上限

User	In					Out
	i_1	i_2	i_3	i_4	i_6	i_5
u_2	1	0	2	1	3	1
u_3	4	5	1	5	1	3
u_4	1	2	3	0	1	4
u_5	0	0	2	4	5	1
u_1	2	3	0	4	1	?

Fig. 2 Training/test set used for BP network.
图 2 评分数据集作为 BP 网络的训练/测试集

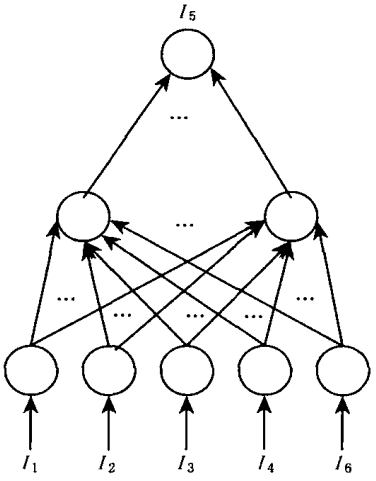


Fig. 3 Structure of the BP network used to predict
图 3 用于预测的 BP 网络结构

BP 网络是一个 3 层网络, 主要输入参数包括 3 层的权值向量初始值 $\boldsymbol{W}^1$, $\boldsymbol{W}^2$, $\boldsymbol{W}^3$, 精度控制参数 ξ 和学习率 α. 激活函数使用 Logistic Function:

$$F(net) = a + \frac{b}{1+e^{-d\times net}},$$

其中, a, b, d 为常数, net 为神经元的加权输入.

BP 神经网络中间层结点数的确定, 我们采用经验公式 $n_m = sqrt(n_i + n_o) + 1$ 计算最小中间层结点数, n_m, n_i, n_o 分别是中间层、输入层、输出层结点数. 采用试错法(trial-and-error)求解合适的中间层结点数, 最后算得的 n_m 值一般分布在 $[n_i, 2\times n_i]$.

4 实 验

4.1 数据集

使用 GroupLens 研究项目组(http://www.grouplens.org)提供的一个著名的电影评分数据 MovieLens 作为测试数据集.

我们使用的是有 10 万条记录的数据集, 这个记录集记录了 943 个用户对 1682 部电影的评分, 每个用户至少对 20 部电影进行了评分, 评分值范围从 1~5.

4.2 度量标准

协同过滤推荐系统评价标准可以分为三大类: 预测精确度度量、分类精确度度量和等级精确度度量[12]. 我们使用预测精确度度量中的平均绝对偏差 MAE(mean absolute error)[3,7,8] 和分类精确度度量中的接收器操作特性 ROC(receiver operation characteristic)敏感度[3,7,8] 对算法进行评价.

MAE 是一个使用最广泛而且最容易解释的统计精确度量法标准. 定义如下: $\langle p_i, q_i\rangle$ 是用户评分对, 其中 p_i 是项目预测得分, q_i 是实际得分, *MAE* 就是 N 个这样评分对差的算术平均值:

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

MAE 越小, 推荐系统的准确度越高.

所谓 *ROC* 敏感度是指随机选择的"好"的项在推荐系统列表中的比例. *ROC* 敏感度值范围是 0~1, 值越大表明推荐系统性能越好. 一个随机生成推荐列表的推荐系统 *ROC* 敏感度理想情况显然是 0.5.

4.3 实验结果

实验在 PC Pentium® Ⅳ 1.7GHz Redhat Linux 9.0, 512MB RAM, GNU C++ 3.3.2 完成.

把第 4.1 节的数据集按照 4/1 的比例划分为训练集/测试集; 为了使 *ROC* 敏感度能正确工作, 定义评分大于或等于 4 项为"好"的项, 否则为"坏"的项.

把我们的算法 BP-CF 和一般的基于余弦法、基于修正余弦法寻找最近邻居的协同过滤算法作比较, 使用 *MAE* 和 *ROC* 两个度量标准, 分别得到图 4 所示 *MAE* 随最近邻居数变化而变动的数据点折线图和图 5 所示 *ROC* 随最近邻居数变化而变动的

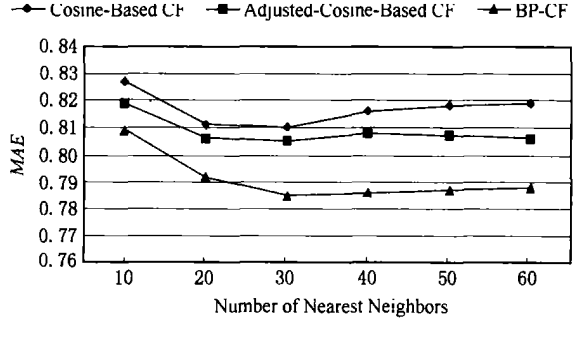


Fig. 4 *MAE* vs. number of nearest neighbors.
图 4 *MAE* 随最近邻居数变化图

数据点折线图. 两图均显示, *BP-CF* 算法有着更好的性能表现

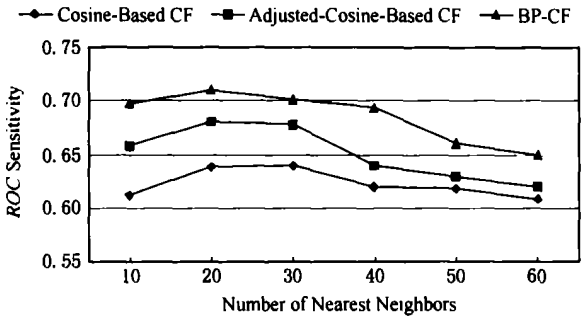


Fig. 5　*ROC* vs. number of nearest neighbors.

图 5　*ROC* 随最近邻居数变化图

5　结论和进一步的工作

本文主要针对协同过滤算法中数据的极端稀疏性对推荐质量的影响, 采用神经网络进行评分预测, 降低数据集的稀疏度, 从而提高协同过滤推荐系统的推荐质量. 这个过程会延长找到最近邻居集的时间, 但问题并不严重, 对算法稍做改动后, 这个过程是可以离线运行的, 这样这个问题就可以得到缓解. 另外我们也在进行使用类似聚类的方法试图解决运行时间所造成可扩展性问题的研究. 本质上, 提高推荐质量和增强系统的扩展性这两个问题是矛盾的, 我们今后的工作重点也是努力在两者中间寻找一个最佳平衡点.

参　考　文　献

1　S. Ansari, R. Kohavi, L. Mason, *et al*. Integrating E-commerce and data mining: Architecture and challenges. In: Proc. 2001 IEEE Int'l Conf. Data Mining. Los Alamitos, CA: IEEE Computer Society Press, 2001. 27～34

2　B. Sarwar, G. Karypis, J. Konstan, *et al*. Analysis of recommendation algorithms for E-commerce. In: Proc. 2nd ACM Conf. Electronic Commerce. New York: ACM Press, 2000. 158～167

3　B. Sarwar, G. Karypis, J. Riedl. Item-based collaborative filtering recommendation algorithms. In: Proc. 10th Int'l World Wide Web Conference. New York: ACM Press, 2001. 285～295

4　J. Breese, D. Hecherman, C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. 14th Conf. Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann, 1998. 43～52.

5　B. M. Sarwar, G. Karypis, J. A. Konstan, *et al*. Application of dimensionality reduction in recommender system——A case study. In: Proc. ACM WebKDD 2000 Web Mining for E-Commerce Workshop. New York: ACM Press, 2000. 82～90

6　C. C. Aggarwal. On the effects of dimensionality reduction on high dimensional similarity search. In: Proc. 12th ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems. New York: ACM Press, 2001. 256～266

7　B. Sarwar, J. Konstan, A. Borchers, *et al*. Using filtering agents to improve prediction quality in the groupLens research collaborative filtering system. In: Proc. ACM Conf. Computer Supported Cooperative Work (CSCW). New York: ACM Press, 1998. 345～354

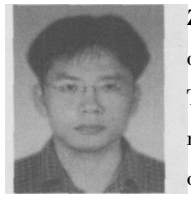
8　N. Good, J. B. Schafer, J. A. Konstan, *et al*. Combining collaborative filtering with personal agents for better recommendations. In: Proc. 16th National Conf. Artificial Intelligence (AAAI-99). Menlo Park, CA: AAAI/MIT Press, 1999. 439～446

9　K. Goldberg, T. Roeder, D. Gupta, *et al*. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 2001, 4(1): 133～151

10　S. H. S. Chee, J. Han, K. Wang. RecTree: An efficient collaborative filtering method. In: Lecture Notes in Computer Science 2114. Berlin: Springer, 2001. 141～151

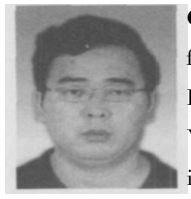
11　Zhao Liang, Hu Naijing, Zhang Shouzhi. Algorithm design for personalization recommendation systems. Journal of Computer Research and Development, 2002, 39(8): 986～991 (in Chinese)
(赵亮, 胡乃静, 张守志. 个性化推荐算法设计. 计算机研究与发展, 2002, 39(8): 986～991)

12　J. L. Herlocker, J. A. Konstan, L. G. Terveen, *et al*. Evaluating collaborative filtering recommender systems. ACM Trans. Information Systems. 2004, 22(1): 5～53



Zhang Feng, born in 1974. Ph. D. candidate in the School of Information Science and Technology, Sun Yat-sen University. His research interests include data mining, E-commerce and machine learning algorithms.

张锋, 1974 年生, 博士研究生, 主要研究方向为数据挖掘、电子商务、机器学习算法



Chang Huiyou, born in 1962. Ph. D., professor and Ph. D. supervisor in the School of Information Science and Technology, Sun Yat-sen University. His research interests include intelligent algorithm design, complicated system modeling, enterprise information systems and their integration, etc.

常会友, 1962 年生, 博士, 教授, 博士生导师, 主要研究方向包括智能算法设计、复杂系统建模、企业信息系统及集成等

Research Background

With the booming of E-commerce, recommender systems are more and more widely used in this area. Collaborative filtering is one of the major technologies used in recommender systems. Scalability and quality are two major challenges in collaborative filtering recommender systems. This paper addresses the quality issue, which is mostly invoked by the sparsity of datasets. BP neural networks have powerful learning and modeling capabilities. They are effective in processing non-complete information. Borrowing these capabilities from BP neural networks, this study fills in the null values with reasonable predicts, thus decreasing the sparsity of datasets and increasing the recommendation quality of collaborative filtering recommender systems. This work is supported by the Guangdong Natural Science Foundation (No. 05100302).

2006 年全国软件与应用学术会议
征文通知

全国软件与应用学术会议(NASAC)由中国计算机学会软件工程专业委员会和系统软件专业委员会联合主办, 是中国计算机软件领域一项重要的学术交流活动. 第五届全国软件与应用学术会议 NASAC2006 将由国防科学技术大学计算机学院承办, 于 2006 年 9 月 7 日至 9 日在湖南长沙举行. 此次会议将由国内核心刊物(计算机工程与科学)以增刊形式出版论文集, 还将选择部分优秀论文推荐到核心学术刊物(EI 检索源)发表, 并将评选优秀学生论文. 欢迎踊跃投稿

征文范围(但不限于下列内容)

- 需求工程
- 构件技术与软件复用
- 面向对象与软件 Agent
- 软件体系结构与设计模式
- 软件开发方法及自动化
- 软件过程管理与改进
- 软件质量、测试与验证
- 软件再工程
- 软件工具与环境
- 软件理论与形式化方法
- 操作系统
- 软件中间件与应用集成
- 软件语言与编译
- 软件标准与规范
- 软件技术教育
- 应用软件

论文要求

① 论文必须未在杂志和会议上发表和录用过

② 论文篇幅限定 6 页(A4 纸)内

③ 会议只接受电子文档 PDF 或 PS 格式提交论文. 排版格式请访问会议网址

重要日期 论文投稿截止日期: 2006 年 5 月 31 日

论文录用通知日期: 2006 年 6 月 30 日

学术会议及活动日期: 2006 年 9 月 7 日至 9 日

联系方式

联系人: 舒绍娴, 国防科学技术大学计算机学院

Tel: 0731-4576321

Email: nasac2006@nasac.net

更详细的内容请访问 NASAC 2006 网址: http://www.nasac.net