浙江大學

本科生毕业论文
开题报告

学生姓名：杨煜溟

学生学号：3130000328

指导教师：郑小林

专　　业：2013 级　计算机科学与技术

学　　院：计算机科学与技术学院

一、题目：结合标签主题的跨域推荐系统研究

二、指导教师对开题报告、外文翻译和文献综述的具体要求：

1. 文献综述要求围绕个性化推荐系统的国内外研究现状进行深入分析，阅读文献20篇以上，形成对推荐系统相关研究的深入理解，分析存在的问题。
2. 外文翻译要求选择与推荐系统相关的经典文献，翻译必须做到语句通顺，语义贴切。
3. 在此基础上，开题报告要提出跨域推荐相应的解决方案，提出可行的技术路线，以及合理的研究计划。

指导教师（签名）：

年　　月　　日

毕业论文开题报告、外文翻译和文献综述考核

导师对开题报告、外文翻译和文献综述评语及成绩评定：

成绩比例	开题报告 占 (20%)	中期报告 占 (10%)	外文翻译 占 (10%)
分值			

导师签字 _____

年　月　日

答辩小组对开题报告、外文翻译和文献综述评语及成绩评定：

成绩比例	开题报告 占 (20%)	文献综述 占 (10%)	外文翻译 占 (10%)
分值			

答辩小组负责人(签名) _____

年　月　日

目 录

本科毕业论文文献综述

摘要 推荐系统通过分析用户的兴趣模式，在互联网海量数据中为用户提供个性化的产品和内容建议，大大降低了用户检索信息的成本。推荐系统是一个热门的研究领域，越来越多的与推荐系统相关的工作被发表。良好的个性化推荐对提升用户体验有很大帮助，近年来推荐系统被成功地运用在了许多电子商务和内容提供的网站。本文将讨论近年来推荐系统领域的热门研究，并总结推荐系统中冷启动和稀疏性等关键的问题。

1 背景

随着信息技术和互联网的发展，人们逐渐从信息匮乏的时代走入了信息过载的时代，海量信息的复杂性和不均匀性使得信息获取变得困难而耗时，无论信息消费者还是信息生产者都遇到了很大的挑战。关于信息过载的问题，代表性的解决方案是分类目录和搜索引擎，分类目录只能覆盖少量内容而越来越不能满足用户的需求，搜索引擎可以让用户通过搜索关键词找到自己需要的信息，但是，搜索引擎需要用户主动对需求提供描述，当用户无法找到准确描述自己需求的关键词时，搜索引擎就无能为力了。

与搜索引擎不同，推荐系统不需要用户提供明确的需求，而是通过分析用户的过往兴趣模式，自动为用户过滤掉低相关的内容，呈现符合他们兴趣和需求的个性化建议。从某种意义上来说，推荐系统和搜索引擎是两个互补的技术，推荐系统满足了用户有明确目的时的主动查找需求，而推荐系统能够在没有明确目的时帮助用户发现感兴趣的内容 [1]。推荐算法的本质是通过一定方式将用户和物品联系起来，常用的方式有利用好友关系、用户的历史兴趣记录以及用户的注册信息等。

越来越多的网站成功地引入了推荐系统，在电子商务领域，精准的推荐使得用户能够快速准确的定位到他感兴趣的商品，提高了用户的购买效率，同时也为商家盈利带来好处；对于诸如音乐、电影、视频这样的娱乐内容，为用户推荐可能感兴趣的内容可以为内容提供商吸引更多的用户流量；互联网广告的个性化定向投放能准确的找到潜在客户群，相比于随机投放提高了效率。

推荐系统依赖于不同类型的用户行为数据，最理想的是高质量的显式反馈行为，即用户对物品兴趣的明确输入，主要的方式就是评分或单纯的喜不喜欢。通常，显式反馈产生稀疏的偏好度矩阵，因为单个用户可能只评价了一小部分物品。和显式反

馈行为相对应的是隐式反馈行为,即那些不能明确反应用户喜好的行为,例如购买商品、浏览页面、评论或甚至鼠标移动。相比显式反馈,隐式反馈虽然不明确,但数据量更大,因此利用隐式反馈缓解数据的稀疏问题。为了简单起见,我们将各种类型的反馈统称为评分(rating)。

推荐系统需要根据用户的历史行为预测未来的行为和兴趣,因此大量的用户行为数据是实现推荐系统的前提,而对于没有大量数据的情况下如何设计出让用户满意的推荐系统就是冷启动问题,冷启动问题一般分为三类:用户冷启动、物品冷启动、系统冷启动。另外,用户物品的偏好度矩阵通常是非常稀疏的,因为单个用户浏览或使用过的物品只是很小的一部分,这样的稀疏矩阵导致潜在的关联度降低,影响推荐算法对用户兴趣的建模。如何克服冷启动和数据稀疏性问题是目前推荐系统研究领域的热点。

概括地说,推荐系统主要基于两种不同的策略或其组合:基于内容的过滤方法和协同过滤方法。

内容过滤:基于内容的过滤方法为每个用户或物品创建描述以表征其性质,例如,电影的描述可以包括其类型、导演、票房等方面,用户的描述可以是个人资料或从已评分物品中识别出的共同特征。这样就可以将用户的描述做为关键词,利用信息检索的方式匹配物品的描述。这种方式的好处很明显,透明度高,推荐方式直接,而且当有新物品出现时,利用物品的描述即可进行推荐。当然,缺点是基于内容的策略需要收集额外的信息,而这些信息可能并不容易得到,同时隐私问题也可能阻碍用户提供个人信息。

协同过滤:另一种策略,不像内容过滤那样需要明确的描述信息,而是基于用户的行为分析用户的兴趣,这种方法被称为协同过滤(Collaborative Filtering)。协同过滤算法是目前推荐系统研究的热点之一,大多数推荐算法都是在此基础上改进而来。协同过滤克服了基于内容的一些限制,它比内容过滤的技术更加精确,但是却无法解决系统新用户和物品的冷启动问题,同时,稀疏且不均匀的历史数据使得分析变得困难。

预测的准确度是度量一个推荐系统预测能力的指标,计算该指标需要离线的包括用户历史行为记录的数据集,并将数据集按时间或随机地分为训练集和测试集,然后通过在训练集上建立用户的兴趣模型来预测用户在测试集上的行为,再计算预测行为和在测试集上的真实行为的重合度作为预测准确度。推荐的任务可以分为评分

预测和 TopN 推荐两种:很多网站有让用户给物品打分的功能,评分预测就是预测用户给他未评分的物品的评分,通常用平均绝对误差(MAE)和均方根误差(RMSE)来评估预测准确度 [2]。网站在提供推荐服务时一般会给用户一个个性化的推荐列表,这种推荐叫做 TopN 推荐,这种方式推荐的准确度一般利用召回率(Recall)和精确率(Precision)来度量。覆盖率(coverage)描述一个推荐系统对冷门物品的发掘能力,定义为推荐系统能够推荐出来的物品占总物品集合的比例。覆盖率对于内容提供商来说是一个重要的指标,一个好的推荐系统不仅需要有比较高的用户满意度,也要有较高的覆盖率。

推荐系统可追溯到很多相关研究领域,例如认知科学、机器学习和信息检索等。由于其与日俱增的重要性,它在 20 世纪 90 年代发展成一个独立的研究领域。在推荐的过程当中,推荐的准确性,以及推荐算法的效率等问题就是推荐算法研究的着重点 [3]。本文将介绍推荐系统目前的研究现状,从基本的协同过滤方法开始讨论,并且围绕冷启动和矩阵稀疏性两个最主要的挑战,对近些年来的热门研究成果进行综述。

2 协同过滤的研究现状

推荐系统利用数据分析技术生成用户物品的预测偏好度,为用户找到那些可能喜欢的物品,通常推荐方法都是利用用户信息、热门物品、历史行为这些数据进行预测。基于内容的过滤方法创建用户和物品的描述,以此来选择符合用户特征的物品。然而纯粹基于内容的推荐系统通常导致推荐过于局限化的问题,即无法推荐丰富多样的物品给用户,另一方面,用于生成特征描述的信息也并不容易获得。

协同过滤 (Collaborative Filtering) 这个术语最早由 David Goldberg 等人于 1992 年创造,用于描述一个实验性的邮件过滤系统 Tapestry[4]。在该系统中,每个用户可以为每个邮件编写注释并且与一组用户共享这些注释,然后,用户可以通过对这些注释进行查询来过滤这些电子邮件。尽管 Tapestry 使用户受益于其他用户的注释,但该系统仍需要用户编写复杂的查询,之后随着推荐系统的发展,出现了自动化生成推荐的技术,最早的自动推荐系统是 GroupLens,它识别相似用户的集合,并筛选该集合中的物品来获得对每个个体的建议。

协同过滤推荐是推荐系统中应用最早和最为成功的策略之一,核心是围绕用户物品的偏好度矩阵展开的,分析用户之间的关系和物品之间的相互依赖性,以获得新的用户和物品的关联。协同过滤克服了基于内容过滤的一些限制,它不使用内容信

息，而是使用系统中其他用户和项目的评分信息。此外，与基于内容的系统不同，协作过滤可以推荐多样类型的物品，只要其他用户已经对这些不同物品表现出兴趣。

随着互联网的不断发展，尤其是电子商务的出现，推荐算法随之快速发展，协同过滤在研究领域和实践中都去得了巨大成功，目前大部分推荐算法都是基于协同过滤算法改进而来。协同过滤的两个主要领域是基于邻域的方法和潜在因素模型。在基于邻域的协同过滤中，存储在系统中的用户项目评分直接用于预测新项目的评分，因此有被称为基于存储的协同过滤。基于模型的方法不直接利用存储的评分进行预测，而是使用这些评分来训练预测模型，模型的参数捕获了用户和项目的潜在特征，这些模型参数从训练数据中学习而来并用于随后的预测。

2.1 基于邻域的协同过滤

最常见的协同过滤是基于邻域的方法，该方法的基本思想借鉴了人们生活中选择物品的方式，如果身边的朋友喜欢某件物品，那么自己就会有很大概率选择该物品。另外，如果用户喜欢某个物品，那么他很可能喜欢与该物品类似的物品。因此，该方法又分为基于用户的方法和基于物品的方法。

基于邻域的协同过滤算法在用户评分矩阵并不稀疏的时候能够产生非常良好的效果，而且该算法并不需要训练，直接通过计算就可以给出推荐，但是当数据量非常庞大的时候，推荐过程伴随着大量的计算，这也从一定程度上阻碍该算法在线上系统当中的使用。

2.1.1 基于用户的协同过滤

协同过滤最初的形式是以用户之间的关系为中心 [5]，系统的实现分为两个阶段，首先在历史数据中发现那些具有相似品味的用户，然后利用邻居对物品 i 的评分计算用户 u 对一个新物品 i 的评分 r_{ui} 。假设我们计算得到了每个用户对 $u \neq v$ 之间的相似度 w_{uv} ，与用户 u 相似度最高的 k 个用户的集合，称为 u 的 k 近邻（k -NN），该集合记为 $N(u)$。然而只有评价过物品 i 的邻居才可以用来预测 r_{ui} ，因此我们将这个邻居的集合记为 $N_i(u)$ ，可以用这些邻居对 i 评分的平均值来估计 r_{ui} ：

$$\hat{r_{ui}} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{v,i}. \tag{2.1}$$

这个式子并没有考虑到邻居间可以具有不同相似度的问题，一个常见的解决方式是利用每一个邻居与 u 的相似度对评分加权。然而，如果这些权重的和不等于 1，那么预测的等级可能超出允许的范围，因此通常使用加权平均的方式进行计算。

2.1.2 基于物品的协同过滤

比较流行的是基于物品的方法 [6],基于用户的推荐依赖"志同道合"用户的观点,而基于物品的方法着重于相似物品的评分。为了使用相似性度量,首先确定 u 选择过的与 i 最相似的 k 个物品,这 k 个邻域的集合由 $N_u(i)$ 表示,预测值 r_{ui} 的计算方法是采用 u 在集合 $N_u(i)$ 中评分的加权平均:

$$\hat{r_{ui}} = \frac{\sum\limits_{j \in N_u(i)} w_{i,j} r_{u,j}}{\sum\limits_{j \in N_u(i)} w_{i,j}}. \tag{2.2}$$

2.1.3 相似性权重计算

构建基于邻域的推荐系统的最关键方面之一是相似性权重的计算,它对推荐系统的准确性和性能具有显著影响。相似性的计算基于这样的共识:相似的用户喜欢相似的物品,同时相似的物品被相似的用户喜欢。目前有多种方式用于相似性的计算,最常见的是将评分矩阵中的行列向量作为对应用户或物品的抽象,然后计算向量余弦夹角(以计算物品间相似性为例):

$$sim(i,j) = \frac{\sum\limits_{u \in U} r_{u,i} r_{u,j}}{\sqrt{\sum\limits_{u \in U} r_{u,i}^2} \sqrt{\sum\limits_{u \in U} {r_{u,j}}^2}}. \tag{2.3}$$

实际上,当使用显式评分作为偏好度时,不同的用户往往会有差异,例如某些用户的评分普遍偏高,可以使用与评分平均值的偏移作为偏好度,因此,采用这种调整的余弦相似度,物品 i 和物品 j 之间的相似度计算方法如下:

$$sim(i,j) = \frac{\sum\limits_{u \in U} (r_{u,i} - \bar{r_u})(r_{u,j} - \bar{r_u})}{\sqrt{\sum\limits_{u \in U} (r_{u,i} - \bar{r_u})^2} \sqrt{\sum\limits_{u \in U} (r_{u,j} - \bar{r_u})^2}}, \tag{2.4}$$

其中,r_u 是用户 u 评分的平均值。

2.1.4 基于用户与物品方法对比

应用基于邻域的协同过滤系统时,可以从以下几个方面考虑选择基于用户还是基于物品的方法 [5]:

1. 准确性:在电子商务系统中,用户的数量往往远大于商品的数量,数据的稀疏性会导致很难匹配到相似用户,因此推荐的精确性将收到严重影响,此时使用基于物品的方法会有更高的准确性,因为少量高置信度的邻居要比大量不那么

相似的邻居好得多。同样的，在用户数量少于物品的场景下，例如学术论文推荐系统，采用基于用户的方法会比较准确。

2. 效率: 推荐算法的计算量和存储量也取决于用户和物品的比例，当用户量远超物品数量，计算用户之间的相似性会产生庞大的计算量，影响系统的可扩展性。因为用户只对少数物品评级，因此仅存储非零的或者前 N 个相似性权重可以降低存储量和在线推荐的复杂度。

3. 稳定性:基于用户和基于物品的方法之间的选择还要取决于系统中用户和物品变化的频率和数量，一般情况下系统中可用的物品与用户相比是更加静态的，因此物品相似性权重可以不用频繁地计算，同时仍然能够向新用户推荐，这样，利用用户当前的数据可以进行实时的查询推荐，相比于基于用户的方法，系统具有更高的稳定性。

4. 可辨识性:面向物品的方法更适合解释推荐背后的原理，这是因为用户往往熟悉之前选择过的物品，而不知道那些所谓志同道合的用户。这样，可以向用户展示在当前预测中使用的邻居物品列表以及它们的相似性权重，通过修改列表或权重，用户可以交互的参与推荐过程。

2.2 基于潜在特征的协同过滤

潜在特征模型尝试从偏好度矩阵中推断出用户和物品的低维的特征向量映射，某种意义上，特征向量隐含了用户和物品在多个维度上的性质。在该模型中，用户对物品的预测偏好度是特征向量的线性结合。例如，每一个物品 i 与向量 $q_i \in R^f$ 相关联，每一个用户 u 与向量 $p_i \in R^f$ 相关联，它们的内积 $q_i^T p_u$ 表现了用户 u 对物品 i 在 f 个特征上的总体偏好度。因此评分的估计由如下式子给出:

$$\hat{r_{ui}} = q_i^T p_u.$$

这种方法最主要的挑战是如何将每一个用户和物品映射到特征向量 $q_i, p_u \in R^f$ ，在完成了映射之后，推荐系统将很容易利用上面的公式预测用户对物品的评分。潜在特征向量映射的实现通常是基于矩阵分解的，这些方法因为具有良好的可扩展性和预测精确性而变得流行。

2.2.1 基本的矩阵分解模型

奇异值分解 [7] 是一种最基本的矩阵分解方式，它的计算方式是使得到的矩阵与原始矩阵对应项的平方和误差最小。因为大多数的评分矩阵都是相当稀疏的，所

以它只关注这些很少的值会导致过拟合问题。早期通过填补矩阵中缺失的评级使矩阵变得稠密，但是随着可见项的增加，计算量可能难以承受，另外，不准确的填充会严重影响预测的效果。可以通过引入正则项缓解过拟合的问题，为了得到特征向量，系统最小化在已知评分上的正则平方误差：

$$\min_{q^*,p^*} \sum_{(u,i)\in} \left(r_{u,i} - q_i^T p_u\right)^2 + \lambda(||q_i||^2 + ||p_u||^2), \tag{2.5}$$

这里，κ 是训练集中所有已知评级的用户物品对 (u,i) 的集合，系统通过拟合之前观测的样本来学习模型的参数，而我们的目标是预测未知的评分，所以应该通过正则化参数来避免过度拟合已知的项，常数 λ 用于控制正则化的程度。

可以通过随机梯度下降或迭代最小二乘的方法最小化上面的式子。

2.2.2 随机梯度下降

随机梯度下降算法（stochastic gradient descent）最优化理论里最基础的优化算法，它首先通过求参数的偏导数找到函数的最速下降方向，然后通过不断迭代优化参数直至收敛。上面定义的损失函数里有两组参数 p_u 和 q_i ，对它们分别求偏导数，然后梯度相反的方向以 步长调整参数，可以得到如下的迭代公式：

$$\begin{aligned} q_i &\leftarrow q_i + \cdot(e_{ui} \cdot p_u - \lambda \cdot q_i) \\ p_u &\leftarrow p_u + \cdot(e_{ui} \cdot q_i - \lambda \cdot p_u) \end{aligned} \tag{2.6}$$

2.2.3 交替最小二乘

如果我们固定正则平方误差式子中的一个未知项，那么问题就转化成了二次函数求最值的问题。因此，交替最小二乘方法交替的固定 q_i 和 p_u ，当所有的 p_u 被固定，系统利用最小二乘法重新计算 q_i 的值，反之亦然。这个方法确保每一次都使之下降直至收敛。

通常情况下，随机梯度下降比交替最小二乘要简单也更快，但是在允许并行的系统下，交替最小二乘可以表现出很强的并行性，因为系统计算每一个 q_i 是独立于其它物品向量计算的，并且计算每一个 p_u 也是独立于其它用户向量的。另一个方面，当基于隐式反馈的数据时，训练矩阵中样本不再稀疏，如果使用随机梯度下降遍历所有样本项将是不可行的。

2.2.4 加入偏移量

基于矩阵分解方法的协同过滤的优点是可以很方便地处理不同种类的数据和一些应用中特定的需求 [8]，这需要在之前的学习框架上作出调整。观察到的样本数据

中通常会存在用户和物品个体性的偏移，例如一些用户给分普遍偏高。因此，直接将 $q_i^T p_u$ 视为最后的评分值是不明智的，系统尝试为每个用户和物品设置偏移量，并将偏移量的近似值加入到 r_{ui} ：

$$b_{ui} = \mu + b_i + b_u,$$

其中，μ 是全部评分的平均值，参数 b_i 和 b_u 是在样本上用户 u 和物品 i 相对于平均值的偏移量，加入偏移后改写为如下形式：

$$\hat{r_{ui}} = \mu + b_i + b_u + q_i^T p_u.$$

这样，观察到的评级被分为了四个组成部分：全局平均、用户偏移、物品偏移、用户物品匹配。这使得每一个部分可以单独解释其含义，系统通过最小化平方误差来学习模型参数：

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} \left(r_{u,i} - \mu - b_i - b_u q_i^T p_u\right)^2 + \lambda(||q_i||^2 + ||p_u||^2 + {b_i}^2 + {b_u}^2),$$

2.2.5 概率矩阵分解

基本的矩阵分解算法如奇异值分解没有办法处理庞大的数据量，而且在矩阵很稀疏的情况下的表现也不理想。概率矩阵分解 [9] 为协同过滤的评分矩阵引入了一种概率模型的表示，它假设用户和物品的潜在特征向量均服从高斯分布，它们由如下过程取样产生：

1. 对于每一个用户 i ，选取用户特征向量 $u_i \sim N(0, {\lambda_u}^{-1} I_K)$.
2. 对于每一个物品 j ，选取物品特征向量 $v_j \sim N(0, {\lambda_v}^{-1} I_K)$.
3. 对于每一个用户物品对 (i, j) ，选取打分：

$$r_{ui} \sim N({u_i}^T v_j, {c_{i,j}}^{-1}).$$

其中，$c_{i,j}$ 作为正态分布的方差控制 r_{ui} 的准确度。

概率矩阵分解的复杂度与样本数呈线性关系，可以在庞大且稀疏的数据集上表现良好。

2.3 隐式反馈数据集上的协同过滤

推荐系统的任务是利用先前的用户反馈数据进行个性化建议来改善客户体验。系统依赖不同种类的输入数据，最方便的就是高质量的显式反馈，包括用户对产品的

兴趣的明确输入，例如，用户对物品的打分、或者仅仅是喜欢或不喜欢的标记。然而显式反馈并不可用，所以可以从更丰富的隐含反馈推断用户偏好，通过观察用户行为间接反映兴趣特征。

这些系统大量跟踪不同类型的用户行为，如购买记录、观看习惯和浏览时间，以模拟用户偏好。与显式反馈明显的不同，我们没有任何来自用户的关于他们的偏好的直接输入，特别是我们缺乏关于用户不喜欢什么物品的实质证据。我们把推断的用户物品偏好作为偏好度矩阵，其中的项 r_{ui} 可以是用户 u 购买物品 i 的次数，也可以是 u 浏览网页 i 的时间，找到合适的偏好度计算方式是基于隐式反馈推荐的关键。

Florham Park 等人提出了隐式反馈上的置信度模型 [10]，该方法将原始的观测值（r_{ui}）转换为两个独立的维度：偏好（p_{ui}）和置信等级（c_{ui}），这更好的反映了隐式反馈的特性。其中，用户 u 对物品 i 的偏好 p_{ui} 是通过二值化 r_{ui} 生成的：

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases} \tag{2.7}$$

也就是说，如果用户 u 访问过物品 i ，那我们可以说有迹象表明 u 喜欢 i ($p_{ui} = 1$)，另一方面，如果用户 u 从未访问过物品 i ，那我们觉得他不喜欢这个物品 ($p_{ui} = 0$)。但是我们信念应该与置信度相关联，因为零值并不一定意味着用户不喜欢这个物品，可能还有其他的一些原因，例如，用户可能不知道这个物品的存在或者由于其价格过高而不能消费它。另外，购买了一个物品也可能是很多不同因素的结果，例如，他可能购买后发现不喜欢这个物品。因此，我们推断用户偏好的项也有其置信等级。一般来说，随着 r_{ui} 增长，我们有更强的信心推断用户喜欢该物品，因此引入一组变量 c_{ui} 来衡量对推断的 p_{ui} 的信心，一种合理的选择是：

$$c_{ui} = 1 + \alpha r_{ui}$$

这样，对于每一个用户项目对，我们对 p_{ui} 有一些最小的信心，当我们观察到更多的正向偏好的证据时，我们对 $p_{ui} = 1$ 的信心就会增加，增加的速率由常数 α 控制。

我们的目标是为每个用户 u 找到一个向量 x_u ，以及为每一个物品 i 找到一个向量 y_i ，使得它们的内积 $p_{ui} = {x_u}^T y_i$ 表示用户对物品的偏好，本质上，这些向量尝试将用户和物品映射到共同的潜在特征空间，使得它们可以直接比较。这与基于显式反馈的矩阵分解技术相类似，但是有两个重要区别：(1) 我们需要考虑变化的置信等级。(2) 训练阶段要考虑所有的用户项目对，而不只是那些可见的样本。因此，通过

最小化如下的估价函数来计算模型的参数:

$$\min_{x^*,y^*} \sum_{(u,i)\in} c_{ui}(p_{u,i} - x_u^T y_i)^2 + \lambda(||x_u||^2 + ||y_i||^2), \tag{2.8}$$

$\lambda(||x_u||^2 + ||y_i||^2)$ 项用于正则化模型,以防止过拟合训练数据。λ 的值是根据不同的数据来确定的。

我们注意到估价函数中包含 $N * M$ 项,M 是用户数量,N 是物品数量,对于典型的数据集,$N * M$ 可以很轻松达到几十亿,这个庞大的数量限制了常见的显式反馈数据集的训练方法,例如随机梯度下降。利用变量的代数结构,可以通过线性时间内遍历所有用户项目对来最优化估价函数。

3 热门研究方向

3.1 混合推荐模型

我们知道数据稀疏性和冷启动问题是协同过滤方法的挑战,一种方法是建立完全基于用户和项目特征的预测模型,这种方法不会受到冷启动问题影响,在一些应用中,用户和项目都与一组信息特征相关联。例如,用户可以在注册时提供诸如年龄、性别、职业等个人信息,当项目是电影时,我们可以知道他们的类型、导演、演员等,对于新闻项目,我们可以从文章内容中提取特征。然而,这种完全基于内容的推荐不利用过去的交互数据,它也不能捕获存在于用户项目中的相关性 [11]。事实上,忽略特征而仅依赖用户与项目过去交互的协同过滤模型,对于旧用户和项目表现出良好的预测准确性。因此,最有吸引力的方式是混合过去的交互数据和特征,并且平滑地处理冷启动和热启动场景间的过度 [12]。

矩阵分解的一个优点是它允许在特征向量中加入附加信息。Wang and Blei [11] 提出了协同主题模型(CTM)用于学术文章的推荐,该方法使用两种类型的数据:用户的收藏历史和文章的内容,结合了基于潜在因素模型 [12, 9] 的协同过滤的思想和基于概率主题模型的内容分析 [13, 14]。对于每个用户,可以推荐类似用户收藏的旧文章和其内容反映用户的特定兴趣的新文章。潜在因素模型适合推荐已知文章,但不能推荐新加入的文章。为了推荐新文章,该算法使用主题模型,主题模型发现文章的潜在主题的表示,该组件可以推荐具有与用户喜欢的文章相似内容的其它文章,在没有评分的情况下,文章的主题表示使得算法可以对文章做出有意义的推荐。

3.1.1 概率主题模型

主题建模算法用于从大量文档集合中发现一组主题，其中主题是关于词项的分布，主题模型提供了文档的低维表示 [14]。最常见的主题模型是隐式狄利克雷分布（LDA）[13]，假设有 K 个主题 $\beta_{1:k}$，每一个是在固定词典上的分布。LDA 生成文档的大致流程如下：对于语料库中的每一篇文档 w_{jn} ：

1. 从狄利克雷分布中选取主题分布 $\theta_j \sim Dirichlet(\alpha)$.
2. 对于文档中的每一个词 n ：
   (a) 选取主题 $z_{jn} \sim Mult(\theta_j)$.
   (b) 选取单词 $w_{jn} \sim Mult(\beta_{z_{jn}})$.

这个过程说明了文档中的每个词是如何从主题的集合中选取出来的：主题分布是文档特有的，但是主题的集合是整个语料库共享的。

LDA 属于非监督学习的范畴，给定一个文档语料库，我们可以使用变分 EM 算法来学习主题并根据它们给文档分配主题 [13]。此外，给定一个新的文档，我们可以使用变分推理来确定其内容的主题。

3.1.2 协同主题回归

协同主题回归（CTR）模型结合了传统的协同过滤与主题模型，最简单的方法是直接使用主题分布表示可见的评分和单词，例如，我们可以使用主题分布 θ_j 替代公式（8）中的物品潜在向量 v_j ：

$$r_{ui} \sim N({p_u}^T \theta_j, {c_{i,j}}^{-1}).$$

这个模型的局限是它不能区分文档内容对不同用户的偏好，例如，两篇文档的主题分布相同，但是内容针对的用户群体不同。协同主题回归可以发现这种区别，该方法用对主题的兴趣表示用户，并且假设文档由主题模型生成。CTR 还包括一个隐式变量 ε_j 来调整主题分布 θ_j 在建模用户评分时的比例，预测时依赖内容和依赖协同过滤的比例由用户打分的数目来决定。CTR 的生成过程如下：

1. 对于每一个用户 u ，选取用户潜在向量 $u_i \sim N(0, {}_u{}^{-1} I_K)$.
2. 对于每一个物品 j ，
   (a) 选取主题分布 $\theta_j \sim Dirichlet()$.
   (b) 选取物品隐式偏移 $\varepsilon_j \sim N(0, {}_v{}^{-1} I_K)$ ，并且设置物品隐式向量为 $v_j = \theta_j + \varepsilon_j$.

(c) 对于文档中的每一个词 w_{jn} ：

i. 选取主题 $z_{jn} \sim Mult(\theta_j)$.

ii. 选取单词 $w_{jn} \sim Mult(\beta_{z_{jn}})$.

3. 对于每一个用户物品对 (i, j) ,选取打分：

$$r_{ui} \sim N(u_i{}^T v_j, c_{i,j}{}^{-1}).$$

CTR 的关键在于物品向量 v_j 如何生成,我们看到 $v_j = \theta_j + \varepsilon_j$,其中 $\varepsilon_j \sim N(0, {}_v{}^{-1} I_K)$,就等价于 $v_j \sim N(\theta_j, {}_v{}^{-1} I_K)$,因此物品向量 v_j 接近于主题分布 θ_j 。注意到 r_{ui} 的期望是 θ_j 的线性函数：

$$E[r_{ui}|u_i, \theta_j, \varepsilon_j] = u_i{}^T(\theta_j + \varepsilon_j).$$

因此这个模型被称为协同主题回归。CTR 模型很好的利用了内容信息,并将其结合到了传统的协同过滤算法中,使得系统在物品冷启动问题上表现良好。

3.2 跨域推荐系统

用户和物品的冷启动问题是推荐系统的固有限制,前文所述的 CTR 模型利用内容信息缓解了内容的冷启动问题,而对于新用户的冷启动问题往往不容易解决,因为新用户的信息通常不容易收集。为了解决它,跨域推荐系统利用辅助域中的用户反馈来协助目标域上的推荐任务 [15]。

现有的推荐系统大多是仅针对属于单个域内的用户物品进行预测推荐,即如果我们给用户推荐电影,则只考虑用户对电影的评分记录;同样给用户推荐音乐,则只考虑用户对音乐的评分记录,因此是在单一域上的建模。事实上,用户在不同域中的偏好之间可能存在依赖性和相关性,例如,直观上来看,喜欢摇滚音乐的用户很可能喜欢科幻片,喜欢抒情音乐的用户或许喜欢爱情片。因此,在一个域中获得的用户兴趣特征可以在几个其他域中传递和利用,而不是独立地处理每种类型的项目。虽然跨域推荐的效果可能不如在单一域上的推荐准确,但跨域推荐将更加多样化,这可能对提高用户的满意度和参与度有好处 [16]。

Leizou [17] 提出跨域推荐的三个主要研究趋势:

1. 通过集成和利用分布在不同系统中的显式用户偏好。
2. 通过记录用户的行为和反应来描述用户特征,并利用这些信息生成多个域上的推荐。

3. 通过组合来自不同域的推荐来生成单一域上的推荐。

当两个域之间的用户和物品存在重叠时，我们可以直接将所有的信息看作属于一个共同的域，这样就可以利用传统的协同过滤进行跨域推荐。然而，当两个域没有重叠或重叠很小时，这种方法就会产生问题。为了解决上述不重叠的情况，我们必须找到某种方法，能够在域之间找到或建立某种类型的显式或隐式关系，其将被用作在推荐系统中连接不同域的语义桥 [16]。

跨域推荐的主要任务是在域之间找到或建立某种类型的关系的方法，这种联系将被用于连接不同域的语义桥。通常所考虑的域之间看上去是不相关的，例如，音乐与感兴趣的地方，使得难以找到它们之间的关联。

3.2.1 迁移学习模型

一种方法是在两个不同的域中映射用户的潜在特征向量，对于一个用户 a ，假定他在目标域的潜在特征向量是 U_a ，辅助域中的为 U'_a ，我们的目标是找到它们之间的映射函数，使得它们可以相互转换以提高两个域的效果。直观上，理想的情况是找到 U_a 和 U'_a 之间的可逆映射函数，但是有时候这种关系是非线性的，在这种情况下不能找到可逆函数 [15]。因此，我们希望找到两个映射，$f(U'_a) \approx U_a$ 和 $g(U_a) \approx U'_a$ 。这样，一个域的用户特征向量可以被转换，用于推断另外一个域的用户特征向量。迁移学习是机器学习领域的热门研究课题，它的目标是通过利用其它域中已知的信息来提高一个特定域的学习效果。在迁移学习的框架下，学习过程的每一次迭代，先利用随机梯度下降等算法更新单个域中的参数，然后利用域间的映射函数进行估计并得到误差，以最大后验概率的方式调整映射函数的参数。实际中，为了降低噪声干扰和复杂性，通常只利用那些在两个域都有密集活动的用户集合来训练模型。

3.2.2 基于语义桥的跨域推荐

利用社交标签和语义信息可以建立不同域之间的桥梁 [18, 19]，因为不同域中使用的标签词汇之间的通常是重叠的 [1]，该方法的好处是可以在没有共享用户的情况下将辅助域和目标域间建立起联系 [20]。

4 总结及展望

本文介绍了协同过滤推荐算法的主要思想及算法面临的主要问题，总结了近些年来的热门研究成果，讨论了针对数据稀疏性和冷启动问题的解决方案。

研究者提出了多种方法来解决协同过滤算法面临的数据稀疏性和冷启动问题，

一个明显的趋势是将其他领域的一些技术用到推荐系统中，例如利用迁移学习的方法实现跨域推荐模型，这些方法一定程度上缓解了这些问题，但是随着电子商务等平台的迅速发展，这些问题将进一步凸显，将领域外的一些技术与推荐算法结合起来解决面临的问题将是一个有意义的发展方向。

参 考 文 献

[1] 项亮. *推荐系统实践*. 人民邮电出版社, 2012.

[2] 王国霞 and 刘贺平. 个性化推荐系统综述. *计算机工程与应用*, 48(7):66–76, 2012.

[3] 肖力涛. 基于隐式因子和隐式主题的跨域推荐算法研究. Master's thesis, 浙江大学, 2016.

[4] David Goldberg. Using collaborative filtering to weave an information tapestry. Communications of the Acm, 35(12):61–70, 1992.

[5] Christian Desrosiers and George Karypis. A Comprehensive Survey of Neighborhood-based Recommendation Methods. 2011.

[6] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In International Conference on World Wide Web, pages 285–295, 2001.

[7] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In Proceedings of KDD cup and workshop, volume 2007, pages 5–8, 2007.

[8] Y Koren, R Bell, and C Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

[9] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In International Conference on Machine Learning, pages 880–887, 2007.

[10] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit

feedback datasets. In Eighth IEEE International Conference on Data Mining, pages 263–272, 2008.

[11] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, Usa, August, pages 448–456, 2011.

[12] Deepak Agarwal and Bee Chung Chen. Regression-based latent factor models. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 19–28, 2009.

[13] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.

[14] Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. Advances in Neural Information Processing Systems, 32:288–296, 2009.

[15] Xin Xin, Zhirun Liu, Chin Yew Lin, Heyan Huang, Xiaochi Wei, and Ping Guo. Cross-domain collaborative filtering with review text. In International Conference on Artificial Intelligence, pages 1827–1833, 2015.

[16] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. In Spanish Conference on Information Retrieval, 2012.

[17] Antonis Loizou. How to recommend music to film buffs: Enabling the provision of recommendations from multiple domains. University of Southampton, 2009.

[18] Manuel Enrich, Matthias Braunhofer, and Francesco Ricci. Cold-start management with cross-domain collaborative filtering and tags. 152:101–112, 2013.

[19] Chaochao Chen, Xiaolin Zheng, Yan Wang, Fuxing Hong, and Deren Chen. Capturing semantic correlation for item recommendation in tagging systems. In AAAI, pages 108–114, 2016.

[20] Yue Shi, Martha Larson, and Alan Hanjalic. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. In International Conference on User Modeling, Adaptation, and Personalization, pages 305–316. Springer, 2011.