结合最大熵模型和 tag 特征的混合推荐系统①

王卫平，杨 磊

(中国科学技术大学 管理学院，合肥 230026)

摘 要： 由于用户数目的不断增多以及信息量的快速膨胀，传统协同过滤（CF）中的数据矩阵稀疏性问题显得愈为突出。为此我们提出了一种新的混合推荐方法。首先，我们在最大熵模型下综合考虑 tag 信息和 rating 信息作为约束条件，然后分别针对 tag 信息和 rating 信息定义相关的特征并且计算其相应的权重，最后利用先前计算出的权重预测当前用户对于目标项目的评分概率分布，并且选出概率最大的作为预测评分。实验证明，该方法能有效提升推荐系统的准确率。

关键词： 最大熵；推荐系统；协同过滤；稀疏性；tag

Hybrid Recommendation System Combining Maximum Entropy and Tag Features

WANG Wei-Ping, YANG Lei

(School of Management, University of Science and Technology of China, Hefei 230026, China)

Abstract: Because of the growing number of users and the rapid expansion of information, sparse problem of data matrix in traditional collaborative filtering becomes more seriously. We proposed a new hybrid recommendation system. Firstly, we consider tag information and rating information as constraints under maximum entropy model. Secondly, we define the features of tag information and rating information and calculate the corresponding weights. Finally, we use previously weights to predict probability distribution of target item for current user, then we choose the highest probability as predicted rating. Experiment results show that the proposed method can effectively improve the accuracy of recommendation systems.

Key words: maximum entropy; recommendation system; collaborative filtering; sparse; tag

1 引言

随着互联网以及电子商务的发展，推荐系统已经成为了一项重要的研究内容，其在传统商业中的角色和销售员非常类似，即：向顾客提出产品信息和推荐，帮助顾客寻找到自己需要的产品，从而完成整个购买过程。根据推荐系统所使用的技术大致可以分成三类：①基于内容的，②基于协同过滤技术的，③基于两种技术混合型的[1]。第一种方法即比较项（产品）的相似性从而形成推荐，第二种方法是根据历史用户以及目标用户的购买行为特征相似从而形成推荐，第三种方法是上述两种方法的综合运用。

协同过滤是迄今为止应用最为成功的个性化推荐技术[2]。其基本思想就是根据用户兴趣的相似性来推荐资源，即首先根据用户的评分历史构建评分矩阵，然后根据一定的度量方法计算当前用户（当前项目）的相似性，从而得出当前用户（当前项目）的邻居，最后根据这些邻居的评分值预测当前用户对于目标项目的评分值。不过现有的协同过滤算法存在这样一个问题，当用户量以及信息量发生剧增时，评分矩阵会变得非常的稀疏[3]，这样会造成用传统相似性度量方法计算出来的邻居不够精确，所以推荐系统的推荐质量也会急剧下降。

Tag 作为一种“大众分类”是伴随着 web 2.0 和 3.0 才被广泛应用的，是分享知识的一种机制。它允许用户用自己主观喜欢的单词或短语来标识当前的资源（电影、音乐、图片等），它最大的特点是带有用户强烈的的主观性，因为用户一般是事后才添加 tag，其次对于同一个目标资源可以用多个 tag 来标识，并且这

① 收稿时间:2010-11-05;收到修改稿时间:2010-12-20

些 tag 节点都是平行关系，不存在父-子节点。

最大熵原理是在 1957 年由 E.T.Jaynes 提出的，其主要思想是，在只掌握关于未知分布的部分知识时，应该选符合这些知识但熵值最大的概率分布。最大熵模型就是在预测随机事件概率分布时，预测应当满足所有的已知约束条件，且不能对结果有任何的主观假设（因为任何预测都会产生新的约束），在这样的情况下，预测结果概率分布最均匀，风险最小[4]。

在论文中，我们提出一个新的混合推荐系统，即在最大熵模型下，综合考虑传统 CF 的 rating 信息和 tag 信息，把其作为模型约束条件，然后预测当前用户对目标项目的评分概率分布，最后选出概率最大的评分作为预测值。

2　基于最大熵模型的混合推荐系统

在本节，我们提出基于最大熵方法的推荐系统模型以及相应的算法。

2.1 相关定义

定义.

$U=\{u_1,u_2,...,um\}$：m 个用户

$T=\{t_1,t_2,...,tn\}$：n 个项目

$G=\{g_1,g_2,...,g_k\}$：k 个 tag

Rm×n：用户-项目评分矩阵，m 行代表 m 个用户，n 列代表 n 个项目，第 i 行 j 列的元素 r_{ij} 为用户 i 对项目 j 的评分值。

Gn×k：项目-tag 评分矩阵，n 行代表 n 个项目，k 列代表 k 个 tag。

其中：

$$g_i^k=\begin{cases}1 & if \quad tag_k\text{标记在项目}\ j\text{上}\\ 0 & Otherwise\end{cases}$$

定义 1. 对于 rating 数据来说，每一个用户可以表示成以下集合，即：$u_i=\left\{\left(t_1^i,r_1^i\right),\left(t_2^i,r_2^i\right),...\right\},t_k^i\in T,r_k^i\in R_{m\times n}$，并且 r_k^i 从｛1，2，3，4，5｝中取值。

定义 2. 对于 tag 数据来说，每一个项目可以表示成以下集合，即：$t_j=\left\{g_1^j,g_2^j,...\right\},g_k^j\in G$，这里 g_k^j 的即标记在项目 t_j 上的 tag。同样，每一个用户亦可以表示成以下集合，即：$u_i=\left\{t_1^i,t_2^i...\right\},t_k^i\in T$。

定义 3. $H(u_i)$，即用户关于 rating 信息和 tag 信息的历史数据，是 U 的一个子集。

2.2 最大熵模型

推荐系统主要包括两部分的信息，即 rating 信息和 tag 信息，为了预测当前用户对于目标项目的评分概率分布，我们用 $P(<t_d,r_d>|H(u_i))$ 表示当前用户在 $H(u_i)$ 为历史数据的情况下，对于目标项目 t_d 评分 r_d 的概率。

在本文中，我们把 rating 信息和 tag 信息看成模型的条件约束，即模型中的特征项（feature）是基于上述两种信息建立，具体定义如下：

1）基于 rating 信息的 feature

对于 rating 信息来说，我们选取用项目之间的相似度来描述项目之间的关系，进而定义对于 rating 信息的 feature，采用的度量标准为修正的余弦相似性：

$$Sim(t_a,t_b)=\frac{\sum_{k\in U_{ab}}(r_a^k-\bar{r}_k)(r_b^k-\bar{r}_k)}{\sqrt{\sum_{k\in U_a}(r_a^k-\bar{r}_k)^2}\sqrt{\sum_{k\in U_b}(r_b^k-\bar{r}_k)^2}}$$

上式中，U_{ab} 表示对项 a 和项 b 共同评分的用户集合，U_a 和 U_b 分别表示对项 a 和 b 评分过的用户集合，r_a^k 和 r_b^k 分别表示用户 k 对项 a 和 b 的评分，$\bar{r}_k$ 表示用户 k 对项的平均评分。

对于每一对 (t_a,t_b)，如果 $Sim(t_a,t_b)$>=u(此处的 u 为用户自行设定的阈值，论文实验中采取的 top-N 算法，当 N 取 4 时，实验结果较好)，我们定义 feature 如下：

$$f_{t_a,r_a,t_b,r_b}^r(H(u_i),t_b,r_b)=\begin{cases}1 & \text{if 用户对}t_a\text{和}t_b\text{分别评分}r_a\text{和}r_b\\ 0 & Otherwise\end{cases}$$

2）基于 tag 信息的 feature

对于 tag 信息来说，由于 tag 具有非常大的灵活性，所以造成了大量的同义词、近义词等，从而导致评分质量下降。相关研究表明，尽管一个项目可能会被添加上百个 tag,但只有很少一部分 tag 是频繁用到的，而且这些少量频繁使用的 tag 集合也是稳定的[5]。所以论文首先对 tag 进行预处理，只提取了频繁 tag 来表示项目。

论文用每个项目所包含的 tag 之间的相似度来描述项目之间的关系，进而定义对于 tag 信息的 feature：

$$Sim(t_a,t_b)=\frac{\min\{\sum_{k\in G_{ab}}g_a^k,\sum_{k\in G_{ab}}g_b^k\}}{\max\{\sum_{k\in G_a}g_a^k,\sum_{k\in G_b}g_b^k\}}$$

上式中，G_{ab} 表示对项 a 和项 b 共同标注的 tag 集合，G_a 和 G_b 分别表示对项 a 和 b 标注的 tag 集合。

对于每一对 (t_a,t_b)，如果对于每一对 (t_a,t_b)，如果 $Sim(t_a,t_b)$>=u（经实验确定，当 u 值为 0.25 时，实验效果较好），我们定义 feature 如下：

$$f^{t}_{t_a,r_a,t_b,r_b}(H(u_i),t_b,r_b)=\begin{cases}1 & \text{if 用户对 } t_a \text{ 和 } t_b \text{ 分别评分 } r_a \text{ 和 } r_b\\ 0 & \text{Otherwise}\end{cases}$$

在定义好关于 rating 信息和 tag 信息的 f_s 后，模型的约束条件就可以表述成：

$$\sum_{u_i}\sum_{t_b\in T}\mathrm{P}(<t_b,r_b>|H(u_i))f_s(H(u_i),t_b,r_b)=\sum_{u_i}f_s(H(u_i),D(H(u_i))) \quad (1)$$

上式中 $D(H(u_i))$ 表示在训练集（training data）中 u_i 的评分信息，由于等式左边的期望分布应该等于等式右边的实际观察值，所以 $\mathrm{P}(<t_b,r_b>|H(u_i))$ 的概率分布亦可表述如下：

$$\mathrm{P}(<t_b,r_b>|H(u_i))=\frac{\exp(\sum_s\lambda_s f_s(H(u_i),t_b,r_b))}{Z(H(u_i))} \quad (2)$$

上式中 $Z(H(u_i))=\sum_{t_b\in T}\mathrm{P}(<t_b,r_b>|H(u_i))$，$Z(H(u_i))$ 是个归一常数，用来保证分布总和为 1，而权重 λ 是需要我们通过 training data 来估计的。这样的话，我们就可以把 rating 和 tag 信息转化成不同的 feature，然后通过 training data 来估计其相应的 λ 值，最后用方程（2）来预测最后对目标项目的打分概率。

现在有很多算法来预测权重 λ 值，最流行的就是 GIS,IIS,SCGIS 这三种，在综合比较这三种算法后，发现前 2 种算法在迭代效率和收敛速度方面都不如 SCGIS 好[6]，所以我们选择 SCGIS 算法来预测权重值。具体算法如下：

$z[1..S]=R$
$s[1..S,1..R]=0$
$observed[s]=\sum_{u_i}f_s(H(u_i),t_b,r_b)$
$r=(t_b,r_b)$
for each feature f_s
  $\exp ected=0$
  for each output r
    for each instance u_i *such that* $f_s(H(u_i),r)\neq 0$
      $\exp ected[s]+=f_s(H(u_i),r)\times e^{s[u_i,r]}/z[u_i]$
  $\delta_s=\frac{1}{\max_{u_i,r}f_s(H(u_i),r)}\log\frac{observed[s]}{\exp ected[s]}$
  $\lambda_s+=\delta_s$
  for each output r
    for each instance u_i *such that* $f_s(H(u_i),r)\neq 0$
      $z[u_i]-=e^{s[u_i,r]}$
      $s[u_i,r]+=\delta_s$
      $z[u_i]+=e^{s[u_i,r]}$

2.3 算法描述

在利用 SCGIS 算法计算好每个 feature 相应的权重后，系统利用方程（2）计算当前用户 u_i 对于目标项目 t_d 评分为 r_d 的概率分布，最后选出概率最大的作为对于目标项目的预测评分。算法具体描述如下：

输入：所有与当前用户评分项目的相关信息，包括 rating 和 tag 信息，通过 training data 估算出的 λ 值。

输出：当前用户对目标项目评分的概率分布。

Step1. 对于所有的目标项目，假设其打分可能为（1-5）。

Step2. 利用方程（2）来计算概率分布。

Step3. 把计算结果从大到小排列，选出概率最大的作为对于目标项目的预测评分。

3 实验结果及其分析

本实验所使用硬件环境：CPU: AMD 3200 内存：1G 硬盘：80 G；软件环境：数据库：Sql-server 2008 开发工具: C#。

3.1 数据集

实验用到的数据集是 MovieLens 站点提供的数据集（http://www.grouplens.org）。官方数据集总共包括 71567 个用户对 10681 部电影的 10,000,054 个评分和 95580 个 tag，每个用户至少对 20 部电影进行了评分。评分值范围从 1-5。我们从中抽取了 3000 个用户对 550 个电影的评分记录，以及相关的频繁 tag 1073 个。

3.2 度量标准

评价推荐系统质量的度量标准有统计精度度量方法和决策支持精度度量[7]。统计精度度量方法的平均绝对误差（mean absolute error ,MAE）易于计算和理解，是最常见的一种推荐质量度量方法，本文亦采用这种方法。

定义 4. 假设目标的预测评分集合为 $\{p_1, p_2, p_3, p_4\cdots p_n\}$，实际值为 $\{q_1, q_2, q_3, q_4\cdots q_n\}$，那么平均绝对误差定义为：

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

从上式可知，MAE 越小，推荐质量越高。

3.3 实验结果

为了检验本文提出的混合推荐系统的有效性，我们拿传统的基于项目的协同过滤方法（Item_based CF）

来进行对比。对于每个用户，我们首先把数据集拆分成 50%的训练集和 50%的测试集。然后分别把两种方法在同样的邻居数下进行对比，邻居数的变化范围定在（4~12），结果如下：

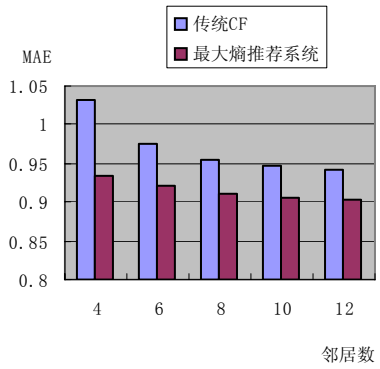

图 1　不同邻居数下 MAE 对比

从图 1 中可以看出，最大熵推荐系统推荐的准确率比传统 CF 更高，同时，我们发现传统 CF 在进行推荐时，其计算最近邻居时的 sim 值并不是特别的高，最高的只有 0.5-0.6 左右，所以我们相信造成这一结果最大的可能就是 rating 数据的稀疏性，而最大熵推荐系统综合考虑了 rating 和 tag 两部分信息，当 rating 的数据受到到稀疏性影响时，tag 信息为推荐准确率提供了保证，所以我们又做一实验，我们把训练集和测试集的比重不断的变小，验证这两种方法的 MAE 值。（传统 CF 的邻居数选其 MAE 值最小时的 12）。

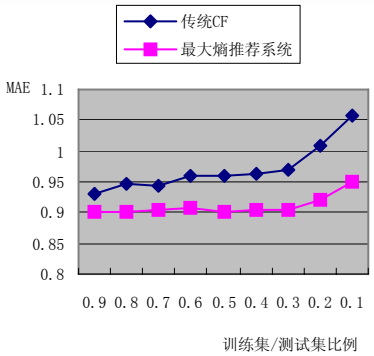

图 2　不同训练集/测试集比值下 MAE 对比

从图 2 中可以看到，随着 training 数据越来越稀疏，最大熵推荐系统的推荐质量也比传统 CF 要越来越好，一个可能的原因就是伴随着 rating 数据变稀疏后，其计算出来的邻居非常不精确，最后影响了传统 CF 最后的推荐质量，对于最大熵推荐系统来说，rating 数据方面的 feature 存在同样的问题，但 tag 数据方面的 feature 受到的影响较小，其仍然能够做出较为精确的推荐。

4　结语

本文针对协同过滤稀疏性问题展开研究，利用最大熵模型，综合考虑了 rating 信息和 tag 信息对于预测评分结果的影响。实验结果表明，本方法能够有效的提高推荐质量。未来的研究重点是如何更有效的利用 tag 信息来提高推荐系统准确率，例如对 tag 在语义上进行处理或是有更好的分类方法的运用等。

参考文献

1 Ansari S, Kohavi R, Mason L, et al . Integrating Ecommerce and data mining : Architecture and challenges. Proc. of the 2001 IEEE Int'l Conf. Data Mining. Los Alamitos, CA: IEEE Computer Society Press, 2001.27–34.

2 高凤荣.个性化推荐系统关键技术研究[博士学位论文].北京:中国人民大学,2003.

3 Breese J. Hecherman D. Kadie C. Empirical analysis of predietive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence(UAI'98). 1998.43–5214. http://baike.baidu. com/view/1313589.html?fromTaglist

5 徐雁斐,张亮,刘炜.基于协同标记的个性化推荐.计算机应用与软件, 2008,25(1):9–13.

6 Goodman J. Sequential conditional generalized iterative scaling. Proc. of NAACL-2002, 2002.

7 Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. Proc. of the 10th International World Wide Web Conference, 2001. 285–295.