

基于 Adaboost 算法的推荐系统评分预测框架^①

徐 日, 张 谧

¹(复旦大学 软件学院, 上海 201203)

²(复旦大学 上海市智能信息处理重点实验室, 上海 201203)

摘 要: 在机器学习领域, Adaboost 算法的实用性和有效性早已被证明. 然而该算法原本是为分类问题设计, 因而在推荐系统领域研究问题中无法直接应用, 对其应用研究相对较少. 本文对 Adaboost 算法进行改进, 通过引入阈值, 将评分预测问题转化为分类问题, 并利用其权重更新的思想训练模型, 提出了一个针对评分预测问题的框架, 可以将训练出的多个模型集成起来得到最终的评分预测, 提高了预测精度. 我们选取矩阵分解模型作为基本模型, 实验结果表明, 使用该框架可以有效提高预测精度.

关键词: 推荐系统; Adaboost 算法; 矩阵分解模型; 评分预测; 推荐精度

引用格式: 徐日, 张谧. 基于 Adaboost 算法的推荐系统评分预测框架. 计算机系统应用, 2017, 26(8): 107-113. <http://www.c-s-a.org.cn/1003-3254/5887.html>

Adaboost-Based Framework For Rating Prediction in Recommender System

XU Ri, ZHANG Mi

¹(School of Software, Fudan University, Shanghai 201203, China)

²(Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 201203, China)

Abstract: In the field of machine learning, the practicality and effectiveness of the Adaboost algorithm has already been demonstrated. However, since this algorithm is originally designed for classification problems, it cannot be applied directly to rating prediction problems in recommender system field. Thus the research in this area is limited. In this paper, we improve the Adaboost algorithm. By introducing the threshold value, we transform rating prediction into classification. By updating weights in the training process, we propose a framework for the rating prediction, which can integrate the multiple training models. The final rating is obtained through the integrated model. We select the Matrix Factorization model as an instance, and the experimental results show that the framework can effectively improve the prediction accuracy.

Key words: recommender system; Adaboost algorithm; matrix factorization model; rating prediction; recommend accuracy

随着信息技术的发展和互联网的普及, 人们获得信息的途径越来越多, 信息量也越来越大, 这也让人们难以从卷帙浩繁的信息中找到自己感兴趣的. 推荐系统是用来解决这一问题的重要方法. 因此推荐系统具有良好的发展前景和应用前景. 目前推荐系统已经走进我们的日常生活, 其范围涵盖了淘宝等电商平台、

电影和视频网站、音乐网站、社交平台、基于位置的服务、广告、阅读、个性化邮件等. 个性化推荐是推荐系统最重要的研究方向, 对于不同用户, 他们的兴趣千差万别, 个性化推荐系统就是为解决用户需求的差异而诞生的. 目前, 推荐系统中最成功以及应用最广泛的技术当属协同过滤, 它根据每位用户的历史选择信

① 收稿时间: 2016-12-04; 采用时间: 2016-12-19

息,以此来找出具有相同兴趣爱好的用户群体,从中找出潜在的信息来进行推荐推荐.不管是理论研究还是应用实践,协同过滤技术都取得了快速的发展.

在机器学习领域,Adaboost 算法的实用性和有效性已被证明,在很多领域均得到应用.然而该算法原本是为分类问题设计,因而在推荐系统领域研究问题中无法直接应用,对其应用研究相对较少.文章[1]利用 adaboost 算法在协同过滤推荐系统中进行攻击检测;文章[2]提出了一个 AdaBPR 的算法,利用用户的隐式反馈,对物品进行 TOP-N 推荐;文章[3]提出了一个名为 AdaMF 的方法,基于 Adaboost,利用矩阵分解算法进行 TOP-N 推荐.

评分预测问题一直以来都是推荐系统的研究热点,很多推荐系统的研究都是利用用户历史评分数据,结合相应的推荐算法来进行评分预测.在评分预测问题中,预测精度是衡量模型好坏的重要指标,本文对 Adaboost 算法进行改进,通过引入阈值,将评分预测问题转化为分类问题,并利用其权重更新的思想训练模型,提出了一个针对评分预测问题的框架,可以将训练出的多个模型集成起来得到最终的评分预测,提高了预测精度.我们选取矩阵分解模型作为基本模型,实验结果表明,使用该框架可以有效提高预测精度.

本文的核心贡献在于将 Adaboost 算法成功应用于推荐系统评分预测问题上来,并解决了算法中权重分配部分存在的问题,并进行了相应的创新,在通过实验证明在提高推荐精度上取得了不错的效果,同时,该

算法可以适用于不同的评分预测模型,是一个集成学习框架.

后文组织如下:第二节介绍 Adaboost 算法及矩阵分解的相关背景知识;第三节介绍我们基于 Adaboost 算法评分预测框架实现;第四节介绍数据集以及实验结果;第五节进行总结.

1 相关背景

1.1 Adaboost 算法介绍

Adaboost 算法是 Adaptive Boosting 的缩写,是一个非常有力量的集成算法.它的诞生是源于回答由 Kearns 和 Valiant 在 1988 年提出的问题:弱可学习 (weakly learnable) 与强可学习 (strongly learnable) 是否是等价的^[4,5].如果可以证明两者是等价的,那么就意味着一个仅仅比随机猜测效果好上一点的弱学习器的学习效果可以被提升到很高的层次,与一个效果显著的强学习器等价.这对于机器学习领域具有显著意义. Robert Schapire 通过研究,发现两者是等价的.并通过构造了一个名为 Boosting 的算法来证明^[6].但这个方法在实践中有一个显著的缺陷是它要求提前知道基础算法的错误范围,但这在实践中经常是不可知的.在 1997 年, Yoav Freund 和 Robert Schapire 提出了一个自适应的 Boosting 算法^[7],克服了以上的缺点,并把它命名为 Adaboost. Adaboost 算法的提出具有极为显著的理论意义,引起了非常多的学习和研究.

下面为 Adaboost 算法的伪代码^[8]:

Input : Data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$; learning algorithm L ; Number of learning rounds T ;

Process :

1. $D_1(i) = 1/m$; %Initialize the weight distribution
2. **for** $t = 1, \dots, T$:
3. $h_t = L(D, D_t)$; %Train a classifier from D under distribution D_t
4. $\varepsilon_t = \Pr_{x \sim D_t, y} [h_t(x) = y]$; % Evaluate the error of h_t
5. **if** $\varepsilon_t > 0.5$ **then break**
6. $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$; % Determine the weight of h_t
7. $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \end{cases}$ %Update the distribution, where
% Z_t is a normalization factor
 $= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ %which enables D_{t+1} to be a distribution
8. **end**

Output : $H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$

1.2 矩阵分解(Matrix Factorization)模型介绍

矩阵分解模型是基于模型的推荐算法的重要部分^[9],它通过矩阵降维的方法将原本较为稀疏的评分矩阵进行补全.该方法最早源自于数学中奇异值分解,但是数学上的解法的计算复杂度很高,在推荐领域,对该算法做了简化,将评分矩阵 R 分解为两个低维度矩阵相乘,形式如下:

$$\hat{R} = Q^T P \quad (1)$$

其中 $P \in \mathbb{R}^{k \times m}$, $Q \in \mathbb{R}^{k \times n}$ 是两个降维后的矩阵.其中, P 矩阵包含了全部用户的偏好, Q 矩阵包含了全部物品的特点.均是 K 维, K 的大小是事先确定的.这样,对每一个用户来说,他都有一个 K 维的向量,其中每个维度隐式的包含了该用户的偏好,与之对应的,每一个物品的向量的每个维度包含了其内在特点.这样,将这两个向量相乘便可以得到一个值,表明用户对该物品的喜爱程度,进而进行后续的推荐.所以,我们定义对于用户 u 对物品 i 的评分的预测值 $\hat{R}(u, i) = \hat{r}_{ui}$,可通过下面公式计算:

$$\hat{r}_{ui} = \sum_k p_{uk} q_{ik} \quad (2)$$

其中 $p_{uk} = P(u, k)$, $q_{ik} = Q(i, k)$.这个式子将向量的每个维度隐式的赋予了意义,将用户喜好与物品特点联系在一起.但是,在现实场景中,除去用户和物品之间存在着隐式的关系外,还会有很多其他情况,包括:物品本身的固有的特性,用户自身的独立的特点以及与物品和用户均无关的性质.这里我们引入 b_i, b_u, μ 三个参数,分别对应于上面的三者.所以在本文中我们使用的预测模型如下:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \quad (3)$$

人们习惯将其成为 SVD 模型,我们也延续这个说法.

2 基于 Adaboost 算法评分预测框架实现

2.1 问题定义

针对推荐系统中的评分预测问题,首先,我们定义在一个推荐系统中,用户集合为 $U = \{u_1, u_2, \dots, u_{|U|}\}$,物品集合为 $I = \{i_1, i_2, \dots, i_{|I|}\}$.来代表用户 u 对物品 i 的评分,我们的数据集表示如下: $D = \{(u, i, r_{ui})_1, (u, i, r_{ui})_2, \dots, (u, i, r_{ui})_m\}$,共计 m 个评分,通常, $m \ll |U| \times |I|$.

2.2 算法改进

在 1.1 节已经提到过 Adaboost 算法最初适用于分类问题,介绍的伪代码同样是用于分类问题.对于评分预测问题,虽然每一个用户 u 对于物品 i 的评分都是整数,但是我们使用的基本模型是矩阵分解模型,该模型在预测用户对物品的评分时做出的预测均为连续函数.我们希望对 Adaboost 算法进行改进,使其可以成为一个应用于评分预测问题的框架.为此,我们针对以下几个问题进行改进和创新:

- (1) 如何将评分预测问题转化为两类分类问题?
- (2) 模型的错误率该如何计算?
- (3) 怎样将训练集数据的权重分布应用与矩阵分解模型上来?原有的权重更新公式是否依然适用,该如何调整?
- (4) 怎样将最后训练出来的多个模型集成起来?

下文四小节分别对应于上面的四个问题进行解决.

2.2.1 问题转化

之所以选择将我们所研究的评分预测这个回归问题转化为两类分类问题,是因为 Adaboost 算法在两类分类问题上具有非常好的效果,其效果早已得到验证.因而在应用于评分预测问题时,我们的考虑是其转化为两类分类问题.所以我们引入参数 ϕ , 作为一个阈值,这个阈值的作用是判断使用矩阵分解模型进行评分预测结果的好坏.举例来说:假设用户对与物品的实际评分为 4 分,那么我们认为,当矩阵分解模型经过训练后,如果用户对物品的评分在 $[4-\phi, 4+\phi]$ 的范围内,我们认为这个预测是较为准确的;如果不在这个范围内,我们认为这个预测是不准确的.通过引入这个阈值 ϕ ,我们将评分预测问题成功转化为两类分类问题.

2.2.2 模型错误率

在分类问题中,模型的错误率只需统计分类的正确性便可以很容易计算得到,但是在评分预测问题中,模型的预测结果 \hat{r}_{ui} 不可避免与真实值 r_{ui} 存在差距.这里我们利用前面提到的阈值,当评分在 $\hat{r}_{ui} \in [r_{ui} - \phi, r_{ui} + \phi]$, 不计算其错误率,否则,我们计算其错误率,计算方式如下:

$$\varepsilon_l = \sum_{u, i \in D^-} w_{ui} \quad (4)$$

2.2.3 权重更新

第三个问题, Adaboost 算法之所以可以有那么明显的效果的一个重要原因,是因为通过利用权重分布,

来影响基本学习算法的训练效果. 具体的方式便是: 对于错误分类的数据, 会加大其权重, 从而在下一轮训练中更加重视对于它们的训练, 而正确分类的数据则会减小其权重, 通过这样的调整, 使学习器在学习的过程中更加重视训练错误的数据. 最后将所有训练出来的模型集成起来, 从而达到更好的效果. 在评分预测问题中, 我们使用的模型的损失函数均可以表示为:

$$\min \sum_{(u,i) \in D} (r_{ui} - \hat{r}_{ui})^2 \quad (5)$$

我们将每一个训练数据的权重以如下方式加入损失函数, 便可以在训练过程起到效果:

$$\min \sum_{(u,i) \in D} w_{ui} [(r_{ui} - \hat{r}_{ui})^2] \quad (6)$$

以矩阵分解模型为例, 在具体训练过程中使用的是随机梯度下降算法来实现收敛. 考虑权重影响, 一个很自然的考虑便是将权重赋给下面求和公式的每一项:

$$\min_{b^*, p^*, q^*} \sum_{(u,i)} w_{ui} \left[(r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \right] \quad (7)$$

其中 w_{ui} 表示用户 u 对于物品 i 的评分的权重.

从而得到新的递推公式:

Input: Data set $D = \{(u, i, r_{ui}, w_{ui})_1, (u, i, r_{ui}, w_{ui})_2, \dots, (u, i, r_{ui}, w_{ui})_m\}$;

Base rating prediction algorithm L , eg: SVD, PMF...;

Number of learning rounds T ; Threshold: ϕ ; Update control parameter: μ ;

Process:

1. $(u, i, r_{ui}, w_{ui})_i^1 = 1/m$; %Initialize the weight distribution
2. **for** $t = 1, \dots, T$:
3. $h_t = L(D, D_t)$; %Train a recommender algorithm h_t from D under distribution D_t
4. **for all** $\hat{r}_{ui} \notin [r_{ui} - \phi, r_{ui} + \phi]$ in D_t , the set is D_t^- :

$$\varepsilon_t = \sum_{u,i \in D_t^-} w_{ui}^t$$
 % Evaluate the error rate of h_t
5. $\alpha_t = \mu \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$; % Determine the weight of h_t
6. $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } \hat{r}_{ui} \notin [r_{ui} - \phi, r_{ui} + \phi] \\ \exp(\alpha_t) & \text{if } \hat{r}_{ui} \in [r_{ui} - \phi, r_{ui} + \phi] \end{cases}$ %Update the distribution, where
 Z_t is a normalization factor
 %which enables D_{t+1} to be a distribution
7. **end**

Output: $\hat{r}_{ui}(u, i) = \sum_{t=1}^T \alpha_t h_t(u, i)$

$$\begin{aligned} b_i &\leftarrow b_i + w_{ui} \cdot \gamma \cdot (e_{ui} - \lambda \cdot b_i) \\ b_u &\leftarrow b_u + w_{ui} \cdot \gamma \cdot (e_{ui} - \lambda \cdot b_u) \\ q_i &\leftarrow q_i + w_{ui} \cdot \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \\ p_u &\leftarrow p_u + w_{ui} \cdot \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \end{aligned} \quad (8)$$

在原始的 Adaboost 模型中, 决定权重更新的公式为:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \quad (9)$$

其中, $1/2$ 是其中一个参数. 这个参数的得到是源于其主要解决的两类分类问题的损失函数. 然而在我们的目前要解决的问题的损失函数在上文已经提到. 所以该参数很可能并不适用于我们当前的问题. 具体的数值要在实验中进行调节, 我们使用参数 μ 来控制错误率在权重更新过程中的影响, 形式如下:

$$\alpha_t = \mu \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \quad (10)$$

2.2.4 模型集成

通过前面的训练, 我们可以得到 T 个模型, 我们使用加权平均的方法来集成各个模型的评分结果. 因为每一个模型均有其对应模型权重 α_t , 最终评分预测的结果:

$$\hat{r}_{ui}(u, i) = \sum_{t=1}^T \alpha_t h_t(u, i) \quad (11)$$

2.2 最终算法

最终的算法如下:

上述算法伪代码中的输入有五个: (1) 基本的推荐算法, 如: SVD 算法; (2) 训练集 $D: D = \{(u, i, r_{ui}, w_{ui})_1, (u, i, r_{ui}, w_{ui})_2, \dots, (u, i, r_{ui}, w_{ui})_m\}$; (3) 算法迭代的轮数 T ; (4) 阈值 ϕ ; (5) 权重更新控制参数 μ .

训练过程如下, 对应上述算法伪代码:

(1) 第 1 行: 将所有训练集数据赋予一个权重, 初始权重相等, 为数据总量的倒数 $1/m$.

(2) 第 2~7 行: 利用推荐算法进行模型训练, 参数 T 表示训练的总轮数, 每一轮均得到一个模型.

1) 第 3 行: 在第 t 轮有推荐算法训练得到的第 t 个模型 h_t .

2) 第 4 行: 计算该模型错误率 ε_t .

3) 第 5 行: 计算该模型 h_t 的权重 a_t . 这里可以看到, 错误率越低, 其权重越大.

4) 第 6 行: 更新训练数据权重. 更新策略为: 在模型 h_t 中, 被预测评分 $\hat{r}_{ui} \in [r_{ui} - \phi, r_{ui} + \phi]$ 时, 权重减小; 否则权重增大. 权重更新后重新将权重归一化.

最终预测评分的结果由训练得到的 T 个模型的结果进行加权平均.

3 实验与结果分析

3.1 数据集

我们使用的数据集为美国明尼苏达大学 GroupLens 研究项目组所收集到 MovieLens 数据集. 我们使用了其中两个不同大小的数据集: MovieLens 100 K 和 MovieLens 1 M^[10]. MovieLens 数据集包含用户对电影的评分信息, 分值范围为 1~5 分, 均为整数.

表 1 MovieLens 100 k 和 MovieLens 1 M 数据集

	100K	1M
用户数	943	6040
物品数	1682	3952
平分数	100,000	1,000,209
平均用户打分数目	106.04	165.60

3.2 评测标准

我们使用交叉验证方法, 将上述数据集分为五份, 每一份都包含 20% 的评分数据, 每次选取其中一份作为测试集, 剩下的四份作为训练集.

评分预测的预测精度常用的评价标准是: 均方根误差(RMSE)和平均绝对误差(MAE). 对于测试集中的一个用户 u 和物品 i , 令 r_{ui} 是用户 u 对物品 i 的实际评分, 而 \hat{r}_{ui} 是推荐算法给出的预测评分, 那么 RMSE 的定义为:

$$RMSE = \frac{\sqrt{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}}{|T|} \quad (12)$$

MAE 计算每一组真实评分与预测评分差距的绝对值, 再除以总数, 它的定义为:

$$MAE = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|} \quad (13)$$

针对 RMSE 和 MAE 这两个指标的优劣, 文章^[11]中作者认为, 认为 RMSE 使用平方项的惩罚加大了对预测不准确的评分的惩罚, 相比 MAE 标准来说, 更加苛刻. 并且在由 Netflix 公司举办的 Netflix Prize 大赛中, 将 RMSE 指标作为其最终的标准. 所以在本文中我们使用 RMSE 作为评价标准.

3.3 实验结果与分析

我们选用矩阵分解模型作为我们的基本模型, 为了证明我们的框架推广性, 除了 1.1 中展示的 SVD 模型, 我们对 PMF^[12]模型也进行实验.

在矩阵分解模型的训练过程中有两个很关键的参数: 学习速率 γ 与惩罚系数 λ , 这两个参数直接影响到训练的效果. 经验来说, 学习速率 γ 的设置不能够太大或太小. γ 太大会导致沿梯度方向下降的“步伐”太大, 从而可能会导致迭代不收敛, 从而发散. 而“步伐”太小, 则一方面会导致迭代次数太多, 时间消耗导致效率降低, 不利于实验结果的获得; 另一方面也导致陷入局部最小值的几率增大, 难以获得全局最优解. 经过我们的实验中实际测试, 最后对于 SVD 模型, 我们选取学习速率 $\gamma=0.01$ 与惩罚系数 $\lambda=0.03$; 对于 PMF 模型, 我们选取学习速率 $\gamma=0.005$ 与惩罚系数 $\lambda=0.02$ 进行后续实验.

3.3.1 效果展示

表 2 展示了我们展示了我们使用 Adaboost 框架的效果. 从表中我们可以的发现使用 Adaboost 框架的模型相比之前的单个模型来说, 效果有了不小的提升. 同时可以发现, 相比于 MovieLens 100 K 数据集, 我们提出的框架在 MovieLens 1 M 的数据集上效果更好. 对于 SVD 模型, 在 MovieLens 100 K 数据集中, RMSE 从 0.9155 下降到了 0.9018, 提升了 1.5%; 在 MovieLens 1 M 数据集中, RMSE 从 0.8057 下降到了 0.7864, 提升了 2.4%. 对于 PMF 模型, 在 MovieLens 100 K 数据集中, RMSE 从 0.9272 下降到了 0.9119, 提升了 1.7%; 在 MovieLens 1 M 数据集中, RMSE 从 0.8216 下降到了 0.8027, 提升了 2.5%.

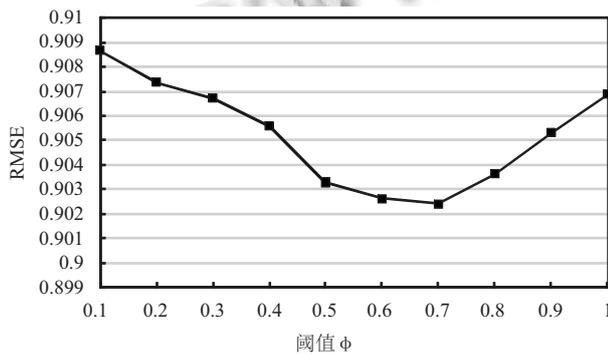
表2 Adaboost 框架效果比对

算法	100 k		1 M	
	未使用	使用	未使用	使用
SVD	0.915536	0.901825	0.805664	0.786394
PMF	0.927213	0.911898	0.821646	0.801073

3.3.2 影响训练效果的相关参数分析

虽然在上一小节的结果已经表明加入我们该框架的有效性,但是在改进 Adaboost 算法时又引入了三个参数: 阈值 ϕ , 训练轮数 T , 权重更新 μ . 因为参数数量较多, 我们使用控制变量的方法来研究不同参数对于训练结果的影响. 下面我们开始具体研究相关参数从而对得到的结果进行分析. 我们使用 SVD 模型, 在 MovieLens 100k 数据集上进行研究.

首先针对阈值 ϕ , 我们测试了其在 0.1-1.0 之间, 框架的训练效果如图 1.

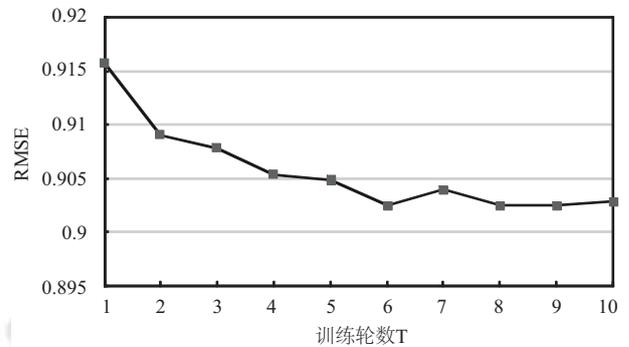
图1 阈值 ϕ 影响

由图 1 可得, 不同阈值 ϕ 对于评分预测效果还是有一定影响的, 整体呈现先下降后上升的过程. 因为阈值的作用我们在 3.2.1 节做了详细描述, 是判断我们预测评分好坏的标志, 如果我们将它设置较小, 因而会导致模型错误率太高, 从而影响权重更新过程. 同理, 如果将其设置过大, 也会导致模型错误率较小. 因而阈值的设定需要适中, 保证模型的错误率不能过大也不能过小.

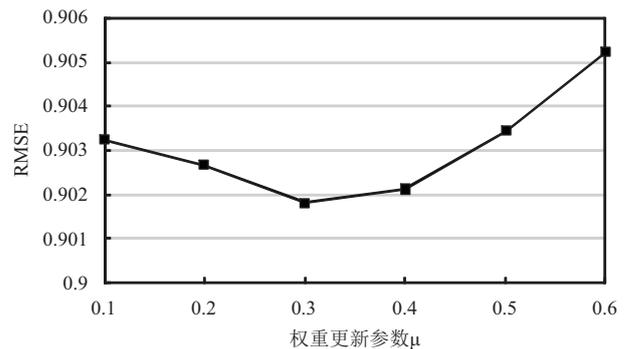
对于训练轮数 T , 我们测试了其在 1-10 之间, 框架的训练效果如图 2.

由上图可得, 随着轮数增多, 我们预测效果整体呈现下降趋势, 但随着模型的增多, 结果也会出现反复的情况. 因为我们知道, 当轮数为 1 时, 我们只有一个模型, 因而很容易理解其效果与单个 SVD 的效果一致. 随着轮数的上升, 我们的模型数量也在增多, 因而我们预测的结果会得到提升, 但是随着模型数量的进一步

增多, 其效果达到一定水平便不再继续下降, 维持一个较低的水平. 同时, 轮数的增多会增加训练时间, 所以我们训练的轮数并非越多越好.

图2 训练轮数 T 影响

对于权重更新参数 μ , 我们测试了其在 0.1-0.6 之间, 框架的训练效果如图 3.

图3 权重更新参数 μ 影响

由图 3 可得, 权重更新参数对与训练效果的影响具有很大的影响, 从图可得, 在权重从 0.1-0.3 的过程, 预测结果得到提升, 但是之后随着权重更新参数的变大, 训练结果逐渐变差.

4 结语

本文成功的将 Adaboost 算法进行改进应用于推荐系统评分预测问题, 通过引入阈值, 将评分预测问题转化为分类问题, 并利用其权重更新的思想训练模型, 提出了一个针对评分预测问题的框架, 可以将训练出的多个模型集成起来得到最终的评分预测, 提高了预测精度. 我们选取矩阵分解模型作为基本模型, 实验结果表明, 该方法预测精度与经典算法相比有所提高. 但

是从效果的提高上来看还是有继续提高的空间, 我们后续的工作会更加深入的研究 Adaboost 算法在推荐系统评分预测问题上的应用, 以求能取得更好的效果。

参考文献

- 1 Yang ZH, Xu L, Cai ZM, *et al.* Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. Knowledge-Based Systems, 2016, 100: 74–88. [doi: [10.1016/j.knosys.2016.02.008](https://doi.org/10.1016/j.knosys.2016.02.008)]
- 2 Liu Y, Zhao PL, Sun AX, *et al.* A boosting algorithm for item recommendation with implicit feedback. Proc. of the 24th International Conference on Artificial Intelligence. Buenos Aires, Argentina. 2015. 1792–1798.
- 3 Wang YH, Sun HL, Zhang RC. AdaMF: Adaptive boosting matrix factorization for recommender system. Li F, Li G, Hwang S, *et al.* Web-Age Information Management. WAIM 2014. Lecture Notes in Computer Science, vol 8485. Berlin, Germany. Springer. 2014. 43–54.
- 4 Kearns M, Valiant LG. Learning Boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88, Cambridge, MA: Harvard University Aiken Computation Laboratory, 1988.
- 5 Kearns M, Valiant LG. Cryptographic limitations on learning Boolean formulae and finite automata. Proc. of the 21st Annual ACM Symposium on Theory of Computing. New York, NY, USA. 1989. 433–444.
- 6 Schapire RE. The strength of weak learnability. Machine Learning, 1990, 5(2): 197–227.
- 7 Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997, 55(1): 119–139. [doi: [10.1006/jcss.1997.1504](https://doi.org/10.1006/jcss.1997.1504)]
- 8 Zhou ZH. Ensemble Methods: Foundations and Algorithms. Boca Raton: CRC Press, 2012.
- 9 Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, Nevada, USA. 2008. 426–434.
- 10 Harper FM, Konstan JA. The movielens datasets: History and context. ACM Trans. on Interactive Intelligent Systems (TiiS), 2016, 5(4): 19.
- 11 Takács G, Pilászy I, Németh B. Major components of the gravity recommendation system. ACM SIGKDD Explorations Newsletter, 2007, 9(2): 80–83. [doi: [10.1145/1345448](https://doi.org/10.1145/1345448)]
- 12 Salakhutdinov R, Mnih A. Probabilistic matrix factorization. Proc. of Advances in Neural Information Processing Systems 20. Vancouver, BC, Canada. 2007. 1257–1264.