利用 Tri-training 算法解决推荐系统冷启动问题

张栩晨
(复旦大学计算机科学技术学院 上海 201203)

摘 要 随着社交网络的发展,推荐系统日趋重要,而冷启动问题是推荐系统中的关键问题。设计了一种基于上下文的半监督学习框架 TSEL,对矩阵分解模型 SVD 进行扩充以支持更多形式的上下文信息,利用 Tri-training 框架训练各个模型。与其他解决推荐系统冷启动问题的半监督方法(如 Co-training)相比,该方法有着更好的效果。Tri-training 框架能够更加方便地引入更多推荐模型,具有更好的可扩展性。将 Tri-training 框架加以扩展,提出了基于用户活跃度生成无标记教学集合的算法和更加丰富的对矩阵分解模型扩充的形式。在真实数据集 MovieLens 上进行验证,获得了更好的实验效果。

关键词 推荐系统,机器学习,Tri-training

中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.019

Utilizing Tri-training Algorithm to Solve Cold Start Problem in Recommender System

ZHANG Xu-chen
(School of Computer Science, Fudan University, Shanghai 201203, China)

Abstract With the development of social network, recommender system is becoming more and more important. Cold start is one of the most important problems in recommender system. A context-based semi-supervised learning framework TSEL was designed. We expanded matrix factorization model SVD to support more kinds of context information, and used Tri-training framework to train individual models. Compared with other methods which solve cold start problems in recommender system (e.g. Co-training), our algorithm has better performance. Tri-training framework can incorporate more recommender models and has good expansibility. We expanded Tri-training framework, and proposed a user activeness-based unlabeled teaching set generating algorithm. We proposed more kinds of models which expand the matrix factorization. We evaluated our algorithm on real world dataset, i.e. MovieLens, and got better performance.

Keywords Recommender system, Machine learning, Tri-training

1 介绍

随着互联网的出现与发展,推荐在很多领域中都是极其重要的组成部分。例如,Netflix 曾做过统计,售出商品的 3/4 来自于推荐。再比如京东、淘宝等电子商务网站上,用户可以浏览和选择他们感兴趣的商品,系统会推荐给用户很多符合用户兴趣的商品。之后,用户可以给商家提供反馈,说明该商品的体验如何(例如,评分 1 星到 5 星)。在推荐系统中,一个很重要的任务就是根据用户历史评分行为来推测用户的兴趣所在。

在推荐系统中另一个有挑战的任务是对于新注册用户或者新上架的商品,在推荐与被推荐过程中如何提高推荐的准确度。如协同过滤[1]这样的传统方法,由于评分信息的缺失,很难给出很高的推荐准确度。图 1 示出了关于商品被购买数与推荐准确度在本文实验数据集上的关系(数据集 D_1',以物品被购买数分为 10 个区间,协同过滤算法详见 6.1 节)。推荐误差(RMSE 均方误差)随着物品被购买数量的减少而迅速升高。被购买最少的物品区间(10)误差达到了购买最多的物品区间(1)误差的两倍。该误差在新注册用户或者购买非常少的用户区间也尤为突出,这样的误差问题被称为冷启动,几乎所有的推荐系统都存在着冷启动问题。

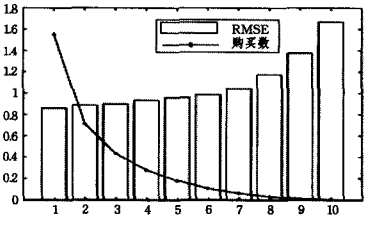

图 1 物品被购买次数与 RMSE 的关系

尽管很多研究都涉及该问题,但是冷启动问题仍然未被解决。文献[2]提出了一个概率模型,能够把很多上下文信息融合在一起。然而,该方法是基于 Bayes 分类器的,无法准确地对用户偏好建模。文献[3]使用 Twitter 中的社交信息获取用户兴趣,较好地解决了 App 推荐问题。然而,这些方法的共性是获取不同源头的信息,而不是单独地使用单个源头

到稿日期:2016-02-03 返修日期:2016-05-27

张栩晨(1990-),男,硕士生,主要研究方向为机器学习、推荐系统,E-mail:13210240075@fudan.edu.cn。

的信息。在利用不同源头信息时，不同的模型是如何相互作用以提高的呢？另一个关键问题是，获取其他源头的数据（标记样本）是受限的，那么另一条很重要的思路是使用未被标记样本，所以本文使用半监督学习算法来利用未被标记样本。

本文为了捕获用户的偏好，提出了形式更为丰富的使用用户/物品上下文信息的矩阵分解模型，而不单单使用用户对物品的评分信息。另外，本文使用半监督 Tri-training 方法构建最终的推荐模型。该方法不仅能够使用大量的未标记数据来提高推荐效果，而且能够帮助提高 3 个不同上下文模型的效果。这 3 个子模型能够使用集成学习（ensemble）融合在一起。该方法能够显著地减轻冷启动问题。

本文在 MovieLens 数据集上验证了所提算法。实验结果表明该算法在解决冷启动问题上较对比算法效果有着显著的提升。均方误差 RMSE 降低了 4%～5%，而对于冷启动用户或者物品的预测误差降低了 11%以上。综上，作为文献[20]的拓展工作，本文贡献如下：

(1)提出了能够捕获用户上下文信息的矩阵分解模型。

(2)提出了能够利用未标记数据的半监督 Tri-training 算法 TSEL。TSEL 能够引入不同源头信息构建不同的模型，能够从不同角度描绘用户/物品的特性，帮助 3 个模型相互学习并最终集成为一个推荐模型。

(3)提出了基于用户活跃度生成无标记教学集合的方法。

(4)提出了模型补充的概念。

2 相关工作

对于大多数推荐系统来说，绝大多数用户只会评价很少一部分物品，冷启动都是不可避免的[4]。对于那些刚进入系统且没有对任何物品进行过评价的用户来说尤为严重。一种传统的解决方案是对缺失数据进行默认值填充[5]，即用户或物品的评分平均值。另一种解决方案是使用上下文信息填充缺失评分[6]，例如文献[7]使用自动填充代理机器人模拟人的行为基于物品的特性对物品进行填充。另一方面，本文研究的问题在协同推荐、交叉领域推荐[8]等方面有着大量的研究。然而，冷启动问题仍然未被解决。

文献[9]使用降维的方法，将用户和物品映射到隐藏维度来捕获它们最明显的特征。文献[10]提出了在不同的应用场景里使用更新的方法来解决冷启动问题。文献[11]将推荐视为排序问题来解决。文献[12]提出在用户登录网站一开始就让用户填写大量的信息以避免冷启动问题。文献[13]同样希望用户在初始阶段填入很多信息，但不同点在于使用决策树将用户细分。文献[3]使用隐藏狄利克雷分布对用户聚类以解决 App 推荐的冷启动问题。另外，文献[21]提出了基于信任网络的推荐系统冷启动的解决算法。文献[22]提出了基于社交选择理论来解决冷启动问题的方法。以上的工作更多在特殊的领域用特殊的上下文信息解决冷启动问题，而本文工作更加具有普适性。本文所提 TSEL 框架可以适用于任何回归模型，不需要额外的信息。

本文使用的半监督学习是半监督学习中基于争论意见（disagreement-based）分类的[14]。在这类方法中，多个学习器同时训练，分类器间的差异性是训练过程的动力。所以使用半监督方法来解决推荐问题是行之有效的，但据了解，之前鲜有工作基于半监督来解决推荐问题。基于这个思路，提出了针对推荐问题的矩阵分解回归模型。在与对比算法的比较中，本文算法在冷启动问题以及总体的推荐准确率上都有着显著的提升。

3 问题与方法概述

冷启动问题定义如下。

用户集合表示为 U，物品集合表示为 I。用户 $u\in U$ 对于物品 $i\in I$ 的评分用 r_{ui} 表示，所有已知评分集合用 $L=\{r_{ui}\}\subset I\times U$ 表示。而所有未知评分集合用 $U=\{x_{ui}\}$ 表示，$U=I\times U-L$。$|\cdot|$ 表示一个集合的基数，例如 $|L|$ 表示 L 中所有评分的总数。另外，每个用户和物品有一些额外的上下文信息（context）。例如用户可能拥有年龄、职业、性别信息，物品拥有类别信息。推荐的目标是利用所有可用信息来学习一个方法 f 对未知评分进行预测。用户 u 对物品 i 的评分预测为：

$$f((u,i)|L,U,\text{other contexts})\rightarrow r_{ui}$$

本文使用标准的矩阵分解算法作为对比算法。它是由 Netflix 竞赛获胜团队提出的基于模型（model-based）的矩阵分解算法[15]。时至今日，该算法仍然是最为流行的推荐算法。该模型可以定义为如下形式：

$$\overset{\vee}{r}_{ui}=\mu+b_u+b_i+q_i^T p_u \tag{1}$$

其中，$\overset{\vee}{r}_{ui}$ 表示用户 u 对物品 i 的评分，该预测评分被分解为 4 个部分：全局平均 μ、用户偏好 b_u、物品偏好 b_i，以及表示用户对物品的个人的偏好交互项 $q_i^T p_u$（其中 $q_i, p_u\in R^k$）。通过最小化预测评分与实际评分之间的差距，可以估计式(1)中的未知参数 $\{b_u\},\{b_i\},\{q_i\},\{p_u\}$。这个模型在本文中被称为 FactMF。在图 1 中，推荐准确度随着被购买数的减少迅速下降，FactMF 仍然无法很好地解决冷启动问题。

本文提出了半监督 Tri-training 学习算法 TSEL 解决冷启动问题。算法分为两大阶段：

(1)基于上下文矩阵分解模型的构建。提出了能够引入用户/物品上下文信息的矩阵分解模型，该模型能够考虑上下文信息在用户/物品维度上的交互。

(2)半监督 Tri-training 过程。基于上一步获得的上下文模型，本文提出了一个半监督学习框架。该框架能够使不同的模型利用不同源头信息进行训练，然后模型在 Tri-training 训练过程中互相帮助提高。该框架能够应用任何模型（即不使用上下文信息），甚至无需任何其他源头的信息就可以训练。

4 基于上下文的矩阵分解模型

在推荐系统中，由于很多用户/物品的评分过少，对其品味准确估量很困难。解决这个问题的一种方式是引入额外的其他源头的信息。本文沿着这个思路提出了一个基于上下文信息的矩阵分解模型，这些信息包括用户的年龄、职业、性别、物品的类别等。这些信息相比评分信息来说，有着更强的普适性。因为评分是一个用户对一个物品的，而这些信息是针对单个用户或者物品的，能够表示一类人或者物品的特征。

本节提出的模型基于 FactMF，因为其无法引入上下文信息，所以对其进行扩展以考虑上下文信息。

第一种引入额外信息的模型(Context-1)：

$$\overset{\vee}{r}_{ui}=\mu+b_u+b_i+q_i^T p_u+\frac{\sum_{g\in genres(i)} b_{ug}}{|genres(i)|}+b_{ia}+b_{io}+b_{is} \quad (2)$$

其中，前半部分是对 FactMF(式(1))的扩展。b_{ug} 代表用户 u 对物品类别 g 的偏好程度，$genres(i)$ 代表物品 i 所有所属类别。b_{ia}，b_{is}，b_{io} 分别代表物品受用户年龄段、用户性别、用户职业的群体偏爱的程度。b_{ug}，b_{is}，b_{ia}，$b_{io}\in R^1$，视为模型上下文参数。本文使用随机梯度下降(SGD)算法学习模型参数。每次用户 u 对某物品进行评分时，若该物品属于类别 g，则 b_{ug} 会被更新。同理，当属于某一年龄段 a 的用户对物品 i 进行评分时，则 b_{ia} 会被更新。b_{is}，b_{ia}，b_{io} 代表着物品对一类人的被偏爱程度，而 b_{ug} 代表着用户对一类物品的偏爱程度。

第二种引入额外信息的模型(Context-2)：

$$\overset{\vee}{r}_{ui}=\mu+b_u+b_i+(q_i+b_{ia}+b_{io}+b_{is})^{\mathrm{T}}\cdot(p_u+\frac{\sum_{g\in genres(i)} b_{ug}}{|genres(i)|}) \quad (3)$$

其中，b_{is}，b_{ia}，b_{io}，$b_{ug}\in R^k$ 被建模成一维向量(注意 Context-1 模型中这些参数属于 R^1)，长度为 k，与 q_i，p_u 等长。物品向量 q_i 被扩展成 $(q_i+b_{ia}+b_{io}+b_{is})$，而用户向量 p_u 被扩展成 $(p_u+\frac{\sum_{g\in genres(i)} b_{ug}}{|genres(i)|})$。在 FactMF 中，$q_i^T p_u$ 被解释成用户对物品隐藏维度上偏爱的加和。以用户-电影数据为例，q_i 的第一维度可能代表物品被当成动作片而被接受的程度，而 p_u 的第一维度可能代表用户对动作片的喜爱程度。

在本模型中，当预测物品被某一年龄段用户评分时，对 q_i 进行修正，以考虑该物品在各个维度上被该年龄段用户的接受程度，例如喜欢动作片的小孩子对该物品接受程度大，而喜欢动作片的成年人对该物品接受程度小。该模型不像 Conext-1 模型在该动作片维度上不对用户年龄等上下文信息进行区分。同理对 p_u 同样进行修正建模，以更好地细化模型，达到更好的预测效果。

值得一提的是，本文所提出的两个模型都可以引入更多的用户或物品上下文信息，能够很方便地加入模型进行训练，本文只以这 4 个维度的上下文信息进行展示。

利用以上所有上下文信息，可以定义目标函数，最小化在训练集 L 中的预测误差来学习模型当中的所有参数。

$$\min_{\{b_*,q_*,p_*\}}\sum_{(u,i)\in L}(r_{ui}-\overset{\vee}{r}_{ui})^2+\lambda(b_*^2+\|q_i\|^2+\|p_u\|^2) \quad (4)$$

其中，$\overset{\vee}{r}_{ui}$ 是式(3)、式(4)中的任意模型，$\|\cdot\|$ 代表向量的模，λ 代表正则化参数。该目标函数使用随机梯度下降(SGD)进行训练。SGD 依次训练数据样本，更新模型参数。SGD 降低均方误差 $e_{ui}^2=(r_{ui}-\overset{\vee}{r}_{ui})^2$，对每一个参数 θ 进行分别更新：

$$\Delta=-\gamma\frac{\partial e_{ui}^2}{\partial}-\lambda=2\gamma e_{ui}\frac{\partial \overset{\vee}{r}_{ui}}{\partial\theta}-\lambda\theta$$

其中，γ 代表学习速率。因此，所有的参数都会向着梯度相反的方向进行更新。部分参数的更新法则如下：

$b_*=b_*+\gamma\cdot(e_{ui}-\lambda\cdot b_*)$

$p_u=p_u+\gamma\cdot(e_{ui}\cdot q_i-\lambda\cdot p_u)$

$q_i=q_i+\gamma\cdot(e_{ui}\cdot p_u-\lambda\cdot p_i)$

…

对于每个参数，本文采用不同的学习速率和正则化系数。这样在一个模型中出现次数较多的参数可以学习得更慢也更加准确。在 Context-2(式(3))中的模型得到的实验效果更好，因此在接下来的半监督 Tri-training 过程中将更多地使用这个模型。

5 半监督 Tri-training 过程

基于以上提出的基于上下文的矩阵分解模型，提出了一个半监督 Tri-training(TSEL)框架来解决推荐系统的冷启动问题。TSEL 分为 3 个步骤。

(1)构建多个回归模型。推荐预测问题是一个回归问题，所以首先需要利用训练集 L 构建 3 个回归模型 h_1，h_2，h_3。

(2)Tri-training。在 Tri-training 过程中，每一个回归模型相互学习。准确地说，两个模型对一些未标记数据进行标记之后，另外一个模型将其加入自己的训练集进行模型重新训练，3 个模型互相教学。这个过程循环往复地进行直到模型收敛。

(3)集成(ensemble)。最终形成的 3 个模型通过集成学习的方式产生最终模型。

算法 1 半监督 Tri-training(TSEL)

输入：包含标记数据的评分样本训练集 L，未标记样本集合 U

输出：回归模型 h(u,i)

步骤 1 构建多个回归模型

通过对训练集进行分割，生成 3 个回归模型 h_1，h_2 和 h_3。分割方式可以是分割样本点集，也可以是分割上下文信息(即分割模型属性)。

步骤 2 Tri-training

Repeat

生成符合样本分布的未标记集合 T_1，T_2

方法 1：基于概率的轮盘赌方法

for each $x_{ui}\in U'$ do

获取 x_{ui} 的产生置信概率 $C(x_{ui})=\frac{d_u\times d_i}{N}$ (式(9))

end

以轮盘赌算法选取 N 个样本进入教学集合 T

方法 2：基于活跃度的方法

for each ub do

for each ib do

获取每个用户-物品活跃度区间(ub,ib)占训练集合 L 的比例 percent(ub, ib)(式(10))

end

end

以活跃度随机算法选取 N 个样本进入教学集合 T

将 T 均匀分为 T_{12}，T_{23}，T_{13}。

以 h_1、h_2 对 T_{12} 标注(取预测均值)

以 h_2、h_3 对 T_{23} 标注

以 h_1、h_3 对 T_{13} 标注

互相教学过程：$L_1=L_1\cup T_{23}$；$L_2=L_2\cup T_{13}$；$L_3=L_3\cup T_{12}$；

回归模型更新：$h_1\leftarrow L_1$；$h_2\leftarrow L_2$；$h_3\leftarrow L_3$；

Until t rounds;

步骤 3　集成

$H(u,i)=\text{Assemble}(h_1(u,I),h_2(u,i))$

算法 1 给出了算法的框架。下文将介绍构建不同回归模型、构建教学集合以及集成(ensemble)的细节。该算法框架具有良好的可扩展性,能够适应大于 3 个回归模型,本文将以 3 个回归模型进行讨论。

5.1　构建多个回归模型

通常意义上,应用集成学习或应用半监督学习的前提是构建多个不同的回归模型。而构建的方式一般分为分割训练集和分割模型属性两种方式。本节将针对上述两种方式进行讨论。

(1)通过分割训练集合构建回归模型。一种常见的分割数据集的方法被称为 bagging[16]。在每一次迭代中,bagging 将数据集有放回地随机抽取 k 个训练样本。

在本文中,原始数据集通过 bagging 方式被分为 3 个有重叠的子集,以便在 Tri-training 过程中协同工作。而每个模型可以是由第 4 节提出的基于上下文的矩阵分解模型。

$$h_1(u,i)=h_2(u,i)=h_3(u,i)=\mu+b_u+b_i+(q_i+b_{ia}+b_{io}+b_{is})^{\mathrm{T}}\cdot\left(p_u+\frac{\sum_{g\in genres(i)}b_{ug}}{|genres(i)|}\right) \tag{5}$$

一些相关工作使用不同的模型来减少由 Tri-training 过程产生的标记噪声[17]。本文实验同样表明模型之间的差异性是提高 Tri-training 表现的源动力。本节中,3 个回归模型的差异产生于其训练样本的差异,并且可以从它们预测的结果来进行评估。在 Tri-training 中,3 个不同回归模型需要满足充分性(sufficient)与冗余性(redundant)的条件。其意味着每个回归模型的效果要足够好,并且拆分数据集不会对模型原效果有较大的影响。

然而,由于现实中是不存在完全满足冗余性的数据集的,差异性与充分性是相互矛盾的。如果让 3 个回归模型有着更好的初始效果,势必会降低差异性。而如果提高差异性,3 个模型的效果势必会受到影响。该方法被称为 M_s。

(2)通过分割属性构建回归模型。另一种分割方式是通过分割属性实现的,即将模型额外的属性切分成不同的视野(view)。

$$h_1(u,i)=\mu+b_u+b_i+(q_i+b_{ia}+b_{io}+b_{is})^{\mathrm{T}}\cdot p_u \tag{6}$$

$$h_2(u,i)=\mu+b_u+b_i+q_i^{\mathrm{T}}\cdot\left(p_u+\frac{\sum_{g\in genres(i)}b_{ug}}{|genres(i)|}\right) \tag{7}$$

$$h_3(u,i)=\mu+b_u+b_i+q_i^{\mathrm{T}}p_u+\frac{\sum_{g\in genres(i)}b_{ug}}{|genres(i)|})+b_{ia}+b_{io}+b_{is} \tag{8}$$

前两个模型都是 Context-2 模型(式(3))的一部分,最后一个模型是 Context-1 模型(式(2))。与 M_s 不同的是,3 个回归模型都用的是全部的训练集。为保证 3 个模型的差异性,上下文信息被拆分给前两个模型,一部分是关于用户的,一部分是关于物品的,而最后一个模型使用全部的上下文信息。由于模型参数维度上的差异,即模型的差异,前两个模型与第三个模型仍然有较大的差异性。另外,为了保证 3 个模型都有比较好的效果,$\mu+b_u+b_i$ 被保留给 3 个模型,以便满足充分性。该方法被称为 M_v。

(3)通过混合方式构建回归模型。如上文所述,M_s 用相同的模型训练不同的训练集子集,而 M_v 使用不同的属性构建模型。本节混合的方式就是既分割训练集又分割属性。这种方法将带来大量的模型间的差异性。该混合的方法被称为 M_{sv}。

5.2　半监督 Tri-training

根据上述的多个回归模型,可以应用 Tri-training 提高各个模型的效果。在 Tri-training 中,需要构建"教学集合"实现多个模型间的相互教学,这是实现半监督学习的关键步骤。一个重大的挑战是如何在大量的无标记数据集合 U 中选取合适的集合以便模型进行标记与相互教学。半监督学习的产生初衷就是标记数据特别少,为弥补不足,尽量去产生与原标记数据相近的数据帮助模型训练。而原标记数据除了体现标记外,还有一个很重要的信息即数据的分布。因此一个好的教学集合应该首先尽量去拟合原训练数据集合的分布情况。本文提出了两种方式生成教学集合以拟合原数据分布的方法:1)基于概率的轮盘赌方法;2)基于活跃度的方法。

(1)基于概率的轮盘赌方法

一个用户历史上购买的物品越多,那么他继续购买其他物品的概率越大。而一个被购买很多次的物品,也会随着购买次数的增多,使其被其他人购买的概率越来越大。基于这样的分析,构建出一个直观的概率公式:

$$C(x_{ui})=\frac{d_u\times d_i}{N} \tag{9}$$

其中,N 代表归一化参数,以使概率落在[0,1]。d_u,d_i 分别代表用户购买物品数量与物品被购买数量。

使用类似于轮盘赌的方法,首先初始化教学集合为空。对所有属于 U 的样本算出其产生概率,每次按每个样本概率大小随机产生一个样本,观察是否在 3 个回归模型训练集合或者教学集合中出现过,如果没有则加入教学集合。此过程类似于将所有 U 中样本按概率在轮盘上分配对应空间,转动轮盘产生结果,故该方法称为轮盘赌方法。

(2)基于活跃度的方法

对用户按活跃度(购买数量)进行划分,将物品按活跃度(被购买数量)进行划分。计算集合中所有(用户桶 ub-物品桶 ib)出现的频率。

$$percent(ub,ib)=\frac{\sum_{(u,i\in L)}I(u\in ub,i\in ib)}{|L|} \tag{10}$$

接下来按照比例产生无标记集合中每个桶的样本。若最活跃用户桶对次活跃物品桶的样本频率为 1/10,则产生 1/10 的该桶样本。每次产生样本时,等概率产生该用户桶内用户与物品桶内物品的(用户-物品)未标记数据,若并未在 3 个回归模型训练集合或者教学集合中出现,则放入教学集合。

值得一提的是,通过大量的实验发现,若 3 个回归模型对某一无标记数据的预测十分接近,那么教学效果是很差的。这与文献[18]提出的模型的差异性要足够大是一致的。为了保证这一性质,本文在教学集合的选择过程中,加入了过滤环节。即设定了一个阈值 t,只过滤 3 个模型中差异大的无标记样本。

得到教学集合后,将教学集合拆成等大的 3 部分 T_{12},T_{23},T_{13},让模型 h_1,h_2 对 T_{12} 进行标记(取预测均值),模型 h_2,h_3 对 T_{23} 进行标记,模型 h_1,h_3 对 T_{13} 进行标记,互相

教学。之后以 T_{12} 训练 h_3，以 T_{23} 训练 h_1，以 T_{13} 训练 h_2，达到相互学习的目的。这个过程循环往复地进行，直到教学集合的加入无法改善模型，训练过程结束。

5.3 集成(Ensemble)

算法的最后一步是集成 3 个不同的回归模型，一种显而易见的集成方法是取平均。

$$h(u,i)=\frac{1}{l}\sum_{j=1\cdots l}h_j(u,i) \tag{11}$$

在本文的方法中，$h(u,i)=\frac{1}{3}[h_1(u,i)+h_2(u,i)+h_3(u,i)]$。

集成方法可以被应用在半监督学习之后，也可以被应用在单纯的模型集成而没有半监督学习的过程中。Tri-training 中的每一轮，集成模型一般都好于任意单个模型，而半监督学习需要较为精准的预测，所以本文改用集成模型教学给 3 个模型互相教学，而不是单单“二教一”，达到了更好的实验效果。

模型补充：3 个模型若以分割训练集方式获取，则初始最为准确的训练样本缺失了一部分。若以分割属性方式获取，则与效果最好的上下文模型在模型形式上有所差距。所以在最终集成为一个模型之前，可以进行模型补充过程。对于第一种分割样本的方式，补全训练集。对于第二种分割属性的方式，全部替换成 $Context_2$，以便在集成之前获得更好的效果。值得一提的是，模型补充不论哪种方式，都是一个消耗差异性的过程，而模型集成效果的好坏取决于模型间的差异性，所以此过程相当于减少了集成过程所能带来的提高效果。通过实验发现，此过程对于实验结果有少量提升。

6 实验

6.1 实验设置

本文在公开数据集 MovieLens 上验证算法。其中较小的规模数据集包含 943 个用户，1682 件物品，用户对商品有 100000 个评分。实验中，将数据集拆成两部分，一部分是数据集初期 D_1（3 个月），包含 489 个用户对 1466 件物品的 50000 条评分。另一部分 D_2 包含全部 7 个月的数据。另外，本文算法也在 MovieLens 上的中等规模数据集上进行验证，该数据集包含 6040 个用户对 3952 件物品的 1000000 条评分。该数据集同样被拆成两部分，初期 D_1'（1429 个用户对 3266 件物品的评价）和全部 D_2'。除了评分信息外，MovieLens 还提供了上下文信息，包含 19 维度的物品类别（音乐风格）、用户的年龄（7 个阶段）、职业（20 种）、性别（M/F）信息。

实验中，10%的评分信息放入测试集，10%放入验证集，其他放入训练集。验证集的作用是学习模型中的超参数 k，然后将训练集与验证集合并供模型训练。

对比算法如下。

FactMF：矩阵分解模型，如式(1)所示。

Context-1：第 3 节提出的基于上下文的矩阵分解模型，如式(2)所示。

Context-2：第 3 节提出的基于上下文的矩阵分解模型，如式(3)所示。

$Ensemble_s$：将由分割样本方式得到的 h_1，h_2 和 h_3 进行直接合并得到的模型。

$Ensemble_v$：将由分割属性方式得到的 h_1，h_2 和 h_3 进行直接合并得到的模型。

$CSEL_s$，$CSEL_v$，$CSEL_{sv}$：将由分割样本、分割属性与两种都分割这 3 种方式得到的 h_1 和 h_2 进行半监督 Co-training 学习产生的模型[20]。

$TSEL_s$，$TSEL_v$，$TSEL_{sv}$：将由分割样本、分割属性与两种都分割这 3 种方式得到的 h_1、h_2 和 h_3 进行半监督 Tri-training学习产生的模型。

在推荐系统中，除了基于模型的矩阵分解算法以外，另一类重要的方法被称为基于内容的推荐，这类算法倾向于将用户的兴趣点与物品的属性进行匹配，侧重于 Top-N 推荐，而其对于预测问题的效果较差，因此不在对比算法之列。

实验评价标准：算法效果由预测的均方误差衡量。

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

其中，r_{ui} 是评分的真实值，而 $\overset{\vee}{r}_{ui}$ 是推荐模型的预测。

6.2 实验结果

所有算法在 4 个数据集上的总体推荐效果如表 1 所列。TSEL 算法超越了 FactMF 方法。在 4 个数据集上超越的比例分别为 4.8%(D_1)，4.8%(D_2)，4.1%(D_1')，4.4%(D_2')。其中 $TSEL_v$ 是效果最好的模型；$TSEL_{sv}$ 并没有比 $TSEL_v$ 更好，原因是 $TSEL_{sv}$ 虽然带来了更大的模型间的差异性，但导致了起始效果的大量倒退，以至于半监督过程需要事前花费更多的步骤弥补这一退步。另外，应用 Tri-training 的 TSEL 算法比应用 Co-training 的 CSEL 算法的 RMSE 最多高出 1.4%(D_2')。

表 1 4 个数据集上的总体推荐效果(RMSE)

模型	D_1	D_2	D_1'	D_2'
FactMF	0.9310	0.9300	0.9260	0.8590
$Context_1$	0.9164	0.9180	0.9140	0.8500
$Context_2$	0.9121	0.9133	0.9088	0.8456
$Ensemble_s$	0.9075	0.9083	0.9044	0.8398
$Ensemble_v$	0.9065	0.9077	0.904	0.8377
$CSEL_s$	0.9013	0.9061	0.9022	0.8375
$CSEL_v$	0.8987	0.8966	0.8963	0.8334
$CSEL_{sv}$	0.9012	0.9020	0.9000	0.8355
$TSEL_s$	0.8971	0.8993	0.8922	0.8262
$TSEL_v$	**0.8944**	**0.8961**	**0.8908**	**0.8214**
$TSEL_{sv}$	0.8976	0.8978	0.8933	0.8257

除了总体推荐效果之外，本文还在推荐系统冷启动用户与冷启动物品的推荐效果上对算法进行了衡量。将所有用户按购买数量多少等分为 10 个桶(bin)，观察每个桶上用户评分预测的效果 RMSE，如表 2 所列。通过观察靠后的几个用户与物品桶，可以观察到冷启动问题在很大程度上得到了缓解。鉴于篇幅限制，本文只展示数据集上的分桶效果，其他 3 个数据集效果类似。

与对比算法 FactMF 相比，$CSEL_v$ 在冷启动用户桶(Bin_u^{10})上得到了 8.5%的效果提升，而在冷启动物品桶(Bin_i^{10})上得到了 11.3%的效果提升。与使用 Co-training 的 CSEL 相比，冷启动用户桶提升了 0.5%，而冷启动物品桶提升了 1.5%。

表 2　在数据集 D_2 上，从最活跃用户桶(Bin_u^1)到最不活跃用户桶(Bin_u^{10})，从最活跃商品桶(Bin_i^1)到最不活跃商品桶(Bin_i^{10})，各算法的推荐效果(*RMSE*)

模型	RMSE	Bin_u^1	Bin_u^2	Bin_u^3	Bin_u^4	Bin_u^5	Bin_u^6	Bin_u^7	Bin_u^8	Bin_u^9	Bin_u^{10}
FactMF	0.9300	0.8929	0.9019	0.9100	0.9432	0.9789	0.9933	1.0291	1.0435	1.0786	1.0923
$Context_1$	0.9180	0.8797	0.8887	0.8920	0.9253	0.9503	0.9630	0.9861	1.0370	1.0573	1.0656
$Context_2$	0.9133	0.8762	0.8871	0.8889	0.9210	0.9473	0.9601	0.9835	1.0002	1.0045	1.0414
$Ensemble_s$	0.9083	0.8698	0.8825	0.8822	0.9170	0.9432	0.9588	0.9811	0.9944	0.9976	1.0366
$Ensemble_v$	0.9077	0.8700	0.8831	0.8811	0.9159	0.9433	0.9577	0.9802	0.9957	0.9971	1.0299
$CSEL_s$	0.9061	0.8779	0.8815	0.8928	0.9170	0.9282	0.9335	0.9714	0.9927	1.0054	1.0316
$CSEL_v$	0.8966	0.8694	0.8773	0.8832	0.9078	0.9215	0.9223	0.9687	0.9849	0.9918	1.0023
$CSEL_{sv}$	0.9020	0.8700	0.8752	0.8849	0.9100	0.9254	0.9367	0.9700	0.9835	0.9985	1.0153
$TSEL_s$	0.8993	0.8693	**0.8714**	0.8755	0.9011	0.9356	0.9413	0.9745	0.9814	0.9914	1.0103
$TSEL_v$	**0.8961**	**0.8670**	0.8811	0.8612	**0.8903**	**0.9211**	0.9399	**0.9723**	**0.9801**	**0.9815**	**0.9978**
$TSEL_{sv}$	0.8978	0.8699	0.8834	**0.8601**	0.8974	0.9225	**0.9353**	0.9756	0.9821	0.9843	0.9997
模型	RMSE	Bin_i^1	Bin_i^2	Bin_i^3	Bin_i^4	Bin_i^5	Bin_i^6	Bin_i^7	Bin_i^8	Bin_i^9	Bin_i^{10}
FactMF	0.9300	0.9071	0.9099	0.9300	0.9824	1.0344	1.0709	1.1066	1.1375	1.1531	1.1983
$Context_1$	0.9180	0.8871	0.9023	0.9177	0.9607	0.9980	1.0316	1.0649	1.1017	1.1171	1.1485
$Context_2$	0.9133	0.8845	0.8935	0.9120	0.9557	0.9921	1.0246	1.0559	1.0989	1.1121	1.1435
$Ensemble_s$	0.9083	0.8801	0.8924	0.9022	0.9535	0.9901	1.0121	1.0498	1.0734	1.1012	1.1399
$Ensemble_v$	0.9077	0.8814	0.8946	0.9002	0.9514	0.9856	1.0007	1.0454	1.0698	1.1011	1.1388
$CSEL_s$	0.9061	0.8821	0.8990	0.9184	0.9383	0.9572	1.0066	1.0166	1.0357	1.0661	1.0868
$CSEL_v$	0.8966	0.8741	0.8907	0.9168	0.9235	0.9561	1.0008	1.0237	1.0454	1.0592	1.0768
$CSEL_{sv}$	0.9020	0.8781	0.8920	0.9151	0.9243	0.9537	1.0076	1.0370	1.0470	1.0654	1.0800
$TSEL_s$	0.8993	0.8801	0.8896	0.8913	0.9411	0.9731	0.9893	1.0334	1.0502	1.0598	1.708
$TSEL_v$	**0.8961**	**0.8756**	0.8845	**0.8911**	0.9389	**0.9701**	**0.9891**	**1.0257**	**1.0479**	**1.0514**	**1.0621**
$TSEL_{sv}$	0.8978	0.8771	**0.8831**	0.8925	**0.9377**	0.9716	0.9899	1.0278	1.0499	1.0678	1.0688

6.3　实验分析

本节首先分析不同的部分对算法整体提升的帮助；接下来将讨论基于上下文模型、模型融合、生成未标记样本集合算法以及半监督学习对算法效果的影响；最后对算法的效率以及模型的差异性进行分析。

(1)基于上下文的模型。表 3 给出了加入更多的上下文信息后，模型效果的变化趋势。依次加入物品上下文信息 b_{i*} 与用户上下文信息 b_{u*}，而 b_{ug} 带来的效果比 b_{i*} 带来的效果更明显。因为某一用户往往对某类物品感兴趣，而同一种商品在同一年龄段或者同一职业、同一性别用户的受欢迎程度是很难一致的。

表 3　基于上下文的模型随着模型演化的效果变化(随着加入上下文信息，模型 RMSE 下降)

	模型	D_1	D_2	D_1'	D_2'
1)	$\mu+b_u+b_i+q_i^T p_u$(FactMF)	0.931	0.930	0.926	0.859
2)	$1)+(q_i+b_{ia}+b_{io}+b_{is})^T \cdot p_u(h_{v1})$	0.921	0.922	0.917	0.848
3)	$1)+q_i^T \cdot (p_u+\frac{\sum_{g\in genres(i)} b_{ug}}{\lvert genres(i)\rvert})(h_{v2})$	0.914	0.916	0.910	0.841
4)	$2)+q_i^T \cdot (p_u+\frac{\sum_{g\in genres(i)} b_{ug}}{\lvert genres(i)\rvert})$ ($Context_2$)	0.912	0.913	0.909	0.847

与对比算法相比，该上下文模型算法有了一定的效果提升。在 4 个数据集上的提升分别为 2.0%，1.8%，1.8%，1.3%。另外，context-1 模型(式(2))与 context-2 模型(式(3))相比，效果不如后者，所以不在此与对比算法(FactMF)进行对比。

(2)模型集成。模型直接进行拆分后集成的两种方法(M_s，M_v)的结果如表 1 所列($Ensemble_v$，$Ensemble_s$)，它们都强于 FactMF。另外需要强调的是，因为模型集成学习在任一时刻好于任一模型，所以将其应用在半监督过程中的每一轮，对未标注样本进行标注。这一策略带来了轻微的效果提升。

(3)生成未标记样本集合算法。本文提出了两种不同的生成未标记样本集合的算法。在实验中，基于活跃度的生成方法效果略好，实验部分所有算法皆采用此算法，本文认为此方法更能够拟合原始数据集用户对物品可能产生评分的分布。另外，此方法在算法运行时更快，对于某些已经有很高比例的样本在 3 个回归模型训练集中的用户-物品桶(往往是最活跃用户桶-最活跃物品桶)，轮盘赌算法经常随机到桶中元素而无法选出未标记数据，算法效率受到影响。而此方法只在某一桶内取一定数量的样本，可以将桶内所有未标记数据取出，随机从中取出想要数目的元素，进而不会随着半监督过程越来越慢。

(4)半监督学习过程。对于生成回归模型的过程，第一种分割数据集的方式需要参数 $t(0\leqslant t\leqslant 1)$ 控制每个回归模型训练集合所占原训练集的比例。通过大量实验，得到 t 为 0.8 时对分割数据集得到的推荐效果最好。在 5.2 节中提到，选择 h_1、h_2 和 h_3 预测差异较大的样本点进入教学集合对效果提升比较明显。因此本文设定一个阈值 t，在每轮迭代中，无论是基于轮盘赌方法还是基于活跃度的选择方法，都是随机地选择无标记样本，当阈值为 t 时，只选择 3 个模型预测平均差异大于 t 的样本，平均差异定义为任意两模型预测差距的平均值，即 3 个模型间差异(diversity)为：

$$Diversity(h_1,h_2,h_3,u,i)=\frac{|h_1(u,i)-h_2(u,i)|+|h_2(u,i)-h_3(u,i)|+|h_1(u,i)-h_3(u,i)|}{3} \tag{12}$$

随着实验的进行，寻找这样的样本越来越困难，甚至找不到，这时将 t 降低 20%。

算法效率分析：由于半监督算法使轮数增加，推荐效果提升的同时，算法效率照比单个模型有所降低。从渐进意义上讲，半监督过程有 N 轮，而总体时间复杂度将扩大到 N 倍。另外，教学集合的大小 T 对算法收敛效果与效率都有较大影响。本文发现当 T 为初始单个模型训练集大小的 20%时，能达到最好的效果。而一些较大的 T 会导致未标记数据的不准确性，从而增大负面影响，半如监督过程过早收敛，甚至无法弥补由于拆分模型带来的效果损失。而一些较小的 T 大

大增加了半监督训练的时间复杂度，并且没有得到比20%训练集更好的效果。另外，与常规矩阵分解模型算法一样，TSEL算法的训练过程是离线的。

模型差异性分析：3个模型对未知样本的预测差异性是半监督过程的源动力。较大的差异性能够带来模型的快速进步。而整个半监督过程正是通过不断消耗差异性而带来效果的提升。这与集成学习将模型拆分获得差异性再进行集成带来效果的提升在道理上是相通的。

本文3种不同生成回归模型的方法在半监督训练前后，模型间差异性的对比如表4所列(模型间的差异性定义为3个模型对所有未知样本Diversity的均值，单个样本点差异性见式(12))。由表可知，3个模型在训练前的差异性一般在0.4～0.5之间，而当半监督无法前进时，差异性降低到0.1左右。另外，在不失模型精确性的情况下，获得差异更大的3个初始模型是有可能的，比如采用拉大差异性(diversity)策略UDEED[19]，或者尝试使用完全不同的模型。

表4 不同模型拆分方法产生的回归模型 h_1, h_2, h_3 在半监督进行前、后的模型间差异性

方法	D_1	D_2	D_1'	D_2'
方法1前	0.43	0.45	0.4	0.44
方法1后	0.13	0.11	0.10	0.12
方法2前	0.45	0.5	0.5	0.51
方法2后	0.16	0.11	0.13	0.1
方法3前	0.51	0.56	0.55	0.54
方法3后	0.13	0.14	0.12	0.12

结束语 本文使用半监督方法来解决推荐系统的冷启动问题。首先，为了提高模型的精确性并且为半监督过程做铺垫，本文提出了一个基于上下文的矩阵分解模型；其次，设计了一个半监督Tri-training学习框架，它能够很好地利用数据中的未标记样本。实验结果表明，本文算法在解决推荐系统总体推荐效果与冷启动问题上都有了显著的提升。

本文所提算法还有值得深究的地方，包括设计更合理的生成未标记样本集合分布的算法、尝试更多的策略尽量维持半监督过程中的模型间差异性。另外时下流行的关于概率的推荐模型可以被应用到框架中。最后，可以扩展框架以适应更多个回归模型(如Co-forest)。

参考文献

[1] Ma H W, Zhang G W, Li P, et al. Introduction to Callaborative Filtering in Recommender System[J]. Journal of Chinese Computer Systems, 2009, 30(7): 1282-1288(in Chinese)
马宏伟，张光卫，李鹏，等. 协同过滤推荐算法综述[J]. 小型微型计算机系统，2009，30(7)：1282-1288

[2] Schein, Andrew I, Popescul, et al. Methods and Metrics for Cold-Start Recommendations[C]// Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2002: 253-260

[3] Lin J, Sugiyama K, Kan M Y, et al. Addressing cold-start in app recommendation: latent user models constructed from twitter followers[C]// Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2013: 283-292

[4] Sarwar, Badrul M, Konstan, et al. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System[C]// Acm Conference on Computer Supported Cooperative Work. 1999: 345-354

[5] Deshpande M, Karypis G. Item-Based Top-N Recommendation Algorithms [J]. ACM Transactions on Information Systems, 2004, 22(1): 143-177

[6] Degemmis M, Lops P, Semeraro G. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation[J]. User Modeling and User-Adapted Interaction, 2007, 17(3): 217-255

[7] Guo G. Improving the performance of recommender systems by alleviating the data sparsity and cold start problems[C]// Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. AAAI Press, 2013: 3217-3218

[8] Tang J, Wu S, Sun J, et al. Cross-domain collaboration recommendation[C]// Kdd. 2012: 1285-1293

[9] Takács G, Pilászy I, Németh B, et al. Scalable Collaborative Filtering Approaches for Large Recommender Systems[J]. Journal of Machine Learning Research, 2009, 10(2): 623-656

[10] Koenigstein N, Dror G, Koren Y. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy[C]// Fifth ACM Conference on Recommender Systems. ACM, 2011: 165-172

[11] Park S T, Chu W. Pairwise preference regression for cold-start recommendation[C]// RecSys 2009. 2009: 21-28

[12] Zhou K, Yang S H, Zha H. Functional Matrix Factorizations for Cold-Start Recommendation[C]// Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2011: 315-324

[13] Sun Ming-xuan, Li Fu-xin, Lee J, et al. Learning multiple-question decision trees for cold-start recommendation[C]// Proceedings of the sixth ACM International Conference on Web Search and Data Mining. ACM, 2013: 445-454

[14] Zhou Z H, Li M. Semisupervised Regression with Cotraining-Style Algorithms[J]. IEEE Transactions on Knowledge & Data Engineering, 2007, 19(11): 1479-1493

[15] Koren, Yehuda. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]// AcmSigkdd International Conference on Knowledge Discovery & Data Mining. 2008: 426-434

[16] Breiman L. Bagging Predictors[J]. Machine Learning, 1996, 24(2): 123-140

[17] Zhou Z H, Li M. Semi-Supervised Regression with Tri-training [C]// Proceedings of the 19th International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc., 2005: 908-916

[18] Rokach L. Ensemble-based classifiers[J]. Artificial Intelligence Review, 2010, 33(1): 1-39

[19] Zhang M L, Zhou Z H. Exploiting unlabeled data to enhance ensemble diversity [J]. Data Mining & Knowledge Discovery, 2013, 26(1): 98-129

[20] Zhang M, Tang J, Zhang X, et al. Addressing cold start in recommender systems: a semi-supervised Co-training algorithm[C]// International Conference on Research on Development in Information Retrieval. 2014

[21] Bellaachia A, Alathel D. Improving the Recommendation Accuracy for Cold Start Users in Trust-Based Recommender Systems [J]. International Journal of Computer and Communication Engineering, 2016, 5(3): 206

[22] Li L, Tang X. A Solution to the Cold-Start Problem in Recommender Systems Based on Social Choice Theory[M]// Intelligent and Evolutionary Systems. Springer International Publishing, 2016: 267-279