计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(10):190−195 [doi: 10.15888/j.cnki.csa.006003]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于标签的评分信息熵推荐算法①

叶　婷

(南京财经大学 信息工程学院, 南京 210046)

摘　要: 由于标签是由用户根据自己的理解和喜好随意进行标注的因此存在大量的噪声标签, 导致基于标签的推荐系统准确率不高. 针对这种情况, 提出了结合评分信息熵的标签推荐算法. 算法通过判断用户在标注标签的评分稳定程度来确定该标签对于用户的重要性从而过滤掉噪声标签将重要标签赋予较高权重, 并构建用户的兴趣模型, 最后应用到协同过滤算法中产生推荐. 该算法能有效地利用评分权重并结合信息熵来增强推荐准确率, 与以往的基于标签的推荐算法进行对比, 能获得满意的推荐效果.

关键词: 标签; 评分信息熵; 兴趣模型; 协同过滤; 推荐算法

引用格式: 叶婷.基于标签的评分信息熵推荐算法.计算机系统应用,2017,26(10):190–195. http://www.c-s-a.org.cn/1003-3254/6003.html

Label-Based Score Information Entropy Recommendation Algorithm

YE Ting

(College of Information and Engineering, Nanjing University of Finance and Economics, Nanjing 210046, China)

Abstract: As the label is marked by the user according to their own understanding and preferences, the expression of the concept is fuzzy and there are a large number of noise tags, resulting in the low efficiency of the traditional label-based recommendation algorithm recommended. casein view of this problem, a tag recommendation algorithm combining the score information entropy is proposed. The algorithm determines the importance of the tag for the user in order to build the user's interest model for the rating of the label. The algorithm can effectively use the score weight and combine the information entropy to enhance the recommendation accuracy, and compared with the previous label-based recommendation algorithm, it can get a satisfactory recommendation effect.

Key words: tag; score information entropy; interest model; collaborative filtering; recommendation system

随着 Internet 的迅猛发展, 人们的衣食住行交流沟通方面也渐渐离不开互联网技术. 然而, 互联网中充斥着复杂多样的信息, 网络信息过载的问题也日益严重. 不可否认, 搜索引擎的出现和使用在很大程度上给用户带来了便利. 它可以根据用户设置的检索条件进行信息匹配, 从而推送出相关信息返回给用户. 这种检索过程要求用户能够明确地表达自己的信息需求, 然而并不是所有的需求和偏好都能通过关键字来描述. 传统的搜索引擎技术已经不能满足用户的搜索需要, 因此个性化推荐技术孕育而生, 且很好的缓解了信息过载问题[1].

由社会化标签是用户为方便自己使用对资源的概要描述, 它不仅反映了用户的喜好兴趣, 还可以反映被标注的资源的特性, 因此得到了学术界和产业界的重视. 目前已有不少学者将社会化标签应用到推荐系统中. Marimho 等人[2]提出了基于用户、资源、标签三元关分离三个二维二维矩阵, 计算向量矩阵的相似度来求得与目标用户有相似兴趣的其他用户从而进行推荐. Symeonidis[3]根据社会标签系统的用户、项目、标签数据构建三阶张量模型, 并结合聚类方法以减小张量

① 基金项目: 科技部科技支撑项目 (BAH29F01); 江苏省重点研发计划 (BE2016178)
收稿时间: 2017-01-17; 采用时间: 2017-02-20

维度从而简化推荐运算. Jiang 和 Zhou 等[4]为了提高基于标签的推荐性能, 考虑结合时间因素以及社会关系信息从而优化推荐结果. Cao J 等[5]提出了一种新的半监督学习算法, 该算法先在少量标注标签的用户集中训练得到一个贝叶斯分类器, 并融合混合型协同过滤算法提出 Web 服务的双向推荐机制. 以上研究多采用机器学习的相关算法, 由于标签自身的语义模糊等特征且社会化系统中有大量的噪声标签存在, 导致最终的推荐结果不是很令人满意.

尽管目前在社会化标签的推荐算法上已经有很多研究, 但大部分的研究多是考虑标签复杂的语义信息或是对标签进行“一视同仁”的处理. 然而, 由于用户的认知水平、教育背景存在差异, 对事物的理解也会有所差别, 因此标签质量的优劣也不确定. 由此可知, 相同的标签对于不同的用户来说其重要程度也会有所不同. 通常, 如果一个标签多次被一个用户用来标记资源, 则可认为这个标签对这个用户的重要程度很高. 常用的计算标签权重的 TF-IDF[6,7]方法就是利用特性将标签与其他用户的标签区分开来, 如果该标签较少被其他用户使用, 那么该标签对于该用户的意义就较大. 然而, 用户对于标签的使用只是出于自身对于资源的理解, 并不代表其对于该标签的喜爱, 并且根据用户对于标签标注的资源打分不同, 对于同一标签的喜爱程度也是不同的.

本文考虑引入标签信息熵的特征实现对标签区别对待, 之前也有一些应用信息熵的信息到推荐系统当中. 如 Saneeha 等[8]通过将用户的兴趣特征进行层次聚类, 通过计算用户兴趣特征对应到各个类簇的比例从而确定各兴趣项的熵权重. Harita 等[9]通过用户对电影数据是否感兴趣程度分为 1、0 两个类别继而训练最大熵模型和标签特征并应用到推荐系统中. Javier 等[10]考虑结合标签的评分数据来计算用户兴趣项的权重从而产生推荐. 以上研究方法多是通过数据挖掘算法和大量的数据模型训练计算信息熵, 或简单的结合评分数据计算用户的兴趣项权重, 多是要求用户主观上将兴趣项分类而不能通过算法自适应地修正用户的兴趣项权重, 最终推荐效果一般且耗费大量时间.

因此本文提出了基于评分信息熵的用户标签权重计算方法, 通过衡量用户对标注标签的资源的打分不确定程度并结合归一化的评分数据综合评价用户标签权重, 从而构建用户的基于标签的兴趣模型, 并应用到推荐算法中.

1 基于评分信息熵的用户标签权重计算

在本文中, 用户的兴趣模型是通过用户的标签数据进行构建, 并融合了用户的评分、标签频数、标签信息熵等特征计算兴趣项的权重. 其中, $U=u_1,u_2,,u_{|U|}$ 定义为一组用户,$I=i_1,i_2,,i_{|I|}$定义为一组项目,$T=t_1,t_2,,t_{|T|}$ 定义为标注在项目的标签. 总的来说, 用户对项目的评分可以体现用户对其兴趣度, 用户给项目打的标签体现了用户对项目的理解, 本文联系用户的评分数据以及标签数据来计算兴趣项权重从而协同过滤推荐.

定义 1. (用户—资源评分矩阵) 定义为 $U*I$ 矩阵 $R(U*I)=(R_{i,j})$, 其中 U 表示用户集合, I 表示资源集合, $R_{i,j}$ 为用户 u 对资源 i 的评分. 计算用户标签权重前, 先将每个用户的评分归一化即 $\|\vec{r}\|=1$, 用户 u 在资源 i 上标签 t 的评分权重表示为:

$$wr_{u,i}(t)=\frac{R_{u,i}}{\sqrt{\sum_{j=1}^{|I|}R_{u,j}^2}} \tag{1}$$

定义 2. (信息熵) 由于同一个标签可能会被标注在不同的资源中, 因此用户对于同一个标签可能打分多次. 因此给标签定义信息熵如下:

$$H_{ut}=-\sum_{i=1}^{n}p_{ut}(i)\log_2 p_{ut}(i) \tag{2}$$

其中, $p_{ut}(i)$表示用户 u 在标注标签 t 的资源上打分为 i 的比例, n 为用户 u 对标注标签 t 的资源上打分不同值的数目. H_{ut}值大小代表了用户在该标签上打分的不确定程度, 其值越小, 代表打分的不确定性程度低, 用户在该标签的打分比较稳定, 则可认为该标签对用户重要程度很高. 相反, 其值越大, 表示打分不确定性高, 用户在该标签的打分比较混乱, 则可认为该标签对用户的重要程度较低. 综上将标签对用户的重要性可表示为: $1-H_{ut}$.

定义 3. (用户标签兴趣模型) 对于$\forall t_j\in T$, $\forall u\in U$, 用户 u 的标签兴趣模型向量为$T_u=((t_1,w_{u,t_1}),(t_2,w_{u,t_2}),\ldots,(t_n,w_{u,t_n}))$, 其中w_{u,t_j}表示用户 u 对于特征项标签 t_j 的兴趣度权重. 用户对于标签兴趣度的大小可以结合标签的信息熵以及用户对于标注标签资源的评分计算得到, 因此用户 u 对于标签 t 的兴趣度w_{u,t_j}为:

$$wr_{u,i}(t)=\frac{R_{u,i}}{\sqrt{\sum_{j=1}^{|I|}R_{u,j}^2}} \tag{3}$$

其中, T_u 为用户 u 使用的标签集合, $wr_{u,i}(t)$表示用户 u 在标注标签 t 上的资源打分, $f_u(t)$表示用户使用标签 t 标注资源的频数.

2 算法流程

由于标签和评分数据都是用户根据自己的理解喜好标注的, 所以可以很好的表达用户的兴趣. 考虑社会化标签系统中存在一定的噪声标签, 因此通过计算标签的评分信息熵来修正标签的不确定表达从而增强准确性. 本文采用用户的标签数据表示用户兴趣模型的兴趣项, 并通分析用户对于同一个标签的打分不确定程度确定标签的信息熵并结合评分从而计算标签兴趣项的兴趣权重, 考虑将用户对于标注在同个资源的标签的打分稳定性区别对待, 从而能更加精确的表示用户的兴趣. 最后将其兴趣模型应用到协同过滤算法当中.

基于协同过滤的推荐算法需要得到与目标用户兴趣相似的用户集合, 从而挖掘出这个集合中用户喜欢的且没有标注过的资源. 为了找到 k 近邻相似用户, 我们定义用户的相似度如下:

$$sim_{u,v}=\frac{\sum_{t\in(IT_u\cap IT_v)}(w_{u,t}-\overline{w_u})(w_{v,t}-\overline{w_v})}{\sqrt{\sum_{t\in IT_u}(w_{u,t}-\overline{w_u})^2}\times\sqrt{\sum_{t\in IT_v}(w_{v,t}-\overline{w_v})^2}} \tag{4}$$

其中IT_u和IT_v分别表示用户 u, v 标注在项目上的标签集合, $\overline{w_u}$表示用户 u 的平均兴趣度.

对于给定的用户 $u\in U$, 与其相似度较大的 k 近邻用户定义如下:

$$N_k(u)=\arg\max_{v\in U\setminus\{u\}}^{k} sim_{u,v} \tag{5}$$

对于给定用户的 k 近邻用户, 得出用户 u 对于未标注的资源 i 的预测兴趣度定义如下:

$$score(u,i)=\frac{\sum_{v\in N_k(u)}w_{v,t}\times sim(u,v)}{\left|\sum_{v\in N_k(u)}sim(u,v)\right|} \tag{6}$$

则基于标签的评分信息熵推荐算法表述如下:

输入: 用户—标签—资源数据$\{U,T_u,I\}$, 其中$T_u=\{t_{u1},t_{u2},,t_{un}\}$

输出: 目标用户 u 的 Top-N 推荐集$recommend-list$

第 1 步. 结合用户-评分-资源记录, 根据公式 (1) 计算用户对于标签的评分权重并进行归一化处理.

第 2 步. 结合用户在标签上的不同打分数据并计算其对应的概率, 根据公式 (2) 计算不同标签的信息熵值.

第 3 步. 结合标签的信息熵值, 根据公式 (3) 结合用户的评分数据修正用户对于其标签的兴趣度权重.

第 4 步. 构建用户基于标签以及评分信息熵权重的兴趣模型$\{(T_1,W_1),(T_2,W_2),\ldots,(T_n,W_n)\}$, 其中 T_1 表示用户的标签兴趣项, W_1 表示用户对应的评分信息熵权重. 根据公式 (4) 计算用户相似度, 并构造用户的相似度矩阵 S_{N*N}.

第 5 步. 基于用户对资源的历史记录, 根据用户的相似度矩阵 S_{N*N} 查询用户与其他用户的相似度, 并用公式 (5) 得到用户的 k 近邻用户, 依据公式 (6) 依次计算用户与这些资源的预测兴趣度.

第 6 步. 按预测兴趣度从大到小排序, 取前 N 个资源组成 Top-N 推荐集合$recommend-list=\{i_1,i_2,\ldots i_N\}$并输出. 其中算法的伪代码如下.

表 1 基于标签的评分信息熵推荐算法

输入: R$n\times m$, Q=$\{U,T,I\}$, k//R$n\times m$为用户资源评分矩阵, Q为用户-资源-标签, k为邻居数目
输出: $recommend-list$
1: for all user u in R do
2: use eq(1) compute $wr_{u,i}(t)$
3: use eq(2) compute H_{ut}
4: use eq(3) to compute w_{u,t_j}
5: produce user intrest-model
6: end for
7: for all user u in Q do
8: use eq (4) compute the similarity between the u and other users
9: produce the user similarity matrix
10: use eq (5) find KNN neighbors
11: use eq (6) predict the score of user to unlabeled item
12: ascending the order of score($u.i$)
13: take the top-N item
14: end for
15: output $recommend-list=\{i_1,i_2,\ldots,i_N\}$

3 实验

3.1 实验数据

本部分采用 MovieLens 和 Delicious 两组数据集来测试算法的效率, 具体数据集合见表 2, 数据集中的评分数据大小在 0.5-5 之间, 间隔 0.5, 共 10 个分值. 为了验证本文算法的有效性, 将本文算法 SE-CF 与两个较新的算法 tensor-u[11] 采用张量分解模型并考虑用户标签的语义信息进行推荐与 colla-tv[12]结合 TF-IDF 算

法以及用户的评分矩阵进行推荐和一个经典的基于标签的推荐算法 T-CF[13]进行对比验证.

表 2 数据集结构

数据 数据集	用户数	资源数	评分数	标签数
Movielens	71567	10681	1000054	95580
Delicious	3285	79883	1227625	21106

3.2 度量标准

推荐算法实验中的重要组成部分之一就是推荐质量的评价指标计算, 不同推荐算法的评价方法也会有所差别. 在本文中, 实验采取的评价指标如下:

(1) 准确率 (*Precision*) 表示在用户产生的推荐列表中, 有多大比例的资源是用户真正喜欢的, 设 $R(u)$ 指用户在训练集上依据用户的行为作出的推荐列表, $T(u)$ 表示用户在测试集中的行为列表. 则推荐结果的准确率定义如下:

$$Precision = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \tag{7}$$

(2) 召回率 (*Recall*) 表示用户真正喜欢的商品中, 有多大比例的商品进入了推荐列表, 其计算公式如下:

$$Recall = \frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{8}$$

(3) *F-measure*, 因为准确率和召回率在一定程度上是相互矛盾的, 如果推荐的准确率高可能意味着它的召回率会比较低, 因此现在实验中多使用一个平衡以上两种指标的综合评价方法即 *F-measure*, 其计算公式如下:

$$F-measure = \frac{2\times Precision\times Recall}{Precison + Recall} \tag{9}$$

3.3 实验结果分析

(1) 确定邻居数 k 的影响

本部分测试邻居数 k 的值对协同过滤推荐结果的影响, 其中设置 k 的值的变化从 10 到 100, 值变化间隔为 10, 将本文算法 SE-CF 与算法 T-CF 在度量标准 *Recall* 上进行对比验证. 在协同过滤算法中, 邻居用户的数量 k 一定程度上影响着推荐精度, k 如果太小将无法获取足够的候选资源集合, k 如果太大也将增加计算成本以及时间消耗. 设置不同的 k 值, 实验结果的召回率也会随之改变, 其结果如图 1 所示.

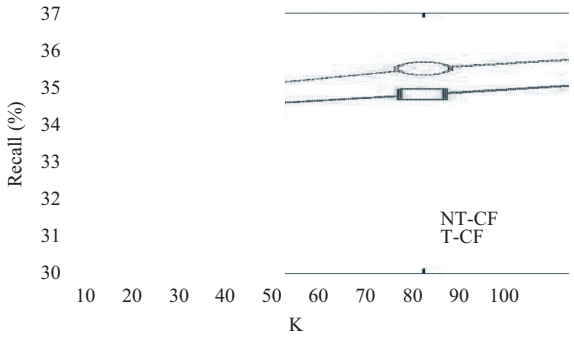

图 1 不同邻居数 k 值对 *Recall* 变化情况

从图 1 可以看出, 当邻居数 k 的值约为 35 时, 本算法 SE-CF 和 T-CF 算法的召回率基本相同, 当 k>35 时, 本算法 SE-CF 的召回率才高于 T-CF 算法. 实验过程分析可得, 经典 T-CF 算法是基于用户共同标注的资源数目来确定用户之间的相似度, 而 SE-CF 算法是通过计算用户标签的评分信息熵构建用户的标签兴趣模型再计算两用户之间的相似程度, 相似度的值与用户共同标注的资源数目没有直接关联, 因此可能存在两个用户相似度较大但共同标注的资源很少的现象. 如图 1 所示, 随着邻居用户数目 k 的增大, SE-CF 算法的召回率的上升速率要明显高于经典 T-CF 算法. 另外, 从图中可以看出两个算法的召回率在 k=70 左右趋于稳定, k>70 时, 二者的召回率上升速度都很缓慢, 因此在后面的协同过滤算法比较中设置邻居数 k=70.

(2) 实验对比验证

为了更好的验证实验, 本文采用五折交叉验证, 每次随机选取 80% 的实验数据集作为训练集, 剩余 20% 数据为测试集, 对五次结果取平均作为最终结果, 同时考虑在 *Precision*、*Recall*、*F-measure* 三个度量标准进行比实验验证. 在本文算法中训练集用于计算用户标签的评分信息熵以确定用户标签的兴趣权重, 并构建用户兴趣模型应用到协同过滤推荐算法当中, 然后测试集中利用训练集得到的推荐结果进行比较计算. 根据前面的实验所知, 设 k=70, 随着推荐列表的数量 N 值的变化, 三个度量标准的值也会不同, 并将 N 设置从 5 到 25 变化, 值变化间隔为 5, 实验数据结果如表 3、表 4 所示.

对比图示结果如图 2、图 3 所示.

通过上图的对比可以看出无论推荐列表数量 N 取任何值, 基于标签的评分信息算法在准确率、召回

率、*F-measure* 三个度量标准都明显好于其他基于标签的推荐算法, 同时也可以看出基于标签的评分信息熵推荐算法在 N=10 时推荐效果最好且比其他几个推荐算法准确度平均提高了 6.7% 左右. 笔者分析认为, 本文算法之所以能优于其他三个算法在于: 一是考虑将丰富信息量的标签信息构成用户的兴趣模型能较好的表达用户的兴趣; 二是考虑有一定的噪声标签的存在因此提出通过计算标签的打分不确定程度来计算标签的兴趣权重从而将标签对用户的重要性区别对待, 对于打分不稳定不能精准表现用户兴趣的标签可以通过计算其信息熵来修正降低其重要性.

表 3 MovieLens 实验数据对比

Number of Recommendation	Precision(%)				Recall(%)				F-measure(%)			
	SE-CF	tensor-u	colla-tv	T-CF	SE-CF	tensor-u	colla-tv	T-CF	SE-CF	tensor-u	colla-tv	T-CF
N=5	59.2	54.1	50.6	42.2	38.1	35.7	33.2	26.9	46.7	42.4	39.5	33.6
N=10	43.4	40.2	40.7	32.4	53.8	46.5	43.3	35.5	49.5	43.8	40.5	34.4
N=15	36.1	34.9	34.2	27.6	64.1	54.3	47.4	40.3	45.6	40.1	37.1	32.5
N=20	33.6	31.2	31.7	25.4	66.6	57.4	52.8	42.4	42.4	38.5	36.6	32.1
N=25	27.5	27.4	27.6	23.8	69.8	59.6	54.6	45.6	36.6	36.6	35.7	31.2

表 4 Delicious 实验数据对比

Number of Recommendation	Precision(%)				Recall(%)				F-measure(%)			
	SE-CF	tensor-u	colla-tv	T-CF	SE-CF	tensor-u	colla-tv	T-CF	SE-CF	tensor-u	colla-tv	T-CF
N=5	62.2	60.2	49.2	48.5	36.2	38.0	31.2	30.2.	472	44.5	32.5	32.3
N=10	51.1	47.5	41.0	39	47.6	47.1	40.3	38.1	50.1	47.1	37.5	35.0
N=15	46.5	43.5	36.1	32.5	53.1	56.3	45.1	44.4	49.3	45.3	36.1	32.5
N=20	40.2	38.5	32.5	30.1	66.3	60.8	49.5	47.3	46.1	43.7	35.2	31.5
N=25	38.5	34.9	31.3	29.3	71.1	67.1	57.1	53.2	44.7	41.3	33.1	30.1

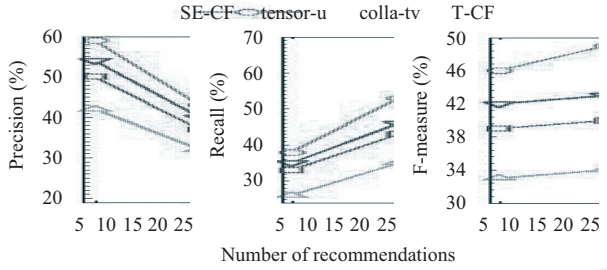

图 2 Movielens 数据集结果对比

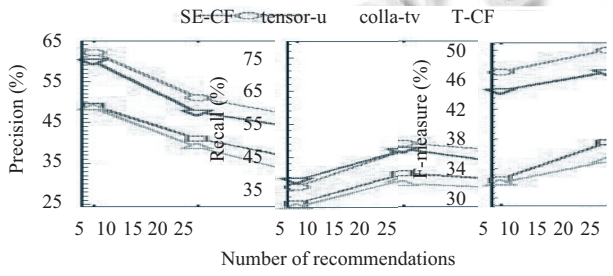

图 3 Delicious 数据集结果对比

4 结语

本章基于标签构建用户的兴趣模型, 并提出了一种用户标签评分信息熵的兴趣度计算方法, 对于用户重要性高的标签予以较高的权重. 通过在 MovieLens 数据集和 Delicious 数据集上将本文算法与其他三个算法进行实验对比, 验证了本文算法要明显好于其他三个基于标签的推荐算法. 由于用户的评分信息局限于几个离散的数值, 因此基于社会化标签的信息熵计算还不够准确. 目前, 将信息熵应用于标签的推荐算法研究还比较新颖, 因为信息熵本身有很丰富的解释信息以及物理背景, 因此其应用前景还是十分可观的.

参考文献

1 Verma C, Hart M, Bhatkar S, *et al*. Improving scalability of personalized recommendation systems for enterprise knowledge workers. IEEE Access, 2016, 4: 204–215. [doi: 10.1109/ACCESS.2015.2513000]

2 Marinho LB, Schmidt-Thieme L. Collaborative tag recommendations. In: Preisach C, Burkhardt H, Schmidt-Thieme L, *et al*. eds. Data Analysis, Machine Learning and Applications. Berlin, Heidelberg, Germany. 2008. 553–540.

3 Symeonidis P. ClustHOSVD: Item recommendation by combining semantically enhanced tag clustering with tensor hosvd. IEEE Trans. on Systems, Man, and Cybernetics: Systems, 2016, 46(9): 1240–1251. [doi: 10.1109/TSMC.2015.2482458]

4 Jiang ZG, Zhou A, Wang SG, *et al*. Personalized service recommendation for collaborative tagging systems with social relations and temporal influences. Proc. of the 2016 IEEE International Conference on Services Computing (SCC). San Francisco, CA, USA. 2016. 786–789.

5 Cao J, Wu ZA, Wang YQ, *et al*. Hybrid collaborative filtering algorithm for bidirectional web service recommendation. Knowledge and Information Systems, 2013, 36(3): 607–627. [doi: 10.1007/s10115-012-0562-1]

6 Buck C, Koehn P. Quick and reliable document alignment via TF/IDF-weighted cosine distance. Proc. of the 1st Conference on Machine Translation, Volume 2: Shared Task Papers. Berlin, Germany. 2016. 672–678.

7 Chen KW, Zhang ZP, Long J, *et al*. Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Systems with Applications: An International Journal, 2016, 66: 245–260. [doi: 10.1016/j.eswa.2016.09.009]

8 Ahmed S, Tepe K. Entropy-based recommendation trust model for machine to machine communications. Proc. of the 8th International Conference Ad Hoc Networks. Ottawa, Canada. 2016. 297–305.

9 Mehta H, Dixit VS, Bedi P. Weighted difference entropy based similarity measure at two levels in a recommendation framework. Proc. of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI). Mysore, India. 2013. 2076–2083.

10 Parra-Arnau J, Rebollo-Monedero D, Forné J. Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems. Entropy, 2014, 16(3): 1586–1631. [doi: 10.3390/e16031586]

11 Zhang SW, Ge YY. Personalized tag recommendation based on transfer matrix and collaborative filtering. Journal of Computer and Communications, 2015, 3(9): 9–17. [doi: 10.4236/jcc.2015.39002]

12 Peng J, Zeng DD, Zhao HM, *et al*. Collaborative filtering in social tagging systems based on joint item-tag recommendations. Proc. of the 19th ACM International Conference on Information and Knowledge Management. Toronto, ON, Canada. 2010. 809–818.

13 Tso-Sutter KHL, Marinho LB, Schmidt-Thieme L. Tag-aware recommender systems by fusion of collaborative filtering algorithms. Proc. of the 2008 ACM Symposium on Applied Computing. New York, USA. 2008. 1995–1999.