

一种基于高斯混合模型的协同过滤算法

成英超 王瑞胡 胡章平

(重庆文理学院机器视觉与智能信息系统重点实验室 重庆 402160)

摘要 协同过滤技术中的矩阵分解是推荐系统中的有效技术手段。而现在主流的矩阵分解算法假设推荐系统评分数据服从高斯分布,因而受数据噪声影响,其鲁棒性达不到预期。为了解决这个问题,提出基于高斯混合模型的矩阵分解算法。设定评分数据服从高斯混合分布,在此基础上应用基于贝叶斯概率的矩阵分解模型。同时,提出一种基于半监督学习的数据实验方法,充分挖掘有标签和无标签数据。实验结果表明,基于高斯混合分布的矩阵分解算法对评分噪声拥有更强的免疫力,同时可以提高预测准确率。

关键词 协同过滤,高斯混合模型,推荐系统,半监督学习

中图分类号 TP39 文献标识码 A

Novel Approach on Collaborative Filtering Based on Gaussian Mixture Model

CHENG Ying-chao WANG Rui-hu HU Zhang-ping

(Key Laboratory of Machine Vision and Intelligent Information System, Chongqing University of Arts and Sciences, Chongqing 402160, China)

Abstract Recommender system is a competitive solution to solve information overload problem. And collaborative filtering is an effective method for the recommender systems. Matrix factorization is widely used for collaborative filtering. However, the existing matrix factorization techniques are affected by the rating noise, and their robustness is not up to people's expectations. We attributed the negative effect of the rating noise to the universal applied hypothesis that the rating data is subject to the Gauss distribution. In order to solve this problem, we proposed a collaborative filtering algorithm based on Gaussian mixture model. We assumed the rating data obeying the Gauss mixture distribution, and then applied the Bayesian probability matrix factorization model for recommendation. Besides, a semi-supervised algorithm has been proposed, which gets both labeled and unlabeled data involved. The experimental results show that the collaborative filtering algorithm based on Gaussian mixture model is much more robust and it can alleviate the negative effect of rating noise and improve the accuracy of prediction as well.

Keywords Collaborative filtering, Gaussian mixture model, Recommender system, Semi-supervised learning

1 引言

互联网已经和人们的日常生活紧密结合在了一起。近年来,推荐系统广泛应用于以电子商务为代表的各种互联网应用,以解决信息过载问题^[1]。大多数推荐系统采用协同过滤技术,这种技术通过分析用户过往的大量网上历史记录,学习用户的网上行为模式并建立模型,帮助用户从海量信息中筛选其需要的信息,并推荐给用户^[2]。亚马逊、淘宝等国内外知名网站的推荐服务大都是基于协同过滤的。

在推荐系统中,用户的历史行为数据被组织成评分矩阵 $R_{U \times I}$ 的形式,其中的数据项 R_{ui} 代表用户 u 对项目 i (如商品、新闻、影音作品等)的评分。现实中,用户只会给极少数的项目评分,相应信息系统的评分矩阵相当稀疏。因此,推荐系统需要预测出缺失的评分,并将其按分值排序,将评分值最高的项目推荐给用户^[3-4]。

基于协同过滤的推荐系统通常采用矩阵分解算法实现。

具体而言,矩阵分解需要从评分矩阵 $R_{U \times I}$ 中推导出一个低阶近似矩阵 $X_{U \times I} (= W_{U \times D} V_{I \times D}^T)$,其中 $D \ll U, I$ (低阶近似矩阵 $X_{U \times I}$ 的秩 D 远小于用户 $user$ 的数目 U 和推荐项 $item$ 的数目 I), $X_{U \times I} = \arg \min_{X^*} Err(R_{U \times I} - X^*)$; $Err(\cdot)$ 表示误差。 $W_{U \times D} (V_{I \times D})$ 的每一个行向量 $W_u (V_i)$ 表示用户评分矩阵的一个特征向量。因此,缺失的评分数据可以由 $W_u^T V_i (= X_{ui})$ 计算得到。

然而,现实数据集中通常存在一些不真实的数据,评分矩阵中存在一定量的数据噪声。用户会下意识地给出不当评分,或者信息系统受到人为评分干扰攻击,因此真实评分数据中存在很多噪声^[5]。现在的矩阵分解算法大都使用 Euclid 范数 $\|\cdot\|_E^2$ 作为误差函数,因此在评分噪声环境下表现得不够鲁棒。之所以选择 Euclid 范数作为误差函数,是因为人们假设评分数据服从高斯分布。高斯分布对离群点的敏感性致使少量的评分噪声会使得推荐准确率显著下降。

高斯混合分布可以平滑数据噪声^[6],且利用统计方法分

本文受重庆市教委人文社会科学研究项目(11SPK05),重庆文理学院机器视觉与智能信息系统重点实验室开放基金(MVHIS2016Z01),重庆市教委科技项目(KJ131225),重庆市永川区科技攻关项目(Ycstc2013ad5001)资助。

成英超 博士,CCF 会员,主要研究方向为算法设计、数据挖掘、推荐系统等;王瑞胡 博士,教授,主要研究方向为机器学习、模式识别、图像处理等,E-mail:wangr@zjut.edu.cn(通信作者);胡章平 硕士,讲师,主要研究方向为智能信息系统等。

析评分数据,发现高斯混合分布能更好地拟合评分数据。据此,在基于贝叶斯概率的矩阵分解框架下构建高斯混合矩阵分解模型。实验结果表明,基于高斯混合模型的矩阵分解算法使推荐系统获得了良好的鲁棒性,可以有效对抗评分噪声,提升推荐的准确率。

2 矩阵分解

有两种主流的矩阵分解算法:基于期望最大化的奇异值分解算法^[7]和基于贝叶斯概率的矩阵分解算法^[8]。这两种方法在预测准确率上都可以取得不错的效果。

基于期望最大化的奇异值分解算法在期望最大化的框架下做矩阵分解:此时,要估计的参数就是低阶矩阵 $X_{U \times I}$ 。每一次迭代,该算法根据已有的评分 $R'_{U \times I}$ 估算缺失值期望,并将其填入矩阵 $X_{U \times I}$,以此估算未获得评分的项目可能得到的评分。收敛后,可以得到 $X_{U \times I}$ 的局部最优估计。

基于贝叶斯概率的矩阵分解算法属于概率生成模型,它通过贝叶斯推理将矩阵分解。该算法将 R_{ui}, W_u, V_i 看作参数服从特定先验分布的随机变量。引入先验分布可以有效防范过拟合。与基于期望最大化的奇异值分解算法不同,该算法用吉布斯抽样估计 W_u 与 V_i ,其样本服从后验分布 $p(W, V | R'_{U \times I})$ 。

这两种算法的前提都是假设评分数据服从高斯分布,导致了它们对评分噪声的弱鲁棒性。

3 基于高斯混合模型的协同过滤算法

3.1 高斯混合矩阵分解算法

为了消除评分噪声的消极影响,基于高斯混合模型的协同过滤算法(GMCF)设定评分数据服从高斯混合分布。高斯混合模型包括了两个均值相等、方差不同的高斯分布。不同方差的高斯分布混合后可以平滑评分数据的噪声。图1示出了基于高斯混合模型的矩阵分解算法的图形化表示。

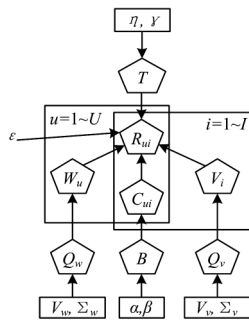


图1 基于高斯混合模型的矩阵分解算法图示

图1也可看作该算法的贝叶斯网络,其中, R_{ui}, W_u, V_i 等分别表示参数服从给定先验分布的不同随机变量。图1中变量的分布如下:

$$p(R_{ui} | C_{ui}, W_u, V_i, T) = N(R_{ui} | W_u^T V_i, T^{-1})^{U_{ui}(1-C_{ui})} \cdot N(R_{ui} | W_u^T V_i, \epsilon T^{-1})^{U_{ui}C_{ui}} \quad (1)$$

$$p(W_u | Q_w) = N(W_u | 0, Q_w^{-1}) \quad (2)$$

$$p(V_i | Q_v) = N(V_i | 0, Q_v^{-1}) \quad (3)$$

$$p(B | \alpha, \beta) = \text{Beta}(B | \alpha, \beta) \quad (4)$$

$$p(C_{ui} | B) = \text{Bernoulli}(C_{ui} | B) \quad (5)$$

$$p(T | \eta, \lambda) = \text{Gamma}(T | \eta, \lambda) \quad (6)$$

$$p(Q_i | v_i, \Sigma_i) = \text{Wishart}(Q_i | v_i, \Sigma_i) \quad (7)$$

其中, $\epsilon > 0, t \in \{w, v\}$ 。 U_{ui} 表示评分数据 R_{ui} 是否缺失: $U_{ui} = 0$

表示评分缺失; $U_{ui} = 1$ 表示评分不缺失。式(5)~式(7)分别表示伯努利分布、伽马分布和维系特分布。

根据 W_u, V_i, T , 可得到式(1)中的 C_{ui} :

$$\begin{aligned} p(R_{ui} | W_u, V_i, T) \\ = \sum_{C_{ui}} p(C_{ui}) p(R_{ui} | C_{ui}, W_u, V_i, T) \\ = p(C_{ui} = 0) N(R_{ui} | W_u^T V_i, T^{-1})^{U_{ui}} + p(C_{ui} = 1) N(R_{ui} | W_u^T V_i, \epsilon T^{-1})^{U_{ui}} \end{aligned} \quad (8)$$

式(8)表示评分数据由高斯混合模型生成,参数 ϵ 是 T^{-1} 的比例因子。

3.2 变分推断

假设 ψ^o 和 ψ^h 分别表示基于高斯混合模型的矩阵分解算法中的已知变量 $\{R\}$ 和隐含变量 $\{Q_w, Q_v, B, C, W, V, T\}$ 。因为很难用贝叶斯推理将后验分布 $p(\psi^h | \psi^o)$ 表示成正则形式,所以采用变分贝叶斯方法^[9](假设各个变量之间相互独立)获取相对简单的 $q(\psi^h)$ 作为复杂的后验分布 $p(\psi^h | \psi^o)$ 的近似。

ψ^o 的对数边际概率可由下式得到:

$$\ln p(\psi^o) = \Gamma(q) + KL(q \| p) \quad (9)$$

其中

$$\Gamma(q) = \int q(\psi^h) \ln \frac{p(\psi^o \cup \psi^h)}{q(\psi^h)} d\psi^h \quad (10)$$

$$KL(q \| p) = - \int q(\psi^h) \ln \frac{p(\psi^h | \psi^o)}{q(\psi^h)} d\psi^h \quad (11)$$

$KL(q \| p)$ 表示不同分布 q 和 p 之间的 KL 距离(相对熵)。因为 KL 距离是非负的,采用 $\Gamma(q)$ 作为 $\ln p(\psi^o)$ 的下界。当最大化 $\Gamma(q)$ ($\Gamma(q) = q(\psi^h)$) 时, $q(\psi^h)$ 会向 $p(\psi^h | \psi^o)$ 逼近,因此 $q(\psi^h)$ 可以看成变分后验分布。为了简化优化过程, $q(\psi^h)$ 被分解为如下形式:

$$q(\psi^h) = \prod_{\varphi \in \psi^h} q(\varphi) \quad (12)$$

对于任意 $\varphi \in \psi^h$, $\Gamma(q)$ 的极大值可由式(13)给出:

$$q(\varphi) \propto E_{-\varphi} [\ln p(\psi^o \cup \psi^h)] \quad (13)$$

式(13)中 $E_{-\varphi}[\cdot]$ 表示分布 $\prod_{\varphi' \neq \varphi} q(\varphi')$ 的期望,利用式(13)

迭代地最大化 $\Gamma(q)$ 。

更进一步地,为了处理未知参数 ϵ , 将 $\Gamma(q)$ 表示成 $\Gamma(q, \epsilon)$, 并用变分期望最大化算法^[10] 优化 $\Gamma(q, \epsilon)$:

$$\text{E-step: } q^{(k+1)} = \arg \max_q \Gamma(q, \epsilon^{(k)})$$

$$\text{M-step: } \epsilon^{(k+1)} = \arg \max_{\epsilon} \Gamma(q^{(k)}, \epsilon)$$

在 E-step 中,先验和变分后验分布都属于同一分布族,将其中一些变分后验分布表示如下:

$$q(B) = \text{Beta}(B | \alpha + \sum_{ui} E[C_{ui}], \beta + \sum_{ui} (1 - E[C_{ui}])) \quad (14)$$

$$q(Q_w) = \text{Wishart}(Q_w | v_w + U, (\sum_{w'}^{-1} + \sum_{ui} E[W_u W_u^T])^{-1}) \quad (15)$$

$$q(Q_v) = \text{Wishart}(Q_v | v_v + I, (\sum_{v'}^{-1} + \sum_{ui} E[V_i V_i^T])^{-1}) \quad (16)$$

$$q(T) = \text{Gamma}(T | \eta', \lambda') \quad (17)$$

$$q(W_k) = N(W_k | \mu, \Lambda) \quad (18)$$

$$q(V_k) = N(V_k | \mu', \Lambda') \quad (19)$$

其中:

$$\eta' = \eta + \frac{1}{2} \sum_{ui} U_{ui} \quad (20)$$

$$\begin{aligned} \lambda' = \lambda + \frac{1}{2} \sum_{ui} U_{ui} E[(R_{ui} - W_u^T V_i)^2] \cdot (1 - E[C_{ui}]) + \\ \frac{E[C_{ui}]}{\epsilon} \end{aligned} \quad (21)$$

$$\Lambda' = E[Q_w] + \sum_i U_{ki} E[V_i V_i^T] \cdot E[T] (1 - E[C_{ki}] + \frac{E[C_{ki}]}{\epsilon}) \quad (22)$$

$$\mu = \Lambda \sum_i U_{ki} R_{ki} E[V_i] \cdot E[T] (1 - E[C_{ki}] + \frac{E[C_{ki}]}{\epsilon}) \quad (23)$$

$$\Lambda'^{-1} = E[Q_w] + \sum_u U_{uk} E[W_u W_u^T] \cdot E[T] (1 - E[C_{uk}] + \frac{E[C_{uk}]}{\epsilon}) \quad (24)$$

$$\mu' = \Lambda' \sum_u U_{uk} R_{uk} E[W_u] \cdot E[T] (1 - E[C_{uk}] + \frac{E[C_{uk}]}{\epsilon}) \quad (25)$$

在 M-step 中,对 $\Gamma(q, \epsilon)$ 中的 ϵ 求偏导数,并令其等于 0,可以得到:

$$\epsilon = E[T] \frac{\sum_{u,i} U_{ui} E[C_{ui}] E[R_{ui} - W_u^T V_i]}{\sum_{u,i} U_{ui} E[C_{ui}]} \quad (26)$$

这两个步骤收敛后,预测评分为:

$$R_{ui} = E[W_u]^T E[V_i] \quad (27)$$

最终,根据式(27)完成整个用户评分矩阵的估算。

4 实验

4.1 半监督学习过程

推荐模型建立完成之后,为了进一步解决数据稀疏性问题,提出一种基于半监督学习的数据学习方法,将有标签数据和无标签数据都纳入推荐过程,以提升推荐系统的准确性。

如图 2 所示,将数据集分为 3 个数据子集,不同的子集中包含差异化的数据样本。因为当数据子集少于 3 个时,数据多样性不足;而数据子集多于 3 个时,会导致过拟合。差异化的子集中包括有评分(有标签)的数据和无评分(无标签)的数据。在不同子集的有标签数据上应用上文提出的基于高斯混合分布的矩阵分解算法,标记各个子集中的无标签数据。这些被标记的原始无标签数据组成不同的教练集(教练集即训练集,但数据是被 GMCF 算法标记的原无标签数据),用于迭代更新其他子集中的原始无标签数据的标签。最后,通过“组合”把获得不同标记的子集合并到一起。

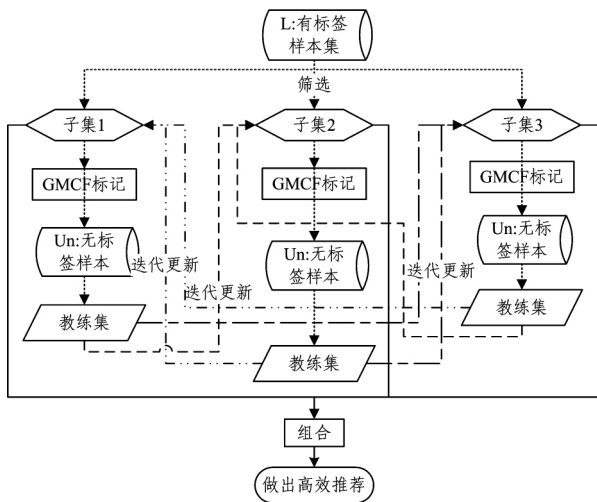


图 2 半监督数据学习过程

不同数据子集的构建至关重要,数据子集间的差异性决定了半监督学习方法的有效性。其差异性主要取决于子集中的有标签数据。以数据置信度为标准,从有标签数据集 L 中筛选样本分配给不同子集。对协同过滤推荐系统而言,用户

u 越活跃,项目 i 被评价的频率越高,相关数据样本的置信度越高,基于此类数据的推荐准确率也越高^[11]。数据集 T_n 中样本 s_{ui} 的置信度定义如下:

$$Z_T(S_{ui}) = \frac{d_u^{(T)} \times d_i^{(T)}}{\tau} \quad (28)$$

其中, τ 是正则化系数, $d_u^{(T)}$ 表示数据集 T_n 中用户 u 的活跃程度; $d_i^{(T)}$ 表示项目 i 的被评分频率。

用基于置信度 $Z_T(S_{ui})$ 的轮盘赌算法 Roulette algorithm 来选取各个子集中的有标签数据,选取样本 s_{ui} 的概率为:

$$\Pr(S_{ui}, T) = \frac{Z_T(S_{ui})}{\sum_{s_{ui} \in L} Z_T(S_{ui})} \quad (29)$$

具体而言,包括以下步骤:

Step1 输入有评分的样本训练集 L 和未评分样本集合 U_n ;

Step2 初始化各个数据子集,设其为空集: $T_1 = \emptyset, T_2 = \emptyset, T_3 = \emptyset$;

Step3

Step3.1 通过式(28)为所有样本计算置信度:

Foreach do

获取置信度 $Z_T(S_{ui}) = \frac{d_u^{(T)} \times d_i^{(T)}}{\tau}$;

End

Step3.2 根据置信度 $Z_T(S_{ui})$,以概率 $\Pr(S_{ui}, T) = \frac{Z_T(S_{ui})}{\sum_{s_{ui} \in L} Z_T(S_{ui})}$,用轮盘赌算法选取有标签数据进入子集 T_n ;

Step4 用 Bagging 方法从无标签数据中随机抽取样本,完成子集分组;

Step5 在不同子集上分别应用基于高斯混合分布的矩阵分解算法,标记各个子集中的无标签数据:

$$R_{ui} = E[W_u]^T E[V_i]$$

获得各个教练集 Θ_n ;

Step6 将教练集中被标记的原始无标签数据加入到其他子集的有标签数据中,迭代更新其他子集中原始无标签数据的标签: $\Theta_1 = \Theta_1 T_2 T_3, \Theta_2 = \Theta_2 T_1 T_3, \Theta_3 = \Theta_3 T_1 T_2$;

Step7 组合 Θ_n 并输出最终的推荐结果。

4.2 实验结果

基于国际公开数据集 MovieLens 获取实验结果。为了验证算法的鲁棒性和效率,在 4 个大小不同的数据集上验证: M1 数据集包含 489 个用户、1466 部电影和对应的 50000 条评分记录; M2 数据集包含 943 个用户、1682 部电影和对应的 100000 条评分记录; M3 数据集包含 1429 个用户、3266 部电影和对应的 200000 条评分记录; M4 数据集包含 6040 个用户、3900 部电影和对应的 1000209 条评分记录。

推荐结果的准确性用均方根误差 RMSE 来表示:

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in TestSet} (r_{ui} - \hat{r}_{ui})^2}{|TestSet|}} \quad (30)$$

其中, u 表示用户, i 表示推荐项, $|TestSet|$ 表示数据集容量, r_{ui} 表示真实评分, $\hat{r}_{ui}$ 表示算法得到的预测评分。RMSE 的值越小,说明预测值与真实值的误差越小。

表 1 首先给出了 3 种经典的推荐算法的实验结果:基于用户的 K 最近邻算法(UB k-NN)、基于项目的 K 最近邻算法(IB k-NN)和经典协同过滤算法(CF)^[12];接着给出了近期其他学者提出的算法的实验结果:通过决策树提取用户偏好的

功能矩阵分解算法(fMF)^[13]和通过挖掘电影类型信息获取更好推荐效果的优化协同过滤算法(LFfMG)^[14];最后给出文中基于高斯混合模型的协同过滤算法(GMCF)的实验结果。

表 1 不同算法推荐准确性的对比

算法	M ₁	M ₂	M ₃	M ₄
UB k-NN	1.0250	1.0267	1.0224	0.9522
IB k-NN	1.0733	1.0756	1.0709	1.0143
CF	0.9310	0.9300	0.9260	0.8590
fMF	0.9508	0.9439	0.9432	0.9413
LFfMG	0.9764	0.9617	0.9515	0.9520
GMCF	0.8982	0.8960	0.8953	0.8338

表 1 中的实验结果证明,基于高斯混合模型的协同过滤算法(GMCF)的推荐准确率获得了较大提升。其中 M₄ 数据集包含最多的用户及其评分数据,相对地更加接近真实场景,更具代表性。图 3 是 G4MCF 算法在 M₄ 数据集上的迭代效果,可以看出算法在迭代大约 25 次后可以达到稳定且较高的推荐准确率;图 4 是包括 GMCF 在内的不同算法在 M₄ 数据集上加入人为噪声(即评分矩阵与随机矩阵相加)后的推荐准确率,图中数据说明 GMCF 算法具有良好的鲁棒性,对评分数据噪声有较强的抵抗力。

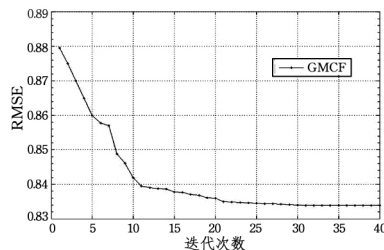


图 3 GMCF 算法在 M₄ 数据集上的迭代效果

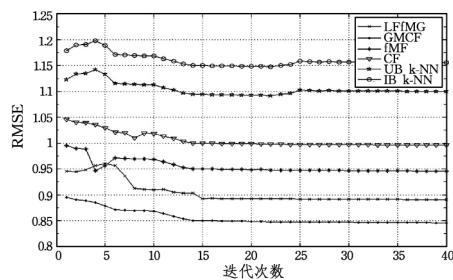


图 4 不同算法在加入数据噪声的 M₄ 数据集上的迭代效果

除了测试推荐算法的预测准确率,还探究了其在解决冷启动问题上的表现。首先评测各个用户的活跃度,然后按活跃度将用户平均分为 5 组,每组分别测评 RMSE 的值。通过观察不同活跃度用户组的 RMSE,可以判断该算法在解决冷启动问题上的表现。其结果如表 2 所列。

表 2 不同算法在冷启动环境下的推荐准确性对比

算法/用户组	组 1	组 2	组 3	组 4	组 5
评分数 #	340	148	77	44	26
UB k-NN	0.9816	1.0191	1.0696	1.1043	1.1381
IB k-NN	1.0156	1.0536	1.1041	1.1396	1.1852
CF	0.8929	0.9100	0.9789	1.0291	1.0786
fMF	0.8749	0.8978	0.9292	0.9704	1.0058
LFfMG	0.8723	0.9011	0.9326	0.9722	0.9987
GMCF	0.8691	0.8829	0.9211	0.9682	0.9910

表 2 中的组 1 是活跃度最高的用户群,组 5 是活跃度最低的用户群。实验结果说明,该算法在解决冷启动环境下的准确率比 k-NN 等算法有了近 10%~20% 的提升。

结束语 本文提出了一种基于高斯混合模型的协同过滤

算法,有效地平滑了评分数据中存在的噪声;还提出了一种基于半监督学习的数据训练方法,将无标签数据纳入推荐过程,缓解了推荐系统中的数据稀疏性问题。实验结果表明,这项工作不仅提升了推荐算法的准确性和鲁棒性,而且有效缓解了冷启动问题。推荐系统是在线服务的,其实时性至关重要,这将是未来工作的重点。一个可能的解决方案是通过 dropout^[15]筛选出具有重大影响力的关键用户以减小训练集规模,通过并行化技术加速训练过程等。

参考文献

- [1] COSTA H, MACEDO L. Emotion-Based Recommender System for Overcoming the Problem of Information Overload[J]. Communications in Computer & Information Science, 2013, 365: 178-189.
- [2] HERLOCKER J L, KONSTAM J A, TERVEEN L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 5-53.
- [3] KANG Z, PENG C, CHENG Q. Top-N Recommender System via Matrix Completion[C]// AAAI. 2016.
- [4] HOFMANN T. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems, 2004, 22(1): 89-115.
- [5] 张燕平, 张顺, 钱付兰, 等. 基于用户声誉的鲁棒协同推荐算法[J]. 自动化学报, 2015, 41(5): 1004-1012.
- [6] 乔少杰, 金琨, 韩楠, 等. 一种基于高斯混合模型的轨迹预测算法[J]. 软件学报, 2015, 26(5): 1048-1063.
- [7] DU S, LIU J, ZHANG C, et al. Probability iterative closest point algorithm for m-D point set registration with noise [J]. Neurocomputing, 2015, 157(CAC): 187-198.
- [8] SALAKHUTDINOV R, MNIH A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]// International Conference on Machine Learning. ACM, 2008: 880-887.
- [9] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 3: 993-1022.
- [10] 李晓旭, 李睿凡, 冯方向, 等. 多视图有监督的 LDA 模型[J]. 电子学报, 2014, 42(10): 2040-2044.
- [11] ZHANG M, TANG J, ZHANG X, et al. Addressing Cold Start in Recommender Systems: A Semi-supervised Co-training Algorithm[C]// Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Springer, 2014.
- [12] KOREN Y. The BellKor solution to the Netflix Grand Prize [OL]. <http://dl.dropbox.com/u/838364/ml-pilot/dinner2-netflixprize.pdf>.
- [13] ZHOU K, YANG S H, ZHA H. Functional matrix factorizations for cold-start recommendation[C]// Proceeding of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011). Beijing, China, July 2011: 69-78.
- [14] MANZATO M G. Discovering latent factors from movies genres for enhanced recommendation[C]// ACM Conference on Recommender Systems. ACM, 2012: 249-252.
- [15] IOSIFIDIS A, TEFAS A, PITAS I. DropELM: Fast neural network regularization with Dropout and DropConnect[J]. Neurocomputing, 2015, 162: 57-66.