浙江大学

本科生毕业论文

文献综述和开题报告

学生姓名　　徐帅

学生学号　　3140104542

指导教师　　郑小林

年级与专业　　2014级计算机科学与技术专业

所在学院　　计算机学院

一、题目：基于深度学习和矩阵分解的推荐算法研究

二、指导教师对文献综述和开题报告的具体要求：

1）文献综述要求围绕基于深度学习的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，形成对深度学习在推荐系统应用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基于深度学习的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，开题报告要求提出基于深度学习的推荐算法研究方面，相应的解决方案，研究目标和研究方案要明确，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

目　录

目　录 3
一、文献综述 5
1. 背景介绍 5
2. 传统推荐系统 6
2.1 主要算法介绍 7
2.1.1 基于内容的推荐 7
2.1.2 协同过滤推荐 7
2.1.3 混合推荐 7
2.2 传统推荐算法存在的问题 7
3. 基于深度学习的推荐系统 8
3.1 深度学习技术 8
3.1.1 多层感知机（Multilayer Perceptron, MLP） 8
3.1.2 自动编码器（Autoencoder, AE） 9
3.1.3 卷积神经网络（Convolutional Neural Network, CNN） 10
3.1.4 循环神经网络（Recurrent Neural Network, RNN） 10
3.1.5 受限制玻尔兹曼机（Restricted Boltzmann Machine, RBM） 11
3.2 深度学习在推荐领域的研究现状 12
3.2.1 深度学习在基于内容的推荐系统中的应用 12
3.2.2 深度学习在协同过滤中的应用 13
3.3 深度学习应用在推荐系统上的优势和存在的问题 14
3.3.1 优势 14
3.3.2 存在的问题 15
4. 总结与展望 15
5. 参考文献 15
二、开题报告 19
1. 问题提出的背景 19
1.1 背景介绍 19
1.2 推荐算法的研究现状 19

1.3 本研究的意义和目的 20
2．论文的主要内容和任务 20
2.1 主要研究内容 20
2.2 研究任务 21
3．技术路线 22
3.1 本卷积神经网络（Text-CNN） 22
3.2 异值分解（SVD） 24
3.2.1 基本的奇异值分解 24
3.2.2 带偏置的奇异值分解 24
3.2.3 加入时间动态的奇异值分解 25
4．可行性分析 26
5．预期目标及研究计划进度安排 27
5.1 预期目标 27
5.2 进度安排 27
6．参考文献 27
三、外文翻译 29
四、外文原文 45

一、文献综述

1. 背景介绍

在信息爆炸性增长的今天，个性化推荐系统已经成为人们生活中必不可少的筛选利器，并以多种多样的形式影响着人们生活的方方面面：亚马逊、淘宝等电商网站利用推荐引擎实时提供用户可能感兴趣的商品推荐；Twitter、Facebook等社交网站利用推荐系统为用户寻找潜在的好友推荐；Yotube、优酷等视频网站在用户观看视频的同时，利用推荐系统为用户提供最可能点击的视频推荐；Quora、今日头条等新闻门户网站则利用推荐系统帮用户筛选出最有价值的新闻。互联网广告利用推荐系统准确的找到潜在客户群，相比于随机投放提高了效率。推荐系统在促进商业发展以及便利人们决策等方方面面发挥着越来越重要的作用。

通常来说，推荐的内容列表一般是根据用户偏好、物品特征、用户与物品交互的历史记录以及一些诸如时间和空间数据这样的辅助信息而生成的。推荐系统的模型主要可以分为基于协同过滤的推荐、基于内容的推荐以及基于不同输入数据的混合推荐[2]。然而这些模型在解决数据稀疏性（用户已评分的物品占总物品数量的很少一部分），冷启动问题（新的用户和新的物品往往没有评分数据)以及如何权衡不同指标下的推荐效果等问题上都有其各自的局限性[2][3][4][5]。

在过去的十年间，我们已经见证了深度学习(Deep Learning)在诸如计算机视觉以及语音识别等领域的应用上，取得了显著的成功。而深度学习在解决很多复杂任务的良好效果，也促使学术界和工业界争相将其应用到其他更广阔的领域。最近，深度学习已经被逐渐应用到了推荐系统上[1][6]，改变了传统推荐系统的架构，并且在提升用户体验和满意度方面取得了令人瞩目的效果。一方面，通过学习一种深层次的非线性网络结构，深度学习可用来表征用户和物品相关的海量数据,利用这种强大的从样本中学习数据集本质特征的能力，可以获得用户和物品的深层次的特征表示。而另一方面，深度学习还能从诸如上下文、文本、图像等繁杂的数据中提取出数据之间错综复杂的内在关系，从而将不同数据映射到一个相同的隐空间，能够获得数据的统一表征[7]，在此基础上结合传统推荐方法来做推荐，可以有效利用多源异构数据，缓解传统推荐系统中的数据稀疏和冷启动问题。

推荐系统是工业领域中必不可少的一部分,在许多在线网站和手机应用中都为提升销量和服务发挥着不可忽视的作用。举例来说，人们在Netflix中所观看的影片有80%是来自于推荐[8],在YouTube上的视频点击有60%是来源于推荐[9]。最近，像Netflix、YouTube这样的公司也借助深度学习来提高推荐的质量[6][1][10]。比如Covington等人就提出了一种应用在YouTube上的借助于深度神经网络的视频推荐算法[1]。Cheng提出了一种应用在Google play App上的深广学习(wide&deep)模型[6]。Shumpei则提出了一种应用在Yahoo新闻上的基于RNN的新闻推荐系统[10]。这些模型都已经在线上进行了测试，并且可以大幅度提高传统模型的推荐性能。可以预见的是，深度学习将引领推荐系统领域的一次巨大变革。

另一项令人瞩目的改变则发生在学术研究领域。近些年，基于深度学习的推荐算法的论文数量呈现出几何倍数的增长。

作为推荐系统领域的顶级会议，ACM推荐系统年会（ACM RecSys）,在2016年就专门召开了第一届基于深度学习的推荐系统研究专题研讨会（DLRS'16），旨在促进基于深度学习的推荐系统在学术研究和工业应用的发展。由此可见，基于深度学习的推荐系统研究已经成为研究推荐系统的热点方向之一。

总而言之，不管是在学术研究还是工业应用，深度学习在推荐领域都已经取得了令人振奋成果。而要想在这一领域继续有所突破，理解和归纳前人研究模型的优缺点，必然是不可缺少的一项工作，而这也正是本文写作的目的所在。

2. 传统推荐系统

自从 20 世纪 90 年代，协同过滤技术被首次提出，推荐系统已经成为了一门独立的学科而受到了广泛的关注。作为推荐系统的核心，推荐算法主要是利用用户与物品之间的二元关系,基于用户历史行为记录或相似性关系来帮助发现用户可能感兴趣的项目。文献[2]给出了推荐算法的形式化定义：用U表示所有用户（user）的集合，用I来表示所有物品(item)的集合。在真实系统中,U和I的规模通常非常大。这里我们定义一个效用函数s,用来计算物品i对用户u的推荐度，即$s:U\times I\rightarrow R$，其中R是一个全序的集合，而推荐算法所研究的主要问题就是通过推荐度的计算，为每一个用户$u\in U$找到最感兴趣的物品$i'\in I$，如下所示：

$$\forall u\in U, i_{u'}=arg\max s(u,i) \qquad i\in I$$

在推荐系统领域，我们所需要解决的一个关键问题是如何将效用函数s从$U\times I$的一个子空间外推到整个$U\times I$空间。一般来说，我们用用户对物品的评分来表示推荐度，而在真实的推荐系统中，用户往往只评分了其中的一小部分项目。因此，在推荐给用户最感兴趣的物品之前,必须先根据已知评分对未知评分进行预测,这就是我们之前所说的外推过程。

根据对未知评分预测方法的不同，传统的推荐算法通常可以分成以下三种[11]：基于内容的推荐（content-based recommendation）、协同过滤推荐（collaborative filtering recommendation）和混合推荐（hybrid recommendation）。

2.1 主要算法介绍

2.1.1 基于内容的推荐

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。这样就能计算出用户与待测物品在内容上的匹配度，进而根据匹配度对所有需要预测的物品进行排序，最终为用户推荐其潜在感兴趣的物品。

2.1.2 协同过滤推荐

协同过滤利用相似用户之间具有相似兴趣偏好的规律，来发现用户对物品的潜在偏好。举个例子来说，基于用户的协同过滤算法（User-based Collaborative Filtering）会搜索那些与目标用户具有相似评分模式的用户作为相似用户,然后利用这些兴趣相投的相似用户的评分来预测其他未评分物品的评分。而基于物品的协同过滤算法（Item-based Collaborative Filtering）则是从物品的角度出发来预测评分。它首先计算一个物品-物品的相似矩阵，来确定每一对物品之间的相似度。然后再根据目标用户对相似物品已有的评分，来预测目标用户对未评分产品可能的评分。

2.1.3 混合推荐

混合推荐通过组合不同推荐算法，往往能相比单一推荐算法，产生更好的性能。举例来说，可以以协同过滤算法为框架，结合基于内容的推荐算法来环节数据稀疏问题，这属于模型层面上的混合推荐。也可以将所有用户属性、用户行为等数据作为输入，通过训练一个统一的分类器来产生推荐结果，这属于特征层面上的混合推荐。

2.2 传统推荐算法存在的问题

协同过滤算法是目前应用最为广泛的推荐算法，但往往面临数据稀疏和冷启动等问题。此外，经典的协同过滤算法采用的浅层模型往往无法学习到用户和物品更深层次的特征。

基于内容的推荐算法可有效缓解数据稀疏和新物品的冷启动问题，但相应的，这种算法依赖于有效的特征提取，而传统的浅层模型往往需要人工设计特征，其有效性和可扩展性非常有限。近些年来，蕴含用户丰富个性化需求信息的文本、图像、标签在内等多源异构数据被越来越多地加入到了传统的推荐算法中。

而融合了多源异构辅助信息（side information）的混合推荐算法也越来越被人们所重视，在解决数据稀疏和冷启动问题上具有不错的效果。但是由于辅助信息往往具有多模

态、数据异构、大规模、数据稀疏和分布不均匀等复杂特征，融合多源异构数据的混合推荐方法研究依然面临着严峻的挑战[12][13]。

3. 基于深度学习的推荐系统

3.1 深度学习技术

深度学习属于机器学习的一个子领域，它从数据中学习到不同层次的数据表示和抽象，可以用来解决有监督和无监督学习任务[14]。本节中，我们主要介绍一些和推荐系统领域密切相关的深度学习技术。

3.1.1 多层感知机（Multilayer Perceptron，MLP）

多层感知机（MLP）是一种前馈神经网络，与传统的只有输入层和输出层的感知机不同，它在输入和输出层之间加入一个或多个隐藏层。因此，多层感知机可以灵活应用激活函数（activation function）使其不仅仅局限于二分类问题。
这里以一个最简单的只含有一个隐藏层的 MLP 为例：

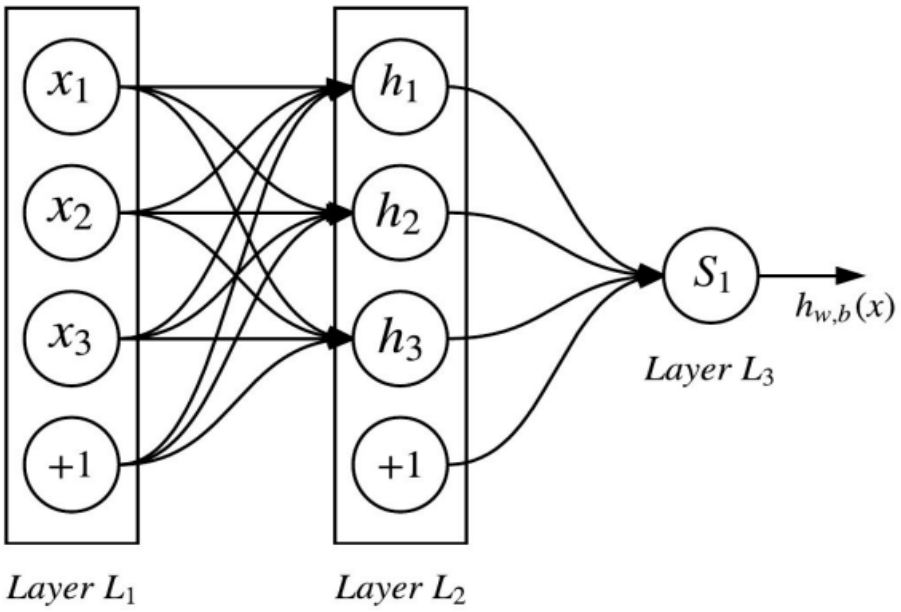


图 1 只含有一个隐藏层的 MLP

我们用一个 3 维向量 $X=[x_1,x_2,x_3]$ 表示图中输入层，用 3 维向量 $H[h_1,h_2,h_3]$ 表示图中隐藏层.则有 $H=f(W^TX)=f(\sum_{i=1}^{3}W_i^1x_i+b^1)$ 其中 W^1 是第一层的 3×3 的权重矩阵，f 是激励函数，b^1 是第一层的偏置项(bias)。这里激励函数 f 通常是非线性的，一般选用 sigmoid 函数、tanh 函数或者 ReLU 函数。代入之后，表达式分别如下所示：

$$f(W^T X) = sigmoid(W^T X) = \frac{1}{1+exp(-W^T X)}$$

$$f(W^T X) = tanh(W^T X) = \frac{e^{W^T X} - e^{-W^T X}}{e^{W^T X} + e^{-W^T X}}$$

$$f(W^T X) = ReLU(W^T X) = \max(0, W^T X)$$

而由隐藏层到输出层则可以看成是一个多类别的逻辑回归，一般选用 softmax 函数作为激励函数。则输出可以表示为 $softmax(\sum_{i=1}^{3} W_i^2 x_i + b^2)$，这里 W^2 是第二层的 3×3 的权重矩阵，f 是激励函数，b^2 是第二层的偏置项(bias)。

总结起来，这个三层 MLP 用公式总结起来就是

$$g(X) = G(b^2 + W^2(f(b^1 + W^1 X)))$$

其中函数 G 就是 softmax 函数，它将多个神经元的输出映射到（0，1）区间内，一般用来进行多分类。softmax 函数定义如下：

$$G(X)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \qquad for \quad j = 1, ..., k$$

这里 K 代表输入变量 X 的维度。

3.1.2 自动编码器（Autoencoder，AE）

自动编码器依靠编码和解码过程来重构输入数据，从而学习数据的隐层表示。传统的自动编码器可以视为一个三层的神经网络：包含有相同规模的输入层 x 和输出层 y，还有一个隐藏层 h，其结构如下图所示。

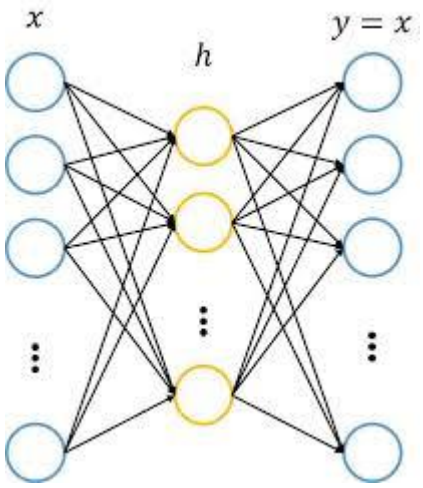


图 2 自动编码器结构示意图

自动编码器的目的是使得输入 x 和输出 y 尽可能接近，并用重构误差来表示这种接近程度。一般重构误差包括均方误差(mean-square error，MSE)和交叉熵(cross-entropy)。为了解决自动编码器容易学习到一个恒等函数的问题,研究者后来又提出了稀疏自动编码器和降噪自动编码器。2007 年，Bengio 等人[16]通过堆叠多个降噪自动编码器，提出了栈式降噪自动编码器（Stacked Denoising Autoencoder，SDAE）的概念。

如今，自动编码器，特别是栈式降噪自动编码器，在推荐系统中主要被用于学习用户和项目的隐层表示,通过重构学习用户和物品的相关信息(如评分数据和文本、图像信息)，从而获得用户或物品的隐层表示，最后基于这种隐层表示来预测用户对物品的偏好，并应用在评分预测、文本推荐和图像推荐等场景中[17]。

3.1.3 卷积神经网络（Convolutional Neural Network，CNN）

卷积神经网络是当今图像识别领域的研究热点。相比于传统的 MLP，卷积神经网络使用池化（pooling）可以减少模型中的神经元数量。特别是当输入层是多维图像的时候，CNN 可以将图像直接作为网络的输入，从而可以避免传统处理算法中复杂的特征提取和数据重建过程。卷积神经网络的基本结构主要分为输入层、卷积层、下采样层（池化层）、全连接层和输出层，如下图所示。

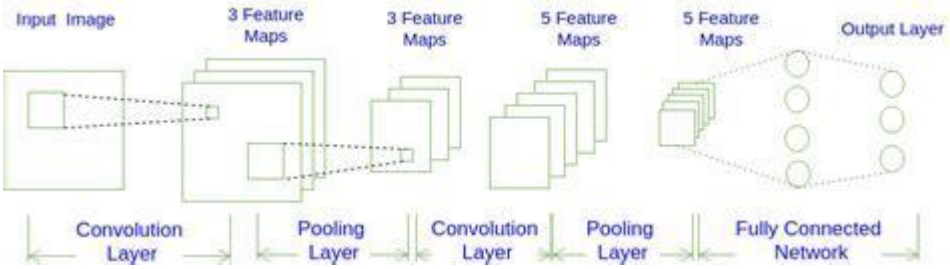


图 3 卷积神经网络结构示意图

在推荐系统领域，CNN 主要被用来从文本、图像、音频等内容中提取物品的隐藏特征，从而获取物品的低维向量表示，其在音乐推荐、图像推荐还有文本推荐等领域都有所应用[18]。

3.1.4 循环神经网络（Recurrent Neural Network，RNN）

传统 CNN 的层与层之间是全连接的,而每一层的节点之间则没有连接。与之不同的是，循环神经网络在网络各隐层的节点之间加入连接,从而能够通过获取输入层的输出和前一时刻的隐层状态来计算当前时刻隐层的输出，也就是说 RNN 能够对过去的信息进行记忆。下图是一个包含输入单元、输出单元和隐层单元的典型 RNN 结构。

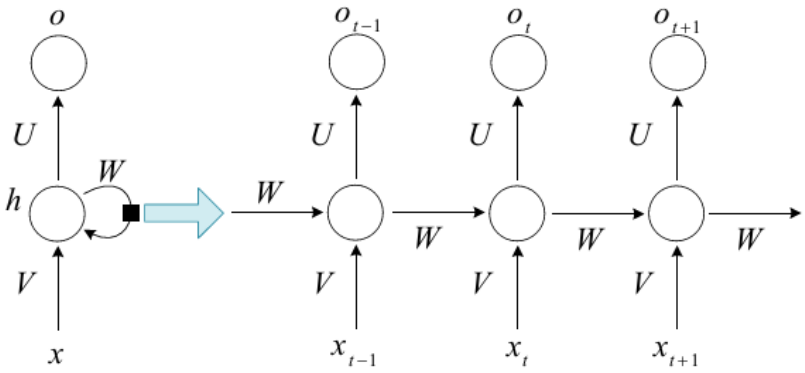


图 4 循环神经网络结构示意图

而为了解决传统 RNN 存在梯度消失和难以学习数据之间的长期以来关系的问题，Hochreiter 等人[19]提出了长短时间记忆网络（Long Shor-Term Memory，LSTM），Cho 等人[20]则提出了门限循环单元（Gated Recurrent Unit，GRU）。LSTM 和 GRU 通过增加保存长期状态的隐层单元，能够更加有效地建模长期以来关系，是目前应用最为广泛的循环神经网络模型。

在推荐系统领域，RNN 主要用来建模数据之间的序列影响，从而帮助获取更为有效的用户和物品隐层表示。一方面，RNN 可以被用来建模推荐系统中用户行为的序列模式[21][22]，另一方面，它还可以被应用在建模用户和物品相关的文本信息中词语之间的序列影响[23]。在文本推荐、图像推荐、评分预测以及基于未知社交网络中的兴趣点推荐等领域应用广泛。

3.1.5 受限制玻尔兹曼机（Restricted Boltzmann Machine，RBM）

玻尔兹曼机（Boltzmann machine，BM）由一些二元的可见单元（对应可见变量，即数据样本）和一些二元的隐层单元（对应隐层变量）构成。当处于状态 0，表示该神经元处于抑制状态，而状态 1 则对应表示该神经元处于激活状态。虽然 BM 具有强大的无监督学习能力，但其训练过程却非常耗时。为此，Sejnowski 等人提出了受限制玻尔兹曼机（Restricted Boltzmann Machine，RBM），通过在原有 BM 基础上除去同层变量之间的连接，可以显著提高学习效率。

如下图所示，RBM 包括可见层 v 以及隐层 h，两层之间是全连接的，而同层的节点之间则是互不连接的。

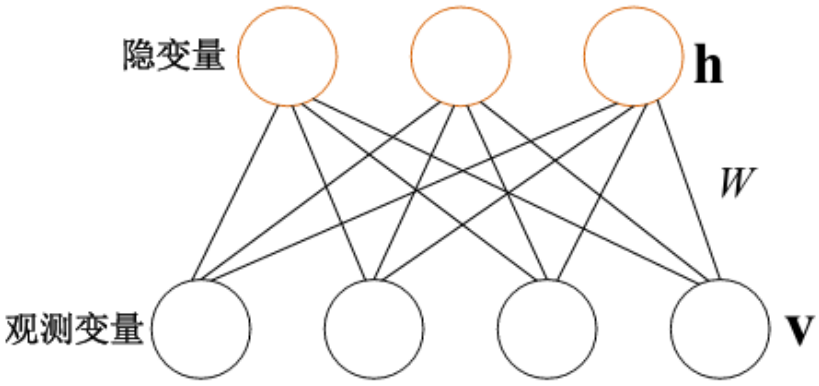


图 5 受限制玻尔兹曼机结构示意图

作为最早被应用在推荐系统中的神经网络模型[24][25]，RBM 当前主要用来重构用户的评分数据，从而学习到用户的隐层表示，进而实现对未知评分的预测。

3.2 深度学习在推荐领域的研究现状

深度学习当前在推荐系统中，主要用来学习用户和物品的输入数据从而获得隐层表示，然后再根据这种隐层表示为用户推荐物品。一个基于深度学习的推荐系统框架，如下图所示通常包含三层：输入层、模型层还有输出层。

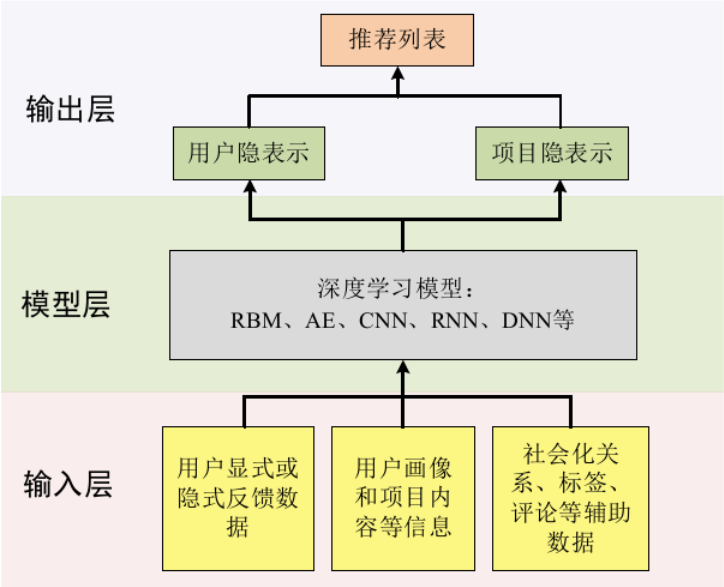


图 6 基于深度学习的推荐系统框架

输入层的数据一般包括：用户的显式反馈（评分、喜欢或不喜欢）或隐式反馈数据（浏览、点击等行为数据）、用户画像（性别、年龄、偏好等）和物品内容（文本、图像等信息）、用户生成内容（社会化关系、标签、评论等辅助数据）。在模型层，通常使用前面提到的深度学习技术（如受限制玻尔兹曼机、自动编码器、卷积神经网络、循环神经网络等）来处理输入数据并获得隐层表示。而在输出层，则将得到的隐层表示通过内积（inner product）、softmax、相似度计算等方法产生项目的推荐列表。

3.2.1 深度学习在基于内容的推荐系统中的应用

在基于内容的推荐系统中，深度学习可以用来从项目的内容信息中提取物品的隐层表示，以及从用户的画像信息以及历史行为数据中获取用户的隐层表示，之后再基于获得的隐层表示来计算书用户和物品的匹配度，进而为用户推荐物品。

1）基于 MLP 的内容推荐

Cheng 等人[6]通过学习用户特征、物品特征和情境特征等多源异构数据，提出了用于 APP 推荐的深广学习（Wide&Deep Learning）模型。如下图所示，该模型联合训练了一个宽广线性模型（图中左侧）和一个深度神经网络（图中右侧）来确保模型记忆能力和泛化能力的均衡。

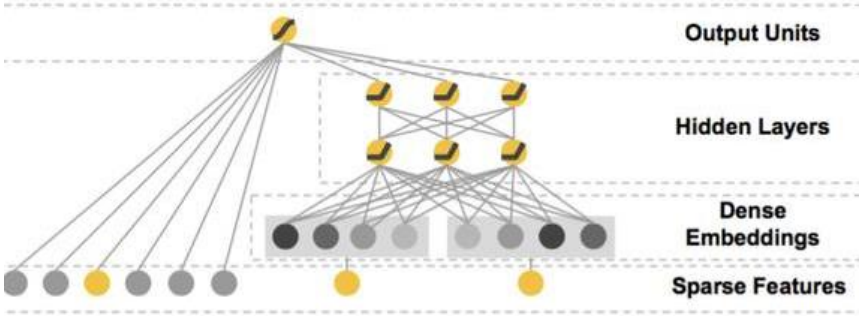


图 7 深广学习模型结构

与深广模型类似，He 等人[26]提出了一种神经协同过滤方法（Neural Collaborative Filtering，NCF），该方法通过将用户和物品的的特征作为输入，并利用多层神经网络来学习用户和物品之间的交互函数,得到一种矩阵因子分解的泛化结构和一种多层感知机结构。接下来利用神经矩阵因子分解模型（Neural Matrix Factorization Model，NeuMF）来组合矩阵因子分解的线性特征和深度神经网络的非线性特征。除此之外，Covington 等人[1]也提出了一种基于 MLP 的深度神经网络模型，并应用在了 YouTube 视频推荐上。

2）基于 CNN 的内容推荐

Gong 等人[27]提出了一种基于注意力的卷积神经网络（Attention-based Neural Network）来进行微博中的 Hashtag 推荐。这里，作者将 hashtag 推荐问题理解为一个多标记分类问题，并用 CNN 来获取微博的特征。之后，Zhang 等人[28]通过利用多模信息来进行微博的 hashtag 推荐。这里，作者采用 CNN 从图像中提取特征，而用 RNN 从文本中提取特征，之后再结合这两方面的特征来进行标签推荐。而注意力机制则用来建模图像和文本信息的局部关联性。除此之外，Wang 等人[29]则利用 CNN 来学习文章中的语义信息，提出了一种动态注意力深度模型（Dynamic Attention Deep Model，DADM），用来研究编辑者的文章推荐问题。Oord 等人[30]则利用 CNN 来学习音乐的音频信号数据和用户的历史收听数据，来解决音乐推荐系统中的冷启动问题。

3）基于 RNN 的内容推荐

与基于注意力的 CNN 模型类似，注意力机制也被用于基于 RNN 的内容推荐中。Li 等人[31]就提出了一种基于注意力的 LSTM 模型来进行微博中的 hashtag 推荐。将 RNN 与注意力机制结合起来，可以帮助提取文本中的序列特征，从而在微博中识别出最具有信息量的词。同时,RNN 还经常被用来做新闻推荐,Okura 等人[32]就首先使用降噪自动编码器(DAE)来从新闻中提取出文章的隐层表示,并采用 RNN 来从用户的历史行为列表中来学习用户的隐层表示，从而来获取用户偏好，之后将基于新闻和用户的隐层表示用点乘进行关联，最终为用户产生新闻推荐列表。

3.2.2 深度学习在协同过滤中的应用

将深度学习应用于协同过滤中，可有效改善传统矩阵因子分解可扩展性不足的问题[33]。其通常做法一般是，将用户的评分向量（或者是物品的被评分向量）作为输入，利用深度学习来学习用户或物品的隐层表示，之后再利用逐点损失（Point-wise Loss）和成对损失（Pair-wise Loss）等损失函数来对模型的参数进行优化[34]，最后再用学习到的隐层表示来进行物品推荐。

1）基于 AE 的协同过滤

自动编码器近些年来，被越来越多地应用到了传统的协同过滤方法中。如 Sedhain 等

人[35]提出的基于自动编码器的协同过滤模型(AutoRec),就利用自动编码器的编码过程和解码过程来产生输出,并通过最小化重构误差来优化模型的参数,进而对评分进行预测。而与 AutoRec 不同的是, Strub 等人[36]利用两个栈式降噪自动编码器 (SDAE),针对评分矩阵的稀疏性问题,在训练过程中直接将评分矩阵中的缺失值归零,从而减少了网络的连接数量。Wu 等人[34]的协同降噪自动编码器 (Collaborative Denoising Auto-Encoders, CDAE) 则是将自动编码器应用到了 top-N 推荐问题上。

2) 基于 RBM 的协同过滤

Salakhutdinov 等人[25]在 2007 年最早将受限制玻尔兹曼机应用到协同过滤推荐模型之中。针对传统 RBM 模型仅仅利用物品之间的关联关系,Georgiev 等人[37]在原有 RBM 模型基础上进行了扩展,并且简化了模型的训练和预测过程。后来,何洁月等人[38]又将好友信任关系加入到 RBM 之中,提出一种基于实值状态的玻尔兹曼机,有效缓解了原有模型的稀疏性问题。

3) 基于 RNN 的协同过滤

RNN 能够建模用户行为之间的相互依赖关系, 也可以用来建模用户的历史行为对当前时刻用户行为的影响。因此可以利用 RNN, 在传统协同过滤之中融入时间序列信息, 从而提升推荐系统的性能。为此, Song 等人[22]通过融入时间信息并在多种粒度上建模用户的兴趣偏好, 提出了一种多等级时间深度语义结构化模型 (Mutli-Rate TDSSM)。而考虑到推荐系统中的用户行为之间往往存在着多种类型,Liu 等人[39]提出了一种循环 Log 双线性模型 (Recurrent Log-BiLinear, RLBL), 该模型用 RNN 来建模用户行为之间的长程依赖关系, 而用 Log 双线性模型 (Log-BiLinear, LBL) [40]来建模短时的情境信息, 从而对用户在下一时刻的行为信息进行预测。

3.3 深度学习应用在推荐系统上的优势和存在的问题

基于深度学习的推荐系统能够有效地融合多源异构数据,可以缓解传统推荐系统中的数据稀疏和冷启动问题。而融合多种推荐方法的混合推荐,虽然也可以在一定成都上缓解数据稀疏问题,但是这种方法却面临着辅助数据难以表示的问题。传统的方法,如协同主题回归 (Collaborative Topic Regression, CTR) [41], 并不能获取辅助数据的有效表示[12]。而利用深度学习来自动提取特征,则可以从辅助数据中学习到有效的用户和物品隐层表示。

3.3.1 优势

总的来说, 相比于传统推荐方法, 基于深度学习的推荐方法一般有以下三点优势:

1) 传统浅层模型提取到的特征通常是稀疏和高维的, 而深度学习可以学习到非线性的多层次抽象特征表示, 获取稠密和低维的特征表示[6]。

2) 深度学习可以克服不同数据之间的异构性[13][1], 可以方便地通过各种粗糙的原始数据的输入来学习到用户和物品的隐层表示。

3) 深度学习可以帮助从图像、视频这样的非结构化中提取特征信息, 避免复杂的人工特征工程[13]。

3.3.2 存在的问题

同时，基于深度学习的推荐方法往往也存在着以下问题：

1）可解释性问题。基于深度学习的推荐模型往往类似于一个黑盒，难以为最终做出的推荐结果找到一个合理的解释。而如何在增强深度学习在推荐系统中的可解释性，仍需要进一步的深入研究。

2）可扩展性问题。深度学习在推荐系统中的另一个难题在于，其模型往往需要长时间的训练，而权衡模型的可扩展性与复杂度仍是当前的一大挑战。

4．总结与展望

本文在传统推荐方法的基础上，介绍了当前基于深度学习的推荐系统的研究现状，并对其中涉及到的深度学习技术做了具体的说明。通过上面的总结，可以看到，深度学习在推荐系统领域的应用目前仍然处于起步阶段，接下来必将会有更多、更广泛的尝试[42]。

而对于未来的发展趋势，则可以从以下三个方面进行展望：

1）将深度学习与现有推荐方法相结合

基于内容的方法和协同过滤方法等传统方法的研究已经日臻完善。而且普遍具有实现方便和可解释性强等优势。而深度学习则可以融合用户评论、标签信息以及社交关系等多源异构数据，学习到用户和物品更深层次的表示。而结合两者的优势，将深度学习与传统推荐方法相结合的研究，也将会是未来研究的发展趋势。

2）基于深度学习的跨域推荐

融合用户在不同平台的数据，从而进行跨域推荐，可以克服单一领域信息不足，并有利于缓解数据稀疏和冷启动问题。目前针对跨域推荐，当前主要的研究方法包括基于协同过滤的方法、基于迁移学习的方法和基于张量分解的方法等。但这些方法通常只能针对特定的信息进行融合，适应性和可扩展性有待提升。而结合深度学习的跨域推荐则方便融合不同类型的异构数据进行推荐，且目前已经在 Google[1]和微软[21]等公司得到了实际应用。基于深度学习的跨域推荐，也将是未来的研究重点。

3）可解释性方面的研究

基于深度学习的推荐方法在提供给用户推荐结果的同时，往往难以给出合理的推荐理由，这在一定程度上会减低用户对推荐结果的接受程度。因此在未来，有必要从模型、数据还有心理学、经济学等方面进行研究，来提升深度学习推荐结果的可解释性。

5．参考文献

[1] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, pages 191–198, 2016.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng., 17(6):734–749, 2005.

[3] Vipul Vekariya and G. R. Kulkarni. Hybrid recommender systems: Survey and experiments. In 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok,Thailand, May 16-18, 2012, pages 469–473, 2012.

[4] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006, pages 1097–1101, 2006.

[5] Saul Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011, pages 109–116, 2011.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. CoRR, abs/1606.07792, 2016.

[7] Yuxin Peng, Wenwu Zhu, Yao Zhao, Changsheng Xu, Qingming Huang, Hanqing Lu, Qing-Hua Zheng, Tiejun Huang, and Wen Gao. Cross-media analysis and reasoning: advances and directions. Frontiers of IT & EE, 18(1):44–57, 2017.

[8] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. ACM Trans. Management Inf. Syst,6(4):13:1–13:19, 2016.

[9] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, pages 293–296, 2010.

[10] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, pages 1933–1942, 2017.

[11] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. Context-aware recommender systems for learning: A survey and future challenges. TLT, 5(4):318–335, 2012.

[12] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, pages 1235–1244, 2015.

[13] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA,August 13-17, 2016, pages 353–362, 2016.

[14] Li Deng and Dong Yu. Deep learning: Methods and applications. Foundations and Trends in Signal Processing, 7(3-4):197–387, 2014.

[15] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis : A survey. CoRR, abs/1801.07883, 2018.

[16] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008, pages 1096–1103, 2008.

[17] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. Relational stacked denoising autoencoder for tag

recommendation. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., pages 3052–3058, 2015.

[18] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. Learning image and user features for recommendation in social networks. In 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 4274–4282, 2015.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[20] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. CoRR, abs/1409.1259, 2014.

[21] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017, pages 495–503, 2017.

[22] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. Multi-rate deep learning for temporal recommendation. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016, pages 909–912, 2016.

[23] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 415–423, 2016.

[24] Carlos Eduardo R. de Mello, Geraldo Zimbrão, and Jano Moreira de Souza. Extensão do WEKA para métodos de agrupamento com restrição de contigüidade do rio de janeiro. In IX Brazilian Symposium on Geoinformatics, 25-28 November, Campos do Jordão, São Paulo, Brazil, pages 277–282, 2007.

[25] Tran The Truyen, Dinh Q. Phung, and Svetha Venkatesh. Ordinal boltzmann machines for collaborative filtering. In UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009, pages 548–556, 2009.

[26] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. CoRR, abs/1708.05031, 2017.

[27] Yuyun Gong and Qi Zhang. Hashtag recommendation using attention-based convolutional neural network. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pages 2782–2788, 2016.

[28] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag recommendation for multimodal microblog using co-attention network. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pages 3420–3426, 2017.

[29] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, pages 2051–2059, 2017.

[30] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 2643–2651, 2013.

[31] Yang Li, Ting Liu, Jing Jiang, and Liang Zhang. Hashtag recommendation with topical attention-based

LSTM. In COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pages 3019–3029, 2016.

[32] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, pages 1933–1942, 2017.

[33] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey E. Hinton. Restricted boltzmann machines for collaborative filtering. In Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon,USA, June 20-24, 2007, pages 791–798, 2007.

[34] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016, pages 153–162, 2016.

[35] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy,May 18-22, 2015 - Companion Volume, pages 111–112, 2015.

[36] Florian Strub and Jérémie Mary. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In NIPS Workshop on Machine Learning for eCommerce, Montreal, Canada, December 2015.

[37] Kostadin Georgiev and Preslav Nakov. A non-iid framework for collaborative filtering with restricted boltzmann machines. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, pages 1148–1156, 2013.

[38] J.-Y He and B Ma. Based on real-valued conditional restricted boltzmann machine and social network for collaborative filtering. 39:183–195, 01 2016.

[39] Qiang Liu, Shu Wu, and Liang Wang. Multi-behavioral sequential prediction with recurrent log-bilinear model. IEEE Trans. Knowl. Data Eng., 29(6):1254–1267, 2017.

[40] Andriy Mnih and Geoffrey E. Hinton. Three new graphical models for statistical language modelling. In Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007, pages 641–648, 2007.

[41] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011, pages 448–456, 2011.

[42] Shuai Zhang, Lina Yao, and Aixin Sun. Deep learning based recommender system: A survey and new perspectives. CoRR, abs/1707.07435, 2017.

二、开题报告

1. 问题提出的背景

1.1 背景介绍

在互联网飞速发展的今天，如何在海量数据中找到对自己真正有用的信息，越来越成为人们不得不去思考的一个问题。而推荐系统则可以为用户过滤掉低相关的内容，根据用户的兴趣、偏好，将符合用户品味的信息筛选出来，并以个性化列表的方式推荐给用户。如今，推荐系统正在不知不觉间改变着我们的生活：从社交网络（Facebook、Twitter、腾讯等）到电子商务（如 Amazon、eBay、Netflix、淘宝等），从新闻推荐（如 Google News、Grouplens、今日头条等）到信息检索（如 iGoogle、MyYahoo、百度等）再到位置服务（如 Foursquare、Yelp、大众点评等），我们无时无刻不在享受着推荐系统所带给我们的便捷。

而同时，近些年来，深度学习在图像处理，自然语言处理和语音识别等领域都取得了突破性的发展，而这也为推荐系统的进一步发展带来了新的机遇和挑战。作为推荐系统领域的顶级会议，ACM 推荐系统年会（ACM RecSys）就在 2016 年专门召开了第一届基于深度学习的推荐系统研究专题研讨会（DLRS'16）。而研究基于深度学习的推荐系统的文章，近些年在数据挖掘和机器学习顶级会议（SIGKDD，NIPS，SIGIR，WWW，AAAI）上的发表量也是连年增加。

由此可见，利用深度学习来改进现有的推荐算法，不仅会是学术界接下来的一个研究热点，其对于传统算法的提升也会进一步方便人类生活的方方面面。

1.2 推荐算法的研究现状

传统的推荐算法主要基于两种方法或它们的组合：基于内容的方法和基于协同过滤的方法。

基于内容的推荐算法的关键在于内容的挖掘，最简单的，比如说我们从一篇新闻的正文和标题中分析出一个人名，而在评论中也分析出其他用户在讨论时也提到了这些人名，那么我们就可以进行推荐。基于内容的推荐天然优势在于推荐的可解释性强以及适合缺少用户行为的新物品的推荐。但这种方法依赖于有效的特征提取，例如，在视频推荐中[1]，视频档案资料的建立就需要耗费巨大的工作量，而这时基于内容的推荐就显得不合时宜。

协同过滤则利用相似用户之间具有相似兴趣偏好的规律，来发现用户对物品的潜在偏好。与基于内容的推荐相比，协同过滤并不使用用户和物品的内容资料，更具一般性，可以用到更多领域的物品推荐，但同时也面临着数据稀疏（用户已评分的物品占总物品数量的很少一部分）和新物品的冷启动问题（新的用户和新的物品之间往往没有评分数据）。

而自从在 Netflix 竞赛大放异彩之后，结合矩阵分解（Matrix Factorization，MF）的推荐模型引领起新一阶段的研究潮流。Salakhutdinov 等人[2]提出了概率矩阵模型（Probabilistic Matrix Factorization，PMF），从概率角度描述了 MF。Koren 等人[3]通过将基于邻域的方法结合起来，得到了具有更强预测能力的 SVD++模型。之后，他们又更进一步，提出了融合时间信息的 Time-SVD++模型[4]。

而近些年来，随着深度学习技术的不断成熟，越来越多研究者开始尝试利用深度学习

来改善传统的推荐算法。Cheng 等人通过学习用户特征、物品特征和情境特征等多源异构数据，并结合多层感知机（Multilayer Perceptron，MLP），提出了用于 APP 推荐的深广学习（Wide&Deep Learning）模型。He 等人[5]则提出了可以组合矩阵分解的线性特征和深度神经网络的非线性特征的神经矩阵因子分解模型（Neural Matrix Factorization，NeuMF）。Gong 等人[6]则利用卷积神经网络（Convolutional Neural Network，CNN）来进行微博中的 HashTag 推荐。

1.3 本研究的意义和目的

传统的基于内容的推荐依赖于特定数据特征的提取，其有效性和可扩展性十分有限；而协同过滤方法则受到数据稀疏和冷启动问题的限制。而与传统推荐方法提取到的稀疏特征相比，深度学习技术可以提取到相对稠密的多层次的特征表示[7]，另一方面也可以方便地通过各种粗糙的原始数据输入来学习到用户和物品的隐层表示。而矩阵分解的方法具有非常好的扩展性，也可以改善数据的稀疏性问题，并且通过将高维矩阵映射为两个低维矩阵的同时，也可以节省存储空间。

因此，基于上述背景，本次的推荐算法研究，旨在利用卷积神经网络来挖掘数据中的隐含关联信息，并结合矩阵分解，融合时间信息因素，提出一种合理的评分预测模型，在预测的准确度和 top-N 推荐等方面改善传统的推荐算法。

2. 论文的主要内容和任务

2.1 主要研究内容

通过前期的文献调研，我们发现，传统的基于内容的推荐和协同过滤推荐方法难以挖掘出数据在更深层次上的关联。而深度学习技术在自然语言处理方面已经有了长足的发展，在挖掘文本数据信息方面效果显著。而同时，我们需要认识到，人们的兴趣偏好并不是一成不变的，几十年前大受欢迎的影片未必就会被现代青年所喜爱，因此合理地将时间因素考虑进来，融合到推荐模型中，也是提高预测准确度以及 top-N 推荐的一个行之有效的方法。

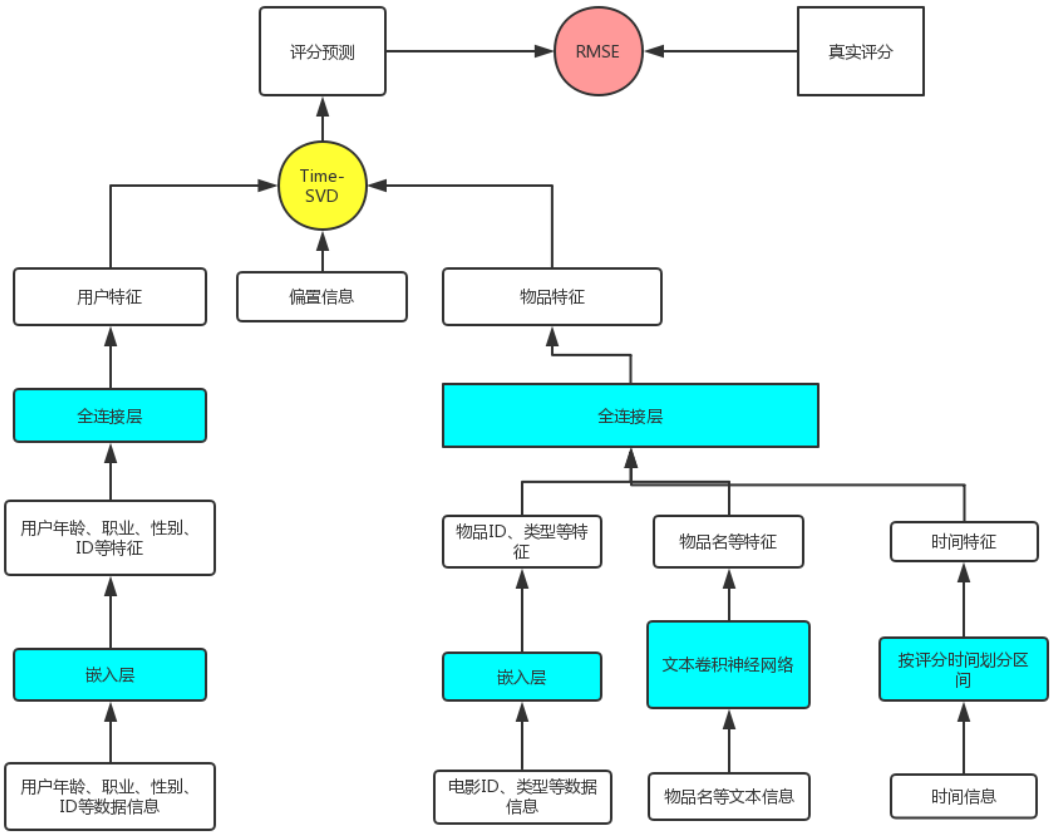


图 1 模型设计示意图

基于上面所述内容，我们的研究将会在真实的数据集上，首先利用卷积神经网络将用户和物品中稀疏、高维的数据信息转化为稠密低维的特征信息。接下来再结合 Time-SVD++ 的矩阵分解方法，将时间信息考虑进来，并与前面所得到的用户、物品向量一起建模，构建出一种如上图所示的，新型合理的评分预测模型。

2.2 研究任务

在这个背景下，我们的研究大致有如下几项任务：

1）获取并分析原始数据

本次实验我们主要使用 MoviLens 的 1M 数据集以及 Netflix 数据集。MovieLens 数据集是由 GroupLens 通过收集整理电影评分网站 MovieLens 的评分数据而发布的，1M 数据集包含 6040 个用户和 3952 部电影，共有 1000209 个用户评分记录。Netflix 数据集则是由在线影片租赁商 Netflix 举办推荐竞赛时提供的。

2）数据预处理

原始的数据包含很多的类别字段，以电影类别为例就包含犯罪片、动作片、科幻片等不同的类型。为了更方便地处理数据，我们需要使用 one-hot 编码用数字来代替原始的类别字段。同时我们还需要对原始数据进行去去噪处理来提取关键项的数据。

3）使用卷积神经网络提取数据特征信息

用户部分的处理相对简单，我们需要将用户信息输入嵌入层，从嵌入层索引出特征以

后，将特征传入全连接层，从而得到用户特征向量。而物品部分的处理则略有不同，这里我们需要额外使用文本卷积网络来单独处理物品名特征，之后再传入全连接层，与其他信息一起得到物品的特征向量。

4）矩阵分解建模

在建模部分我们借鉴了传统的 Time-SVD++的方法,将评论的时间信息划分为不同的区间，利用不同的时间区间分别学习出隐因子向量，从而在建模时加入时间因素。同时我们还需要在建模阶段加入偏置信息，从而使得评分预测结果更具有普遍性。

5）实验和分析

在实验阶段我们需要评估模型在评分预测和 top-N 推荐方面的表现，并与常见的规范化矩阵奇异值分解（Funk-SVD)、非负矩阵分解（NMF）还有 Time-SVD++方法进行比较。

评分预测中，我们使用均方根误差（root-mean-square error，RMSE）来评估预测的误差，其定义如下。

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

对于测试集中的一个用户 u 和物品 i ，另 r_{ui} 是用户 u 对物品 i 的实际评分，而 r_{ui} 是推荐算法给出的预测评分。

而在 top-N 推荐方面，我们则使用，命中率（Hit Ratio，HR）和归一化累积获得指标（Normalized Discounted Cumulative Gain，NDCG）来进行评估。

3. 技术路线

3.1 本卷积神经网络（Text-CNN）

卷积神经网络（CNN）在计算机视觉领域应用广泛，凭借其强大的局部特征捕捉能力，CNN 为分析和利用图像数据的研究者提供了极大的帮助。而 Yoon Kim 等人[8]则将 CNN 应用到了文本分类中，提出了 Text-CNN。在本研究中，Text-CNN 被用来学习物品的文本信息，从而获得隐藏层向量。TextCNN 通常由四部分构成：输入层、卷积层、池化层、全连接层，如下图所示：

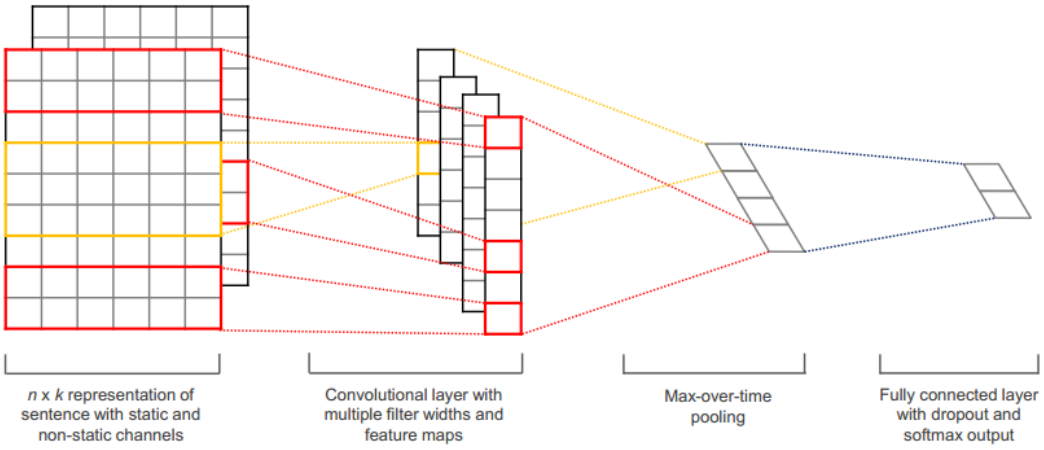


图 2 TextCNN 结构示意图

1）输入层

输入层是句子中词向量自上而下排列的矩阵，假如句子中有 n 个词，向量的维数是 k，那么输入层就是一个 $n \times k$ 的矩阵。这里，我们用 $x_i \in \mathbb{R}^k$ 来表示句子中第 i 个 k 维向量。那么一个长度为 n 的句子就可以表示为：

$$x_{1:n} = x_1 \oplus x_2 \oplus ... \oplus x_n$$

2）卷积层

在卷积层需要做的一件事就是卷积。卷积操作常用来做图像处理，就是取一定窗口大小的图像矩阵与滤波器（权重矩阵）做内积（逐个元素相乘再相加）。

如果用 $w \in \mathbb{R}^{hk}$ 来表示滤波器，那么一个特征 c_i 就是由一个窗口中的词语 $x_{i:i+h-1}$ 所生成：

$$c_i = f(w \cdot x_{i:i+h-1} + b)$$

这里 $b \in \mathbb{R}$ 是偏置项，而 f 则是像双曲函数这样的非线性函数。这样，我们就可以从句子里每 $h \times k$ 的窗口 $x_1 : h, x_2 : h, ..., x_{n-h+1:n}$ 通过卷积操作得到若干个特征图（Feature Map）

$$c = [c_1, c_2, ..., c_{n-h+1}]$$

其中 $c \in \mathbb{R}^{n-h+1}$

3）池化层

在池化层中，TextCNN 采用了一种称为 Max-over-time Pooling 的方法。该方法从之前得到的一维特征图中提取出最大值：$\hat{c} = \max\{c\}$。因为最大值往往代表着最重要的信号。而同时，因为不管特征图中有多少值，只需提取出其中的最大值，池化操作也可以解决不同长度的句子输入问题。

最终池化层输出的结果为各个特征图的最大值的集合，即一个一维向量。

4）全连接层

在 TextCNN 中，池化层中的一维向量的输出通过全连接的方式连接到 softmax 层中进

行分类。而在全连接层中则使用 Dropout 技术，并对全连接层上的权重参数给予 L2 正则化的限制，从而防止过拟合的发生。

举例来说，如果我们用 m 来表示滤波器的个数，那么池化层的输出结果就可以用 $z=[c_1,...,c_m]$ 来表示。在正则化阶段，dropout 会使用如下方式来优化权重参数 w

$$y=w\cdot(z\circ r)+b$$

这里$\circ$ 是向量按位乘法的操作，而 $r\in\mathbb{R}^m$ 则是由概率值 p 为 1 的伯努利随机变量所生成的屏蔽向量（masking vector）。

3.2 异值分解（SVD）

3.2.1 基本的奇异值分解

奇异值分解（SVD）是一种最基本的矩阵分解方式，它的核心思想是认为用户的兴趣只受少数几个因素 k 的影响，因此将稀疏且高维的 User-Item 评分矩阵分解为两个低维矩阵，如下所示。

$$U_{m\times k}V_{n\times k}^T\approx A_{m\times n}$$

这里 m 代表用户数量，n 代表物品数量，k 代表选取的特征数量。真实数据集中 m 和 n 的值很大，而 k 相比之下要小很多，至少两个数量级以上。而要计算物品 i 推荐给用户 u 的推荐分数，则需要计算内积，如下所示。

$$r_{ui}=q_i^T p_u$$

其中 p_u 和 q_i 分别为用户 u 和物品 i 的特征向量。

但是大多数真实数据集上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：

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

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。

3.2.2 带偏置的奇异值分解

基本的矩阵分解方法通过学习用户和物品的特征向量进行预测，其中用户的特征向量

代表了用户的兴趣，物品的特征向量代表了物品的特点。但是我们观测到的评分数据中，有很大一部分与用户对物品的喜好无关而只取决于用户或物品本身的特性。以电影推荐为例，标准宽松的用户往往会给出偏高的评分，而相对严格的用户的评分则普遍偏低；而同样，受大众欢迎的电影，往往会得到偏高的评分，而一些烂片的评分则普遍偏低。
我们把独立于用户或物品的因素称为偏置（Bias）部分，可以用如下公式来表示。

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

μ 是训练集中所有评分记录的全局平均数，它表示了训练数据的总体评分情况。b_u 是用户偏置，表示某一特定用户的打分习惯。b_i 是物品偏置，表示某一特定物品得到的打分情况。
接下来进行参数优化，损失函数如下所示：

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

其中参数 q_i、p_u、b_u、b_i 仍然可以采用交替最小二乘或随机梯度下降进行优化。
而加入以上偏置信息的评分预测公式如下所示。

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

3.2.3 加入时间动态的奇异值分解

在上面加入偏置信息的基础上，我们仍然需要考虑两个变化性的因素：一方面，每个物品的受欢迎程度 b_i 会随着时间变化而变化；另一方面，用户的评价标准 b_u 也会随时间而发生改变，我们使用如下公式来表示。

$$b_{ui}(t) = \mu + b_u(t) + b_i(t)$$

这里 $b_{ui}(t)$ 表明第 t 天用户 u 对物品 i 的基准评价，而 $b_u(t)$ 和 $b_i(t)$ 则是用户和物品偏置随着时间而改变的具体函数值。
这里我们需要对时间按区间划分为时间段，每个时间段内使用一个不同的值。
对于物品来说：

$$b_i(t) = b_i + b_{i,Bin(t)}$$

其中 $Bin(t)$ 为时间对应的函数。

而对于用户 u，我们用 t_u 表示时间的平均值，那么用户打分对于时间 t 的导数 $dev_u(t)$ 可以表示为：

$$dev_u(t) = sign(t - t_u) \cdot |t - t_u|^{\beta}$$

这里的$|t-t_u|$是用来衡量时间t和t_u之间的距离。

于是用户偏好关于时间的偏置信息可以表示为：

$$b_u(t)=b_u+\alpha_u\cdot dev_u(t)$$

新的损失函数为：

$$\min_{u,i,t}\sum_{(u,i)\in\kappa}(r_{u,i}(t)-\mu-b_u-\alpha_u dev_u(t)-b_{u,t}-b_i-b_{i,Bin(t)}{}^2+\lambda(b_u^2+\alpha_u^2+b_{u,t}^2+b_i^2+b_{i,Bin(t)}^2))$$

于是我们得到了将用在本次研究的评分预测公式：

$$r_{ui}=\mu+b_u(t)+b_i(t)+q_i^T p_u(t)$$

4. 可行性分析

1）数据可行性

互联网的飞速发展在改变人们生活方式的同时，也将用户的行为数据保存了下来。不管是用户显示的打分数据，还是点击、浏览商品时产生的隐式反馈数据，这其中都蕴含着丰富的潜在信息，来等待商家还有研究者进行挖掘。而这也为推荐系统领域的相关研究带来了广泛而多样的数据集。

以我们本次所将使用的 MovieLens 数据集为例，其中就包含用户数据 users.dat，电影数据 movies.dat 还有评分数据 ratings.dat。其中的用户数据包含用户 ID、性别、年龄、职业和邮编等，这些数据可以在研究中用来学习用户的特征向量，电影数据中的 ID、风格则可以用来学习物品的特征向量，其中的电影名作为文本信息则可以利用 TextCNN 来进行处理。而评分数据中的时间戳字段则可以用来发现用户的兴趣漂移规律，其中的评分数据则可以分成训练集和测试集，分别用来优化参数和评估模型的实际效果。

2）环境可行性

当前推荐系统的研究依然火热，从最早的 Netflix 电影推荐大赛，KDD Cup 数据挖掘竞赛，再到前段时间的阿里移动推荐大赛，京东的 JData 算法大赛。用于改进推荐算法的竞赛层出不穷,各家平台甚至拿出不菲的奖金来鼓励参赛者对自己现有的推荐算法进行改进。由此可见，在当前外部环境研究推荐算法的改进策略是顺势而为，大势所趋。

3）技术可行性

虽然在推荐系统领域，深度学习还处于一个新兴的阶段。但其在文本处理等方面的成功应用，也为其在挖掘用户和物品的关联信息等方面提供了借鉴。而谷歌推出的 TensorFlow，Facebook 推出的 pytorch，当下也已经成为应用非常广泛的深度学习框架，方便研究者进行深度学习方面的研究。而前面提到的 Time-SVD++和 TextCNN 技术都已经在相应数据集上跑出了不错的效果，其学习过程均可收敛从而达到最优解。由此可见，本次基于深度学习的矩阵分解算法研究具有较高的技术可行性。

4）经济可行性

我们本次使用的均是平台上公开的真实数据集，且仅用于研究，而非处于商业目的。且研究过程主要依赖软件来实现,同时实验所在的浙大电子服务研究中心也已经配备了装有 Tesla K40c GPU 的服务器，可以用于加速接下来深度学习实验的进行。

5. 预期目标及研究计划进度安排

5.1 预期目标

本次的推荐算法研究，旨在利用卷积神经网络来挖掘数据中的隐含关联信息，并结合矩阵分解，融合时间信息因素，提出一种合理的推荐算法模型，并希望能在 RMSE 等评分预测指标以及 HT、NDCG 等衡量 top-N 推荐质量的指标上，相比于传统的矩阵分解模型有一定程度的改进，缓解推荐时常遇到的数据稀疏和冷启动问题。

5.2 进度安排

表 1 研究进度安排

时间	进度安排
2017.7 - 2017.12	阅读相关文献资料，充分理解现有的研究成果，并撰写文献综述
2018.3.1 - 2018.3.20	提出初步的研究方案，与导师进行讨论，做出补充和改进
2018.3.21 - 2018.4.1	根据研究方案确定初步的模型，并撰写开题报告
2018.4.2 - 2018.4.10	获取实验数据，分析数据并进行预处理
2018.4.11 - 2018.4.25	实现论文中的关键算法，并在数据集上测试效果
2018.4.26 - 2018.5.5	对比不同算法，并对各项指标进行分析
2018.5.6 - 2018.5.15	对研究结果进行归纳，整理实验数据，完成论文初稿
2018.5.15 - 2018.5.30	完善和修改，并确定论文终稿

6. 参考文献

[1] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. pages 285–295, 2001.

[2] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007, pages 1257–1264, 2007.

[3] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008, pages 426–434, 2008.

[4] Yehuda Koren. Collaborative filtering with temporal dynamics. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009, pages 447–456, 2009.

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth,

Australia, April 3-7, 2017, pages 173–182, 2017.

[6] Yuyun Gong and Qi Zhang. Hashtag recommendation using attention-based convolutional neural network. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pages 2782–2788, 2016.

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. CoRR, abs/1606.07792, 2016.

[8] Yoon Kim. Convolutional neural networks for sentence classification. CoRR, abs/1408.5882, 2014.

三、外文翻译

文献原文：

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. CoRR, abs/1708.05031, 2017.

神经协同过滤

摘要

近年来，深层神经网络在语音识别，计算机视觉和自然语言处理方面都取得了巨大的成功。然而相对的，基于深层神经网络的推荐系统的探索却受到较少的关注。在这项工作中，我们力求开发一种基于神经网络的技术，来解决在含有隐形反馈的基础上进行推荐的关键问题————协同过滤。

尽管最近的一些工作已经把深度学习运用到了推荐中，但是他们主要是用它（深度学习）来对一些辅助信息（auxiliary information）建模，比如描述文字的项目和音乐的声学特征。当涉及到建模协同过滤的关键因素（key factor）————用户和项目（item）特征之间的交互的时候，他们仍然采用矩阵分解的方式，并将内积（inner product）做为用户和项目的潜在特征点乘。通过用神经结构代替内积这可以从数据中学习任意函数，据此我们提出一种通用框架，我们称它为 NCF（Neural network-based Collaborative Filtering，基于神经网络的协同过滤）。NCF 是一种通用的框架，它可以表达和推广矩阵分解。为了提升 NFC 的非线性建模能力，我们提出了使用多层感知机去学习用户-项目之间交互函数（interaction function）。在两个真实世界（real-world）数据集的广泛实验显示了我们提出的 NCF 框架对最先进的方法的显著改进。

关键字

协同过滤,神经网络,深度学习,矩阵分解,隐性反馈（Implicit Feedback）.

1. 引言

在信息爆炸的时代，推荐系统在减轻信息过载方面发挥了巨大的作用，被众多在线服务，包括电子商务，网络新闻和社交媒体等广泛采用。个性化推荐系统的关键在于根据过去用户交互的内容（e.g. 评分，点击），对用户对项目的偏好进行建模，就是所谓的协同过滤[31,46]。在众多的协同过滤技术中，矩阵分解（MF）[14,21]是最受欢迎的，它将用户和项目映射到共享潜在空间（shared latent space），使用潜在特征向量（latent features），用以表示用户或项目。这样一来，用户在项目上的交互就被建模为它们潜在向量之间的内积。

由于 Netflix Prize 的普及，MF 已经成为了潜在因素（latent factor）建模推荐的默认方法。已经有大量的工作致力于增强 MF（的性能），例如将其与基于邻居（相邻用户 or 项目）的模型集成[21]，与项目内容的主题模型[38]相结合，还有将其扩展到因式分解机（factorization machines）[26]，以实现特征的通用建模。尽管 MF 对协同过滤

有效，但众所周知，其性能会被简单的交互函数（内积）所阻碍。例如，对于显式反馈的评估预测任务，可以通过将用户和项目偏置项加入到交互函数中来改善 MF 模型的性能。虽然内积算子[14]似乎只是一个微不足道的调整,但它指出了设计一个更好的专用交互函数的方向。简单地将潜在特征的乘积进行线性组合的内积，可能不足以捕捉用户交互数据的复杂结构。

本文探讨了使用深层神经网络来学习数据的交互函数，而不是那些以前已经完成的手动工作 [18,21]。神经网络已经被证明具有拟合任何连续函数的能力[17]，最近深层神经网络（DNN）被发掘出在几个领域的出色表现：从计算机视觉，语音识别到文本处理 [5,10,15,47]。然而，与广泛的 MF 方法文献相比，使用 DNNs 进行推荐的工作相对较少。虽然最近的一些研究进展[37,38,45]已经将 DNN 应用于推荐任务，并展示出不错的效果，但他们大多使用 DNN 来建模一些辅助信息，例如物品的文字描述，音乐的音频特征和图像的视觉内容（描述）。对于影响建模效果的关键因素——协同过滤，他们仍然采用 MF，使用内积来组合用户和项目的潜在特征。

我们这项工作是通过形式化用于协同过滤的神经网络建模方法来解决上述研究问题。我们专注于隐性反馈，通过参考观看视频，购买产品和点击项目等间接反映用户偏好的行为。与显性反馈（例如评级和评论）相比，隐性反馈可以自动跟踪，从而更容易为内容提供商所收集。但是，由于隐性反馈不能反映用户的满意度，并且负反馈存在自然稀疏（natural scarcity）的问题，使得这个问题更具挑战性。在本文中，我们探讨了如何利用 DNN 来模拟噪声隐性反馈信号的中心问题。

这项工作的主要贡献有如下几点：

1、我们提出了一种神经网络结构来模拟用户和项目的潜在特征，并设计了基于神经网络的协同过滤的通用框架 NCF。

2、我们表明 MF 可以被解释为 NCF 的特例，并利用多层感知器来赋予 NCF 高水平的非线性建模能力。

3、我们对两个现实世界的数据集进行广泛的实验，以证明我们的 NCF 方法的有效性和对使用深度学习进行协作过滤的设想。

2. 准备工作

我们首先把问题形式化，并讨论现有的隐性反馈协同过滤解决方案。然后，我们简要概括广泛使用的 MF 模型，重点指出使用内积所造成的对模型的限制。

2.1 学习隐性数据

令 M 和 N 分别表示用户和项目的数量。我们将从用户的隐性反馈得到的用户-项目交互矩阵 $Y \in \mathbb{R}^{M \times N}$ 定义为：

$$y_{ui} = \begin{cases} 1, if\ interaction(user\ u, item\ i)\ is\ observed; \\ 0, otherwise. \end{cases} \quad (1)$$

这里 y_{ui} 为 1 表示用户 u 和项目 i 存在交互记录;然而这并不意味着 u 真的喜欢 i。同样的，值 0 也不是表明 u 不喜欢 i，也有可能是这个用户根本不知道有这个项目。这对隐性反馈的学习提出了挑战，因为它提供了关于用户偏好的噪声信号。虽然观察到的条目反映了用户对项目的兴趣，但是未查看的条目可能只是丢失数据，并且这其中存在自然稀疏的负反馈。

在隐性反馈上的推荐问题可以表达为估算矩阵 Y 中未观察到的条目的分数问题（这个分数被用来评估项目的排名）。基于模型的方法假定这些缺失的数据可以由底层模型生成（或者说描述）。形式上它可以被抽象为学习函数 $\hat{y}_{ui} = f(u,i|\Theta)$,其中 $\hat{y}_{ui}$ 表示交互 y_{ui} 的预测分数，Θ 表示模型参数，f 表示表示模型参数映射到预测分数的函数（我们把它称作交互函数）。

为了估计参数 Θ，现有的方法一般是遵循机器学习范例，优化目标函数。在文献中最常用到两种损失函数:逐点损失[14,19](pointwise loss)和成对损失[27,33] (pairwise loss) 。由于显性反馈研究上丰富的工作的自然延伸[21，46]，在逐点的学习方法上通常是遵循的回归模式，最小化 $\hat{y}^{ui}$.及其目标值 y_{ui} 之间的均方误差。同时为了处理没有观察到的数据，他们要么将所有未观察到的条目视作负反馈，要么从没有观察到条目中抽样作为负反馈实例[14]。对于成对学习[27，44]，做法是，观察到的条目应该比未观察到的那些条目的排名更高。因此，成对学习最大化观察项 $\hat{y}_{ui}$ 和未观察到的条目 $\hat{y}_{ui}$ 之间的空白，而不是减少 $\hat{y}_{ui}$ 和 y_{ui} 之间的损失误差。

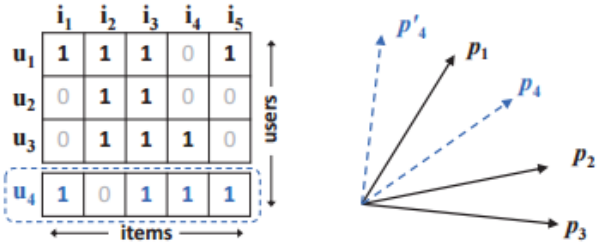
更进一步，我们的 NCF 框架利用神经网络,参数化相互作用函数 f ,从而估计 $\hat{y}_{ui}$ 。因此，它就很自然地同时支持逐点损失和成对损失。

2.2 矩阵分解

MF(Matrix Factorization)用一个潜在特征向量实值将每个用户和项目关联起来。令 p_u 和 q_i 分别表示用户 u 和项目 i 的潜在向量；MF 评估相互作用 y_{ui} 作为 p_u 和 q_i 的内积：

$$\hat{y}_{u,i} = f(u,i|\mathrm{p_u},\mathrm{q_i}) = \mathrm{p_u^T q_i} = \sum_{k=1}^{K} p_{uk} q_{ik}, \quad (2)$$

这里的 K 表示潜在空间（latent space）的维度。正如我们所看到的，MF 模型是用户和项目的潜在因素的双向互动,它假设潜在空间的每一维都是相互独立的并且用相同的权重将它们线性结合。因此，MF 可视为潜在因素（latent factor）的线性模型。



(a) user–item matrix (b) user latent space

Figure 1: An example illustrates MF's limitation. From data matrix (a), u_4 is most similar to u_1, followed by u_3, and lastly u_2. However in the latent space (b), placing p_4 closest to p_1 makes p_4 closer to p_2 than p_3, incurring a large ranking loss.

图 1 展示了的内积函数（inner product function）如何限制 MF 的表现力。这里有两个背景必须事先说明清楚以便于更好地了解例子。第一点，由于 MF 将用户和项目映射到同一潜在空间中，两个用户之间的相似性也可以用内积，或者潜在向量之间的角度的余弦值来衡量。第二点，不失一般性，我们使用 Jaccard 系数作为 MF 需要恢复的两个用户的真实状况之间的相似度。

我们首先关注的图 1(a)中的前三行（用户）。很容易可以计算出 $s_{23}(0.66) > s_{12}(0.5) > s_{13}(0.4)$。这样，$p_1$，$p_2$，$p_3$在潜在空间中的几何关系可绘制成图 1(b)。现在，让我们考虑一个新的用户 u_4，它的输入在图 1(a)中的用虚线框出。我们同样可以计算出 $s_{41}(0.6) > s_{43}(0.4) > s_{42}(0.2)$，表示$u_4$ 最接近 u_1，接着是 u_3，最后是u_2 。然而，如果 MF 模型将 p_4 放在了 最接近 p_1 的位置（图 1(b) 中的虚线展示了两种不同的摆放 p_4 的方式，结果一样），那么会使得 p_4 相比与 p_3更接近于 p_2 （显然，根据图 1(a)，u_4 应该更接近u_3），这会导致很大的排名误差（ranking loss）。

上面的示例显示了 MF 因为使用一个简单的和固定的内积，来估计在低维潜在空间中用户-项目的复杂交互，从而所可能造成的限制。我们注意到，解决该问题的方法之一是使用大量的潜在因子 K（**就是潜在空间向量的维度**）。然而这可能对模型的泛化能力产生不利的影响（比如 数据的过拟合问题），特别是在稀疏的集合上。在本文的工作中，我们通过使用 DNNs 从数据中学习交互函数，突破了这个限制。

3. 神经协同过滤

我们首先提出的总体框架 NCF，阐述 NCF 如何学习强调了隐式数据的二进制属性的概率模型。然后，我们展示了，MF 能够表达为 NCF 的推广（MF 矩阵分解模型是 NCF 的一个特例）。我们探索 DNNs 在协同过滤上的应用，提出了 NCF 的一个实例，它采用了多层感知器（MLP）来学习用户-项目交互函数。最后，我们在 NCF 框架下结合了 MF 和 MLP，提

出了一种新的神经矩阵分解模型（neural matrix factorization model）；它统一了在建模用户项目潜在结构方面，MF 的线性建模优势和 MLP 的非线性优势。

3.1 通用框架

为了允许神经网络对协同过滤进行一个完整的处理，我们采用图 2 展示的多层感知机去模拟一个用户项目交互 y_{ui} ，它的一层的输出作为下一层的输入。底部输入层包括两个特征向量 v_u^U 和 v_i^I ，分别用来描述用户 u 和项目 i 。他们可以进行定制，用以支持广泛的用户和项目的建模，例如上下文感知[28,1]，基于内容[3]，和基于邻居的构建方式[26]。由于本文工作的重点是纯的协同过滤模型设置，我们仅使用一个用户和一个项目作为输入特征，它使用 one-hot 编码将它们转化为二值化稀疏向量。注意到，我们对输入使用这样的通用特征表示，可以很容易地使用的内容特征来表示用户和项目，以调整解决冷启动问题。

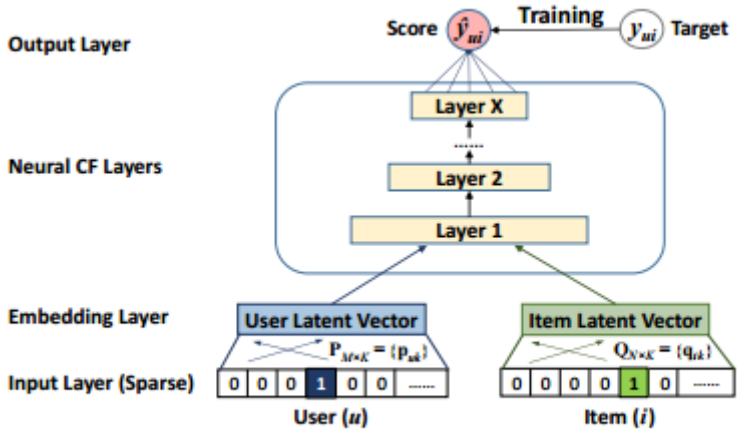


Figure 2: Neural collaborative filtering framework

输入层上面是嵌入层（Embedding Layer）；它是一个全连接层，用来将输入层的稀疏表示映射为一个稠密向量（dense vector）。所获得的用户（项目）的嵌入（就是一个稠密向量）可以被看作是在潜在因素模型的上下文中用于描述用户（项目）的潜在向量。然后我们将用户嵌入和项目嵌入并送入到多层神经网络结构中，我们把这个结构称为神经协作过滤层，它将潜在向量映射为预测分数。NCF 层的每一层可以被定制，用以发现用户-项目交互的某些潜在结构。最后一个隐含层 X 的维度尺寸决定了模型的能力。最终输出层是预测分数 $\hat{y}_{ui}$ ，训练通过最小化 $\hat{y}_{ui}$ 和其目标值 y_{ui} 之间逐点损失进行。我们注意到，另一种方式来训练模型是通过成对学习，如使用个性化贝叶斯排名[27]和基于余量的损失（margin-based）[33]。由于本文的重点是对神经网络建模部分，我们将使用成对学习训练 NCF 留给今后的工作。

当前我们制定的 NCF 预测模型如下：

$$\hat{y}_{ui} = f(\mathbf{P}^T \mathbf{v}_u^U, \mathbf{Q}^T \mathbf{v}_i^I \mid \mathbf{P}, \mathbf{Q}, \Theta_f), \quad (3)$$

其中 $P \in \mathbb{R}^{M \times K}$，$Q \in \mathbb{R}^{N \times K}$，分别表示用户和项目的潜在因素矩阵；$\Theta_f$ 表示交互函数 f 的模型参数。由于函数 f 被定义为多层神经网络，它可以被定制为：

$$f(\mathbf{P}^T \mathbf{v}_u^U, \mathbf{Q}^T \mathbf{v}_i^I) = \phi_{out}(\phi_X(...\phi_2(\phi_1(\mathbf{P}^T \mathbf{v}_u^U, \mathbf{Q}^T \mathbf{v}_i^I))...)), \quad (4)$$

其中 ϕ_{out} 和 ϕ_x 分别表示为输出层和第 x 个神经协作过滤（CF）层映射函数，总共有 X 个神经协作过滤（CF）层。

3.1.1 NCF 学习

学习模型参数，现有的逐点学习方法[14,39]主要运用均方误差（squared loss）进行回归：

$$L_{sqr} = \sum_{(u,i) \in \mathbf{y} \cup \mathbf{y}^-} w_{ui} (y_{ui} - \hat{y}_{ui})^2, \quad (5)$$

其中 y 表示交互矩阵 Y 中观察到的条目（如对电影有明确的评分，评级），y^- 表示消极实例（negative instances，可以将未观察的样本全体视为消极实例，或者采取抽样的方式标记为消极实例）；w_{ui} 是一个超参数，用来表示训练实例 (u,i) 的权重。虽然均方误差可以通过假设观测服从高斯分布[29]来作出解释，我们仍然指出，它不适合处理隐性数据（implicit data）。这是因为对于隐含数据来说，目标值 y_{ui}. 是二进制值 1 或 0，表示 u 是否与 i 进行了互动。在下文中，我们提出了逐点学习 NCF 的概率学方法，特别注重隐性数据的二进制属性。

考虑到隐性反馈的一类性质，我们可以将 y_{ui} 的值作为一个标签————1 表示项目 i 和用户 u 相关，否则为 0。这样一来预测分数 $\hat{y}_{ui}$ 就代表了项目 i 和用户 u 相关的可能性大小。为了赋予 NCF 这样的概率解释，我们需要将网络输出限制到[0，1]的范围内，通过使用概率函数（e.g. 逻辑函数 sigmoid 或者 probit 函数）作为激活函数作用在输出层 ϕ_{out}，我们可以很容易地实现数据压缩。经过以上设置后，我们这样定义似然函数：

$$p(\mathbf{y}, \mathbf{y}^- | \mathbf{P}, \mathbf{Q}, \mathbf{\Theta_f}) = \prod_{(u,i) \in \mathbf{y}} \hat{y}_{ui} \prod_{(u,i) \in \mathbf{y}^-} (1 - \hat{y}_{ui}). \quad (6)$$

对似然函数取负对数，我们得到（负对数可以用来表示 Loss 函数，而且还能消除小数乘法的下溢出问题）：

$$L = -\sum_{(u,i) \in \mathbf{y}} \log \hat{y}_{ui} - \sum_{(u,i) \in \mathbf{y}^-} \log(1 - \hat{y}_{ui}) = \sum_{(u,i) \in \mathbf{y} \cup \mathbf{y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}). \quad (7)$$

这是 NCF 方法需要去最小化的目标函数，并且可以通过使用随机梯度下降（SGD）来进行训练优化。细心的读者可能发现了，这个函数和二类交叉熵损失函数（binary cross-entropy loss，又被成为 log loss）是一样的。通过在 NCF 上使用这样一个概率

处理（probabilistic treatment），我们把隐性反馈的推荐问题当做一个二分类问题来解决。由于分类用的交叉熵损失很少出现在有关推荐的文献中，我们将在这项工作中对它进行探讨，并在 4.3 节展示它的有效性。对于消极实例 y^- ，我们在每次迭代均匀地从未观察到的相互作用中采样(作为消极实例)并且对照可观察到交互的数量，控制采样比率。虽然非均匀采样策略（例如，基于项目流行度进行采样[14,12]）可能会进一步提高模型性能，我们将这方面的探索作为今后的工作。

3.2 广义矩阵分解

我们现在来证明 MF 是如何被解释为我们的 NCF 框架的一个特例。由于 MF 是推荐领域最流行的模型，并已在众多文献中被广泛的研究，复现它能证明 NCF 可以模拟大部分的分解模型[26]。由于输入层是用户（项目）ID 中的一个 one-hot 编码，所获得的嵌入向量可以被看作是用户（项目）的潜在向量。我们用 $\mathbf{P}^T\mathbf{v}_u^U$ 表示用户的潜在向量 p_u ， $\mathbf{Q}^T\mathbf{v}_i^I$ 表示项目的潜在向量 q_i ，我们定义第一层神经 CF 层的映射函数为：

$$\phi_1(\mathbf{p}_u,\mathbf{q}_i)=\mathbf{p}_u\odot\mathbf{q}_i, \quad (8)$$

其中 $\odot$ 表示向量的逐元素乘积。然后，我们将向量映射到输出层：

$$\hat{y}_{ui}=a_{out}\left(\mathbf{h}^T\left(\mathbf{p}_u\odot\mathbf{q}_i\right)\right), \quad (9)$$

其中 a_{out} 和 h 分别表示输出层的激活函数和连接权。直观地讲，如果我们将 a_{out} 看做一个恒等函数，h 权重全为 1，显然这就是我们的 MF 模型。在 NCF 的框架下，MF 可以很容易地被泛化和推广。例如，如果我们允许从没有一致性约束的数据中学习 h，则会形成 MF 的变体。如果我们用一个非线性函数 a_{out} ，将进一步推广 MF 到非线性集合，使得模型比线性 MF 模型更具有表现力。在本文的工作中，我们在 NCF 下实现一个更一般化的 MF，它使用 Sigmoid 函数 $1/(1+e^{-x})$ 作为激活函数，通过 log loss(第 3.1.1 节)学习 h 。我们称她为 GMF（Generalized Matrix Factorization，广义矩阵分解）。

3.3 多层感知机（MLP）

由于 NCF 用两条路线来对用户和项目建模（图 2 中可以明显看出用户和项目两个输入），自然地，需要通过两个路线，把他们各自的特征连接结合起来。这种设计已经在多模态深度学习工作中[47，34]被广泛采用。然而，简单地对向量的连接不足以说明用户和项目之间的潜在特征，这对协同过滤建模来说是不够的。为了解决这个问题，我们提出在向量连接上增加隐藏层，使用标准的 MLP（多层感知机）学习用户和项目潜在特征之间的相互作用。在这个意义上，我们可以赋予模型高水平的灵活性和非线性建模能力，而不是 GMF（广义矩阵分解）那样的简单使用逐元素相乘的内积来描述用户和项目之间的潜在交互特征。更确切地说，我们的 NCF 框架下的 MLP 模型定义为：

$$\mathbf{z}_1 = \phi_1(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}$$

$$\phi_2(\mathbf{z}_1) = a_2\left(\mathbf{W}_2^T \mathbf{z}_1 + \mathbf{b}_2\right),$$

$$\cdots$$

$$\phi_L(\mathbf{z}_{L-1}) = a_L\left(\mathbf{W}_L^T \mathbf{z}_{L-1} + \mathbf{b}_L\right),$$

$$\hat{y}_{ui} = \sigma\left(\mathbf{h}^T \phi_L\left(\mathbf{z}_{L-1}\right)\right), \qquad (10)$$

这里的 W_x，b_x 和 a_x 分别表示 x 层的感知机中的的权重矩阵，偏置向量（神经网络的神经元阈值）和激活函数。对于 MLP 层的激活函数，可以选择 sigmoid，双曲正切（tanh）和 ReLU，等等。我们分析一下每个函数：

1)sigmoid 函数将每个神经元的输出限制在（0,1），这有可能限制该模型的性能;并且它存在过饱和的问题，当输出接近 1 或者 0 的时候，神经元就会陷入停止学习的困境。

2)虽然双曲正切是一个更好的选择，并已被广泛使用[6，44]，但它只是在一定程度上缓和了 sigmoid 的问题，因为它可以被看作是 sigmoid 的缩放的版本。

3)因此，我们选择 ReLU，它更具生物合理性，并且已经被证明不会导致过饱和[9];此外，它支持稀疏的激活，非常适合稀疏的数据，使模型不至于过拟合。我们的实验结果表明，ReLU 的表现略好于双曲正切函数 tanh 和 sigmoid。

至于网络结构的设计，一种常见的解决方案是设计一个塔式模型，其中，底层是最宽的，并且每个相继的层具有更少的神经元数量（如图 2）。（设计这种结构的）前提是，通过在更高层使用少量的隐藏单元，它们可以从数据中学习到更多的抽象特征[10]。根据经验，我们搭建这样的塔结构：对于更高的层，相比于之前一层，缩减一半规模。

3.4 结合 GMF 和 MLP

到目前为止，我们已经开发了 NCF 的两个实例：GMF，它应用了一个线性内核来模拟潜在的特征交互；MLP，使用非线性内核从数据中学习交互函数。接下来的问题是：我们如何能够在 NCF 框架下融合 GMF 和 MLP，使他们能够相互强化，以更好地对复杂的用户-项目进行交互建模。

一个直接的解决方法是让 GMF 和 MLP 共享相同的嵌入层（Embedding Layer），然后再结合它们分别对相互作用的函数输出。这种方式和著名的神经网络张量（NTN,Neural Tensor Network）[33]有点相似。具体地说，对于结合 GMF 和单层 MLP 的模型可以公式化为：

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T a(\mathbf{p}_u \odot \mathbf{q}_i) + \mathbf{W}\begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix} + \mathbf{b}). \quad (11)$$

然而，共享 GMF 和 MLP 的嵌入层可能会限制融合模型的性能。例如，它意味着，GMF 和 MLP 必须使用的大小相同的嵌入层;对于数据集，两个模型的最佳嵌入尺寸差异很大，使得这种解决方案可能无法获得最佳的组合。

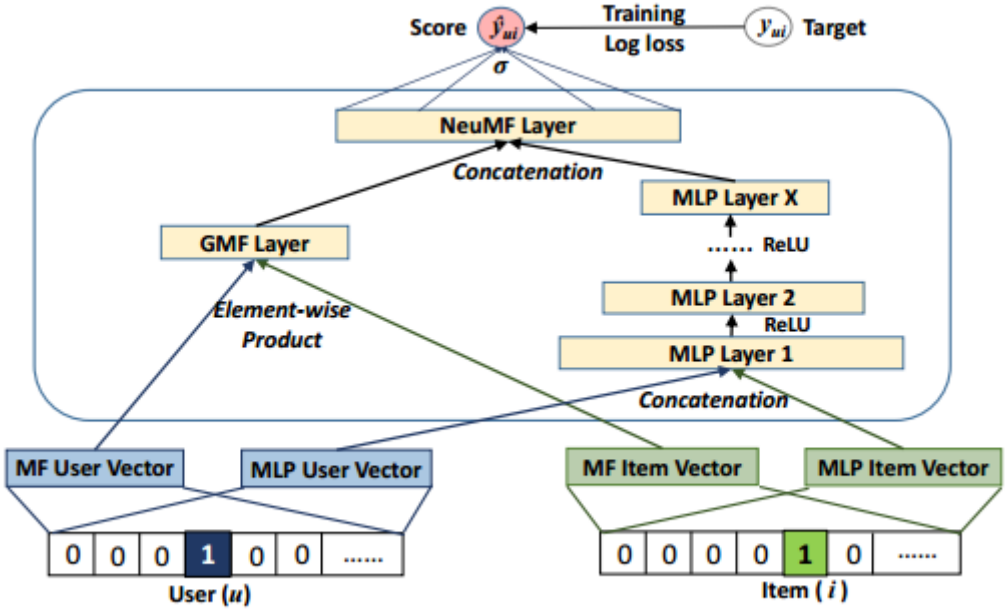


Figure 3: Neural matrix factorization model

为了使得融合模型具有更大的灵活性，我们允许 GMF 和 MLP 学习独立的嵌入，并结合两种模型通过连接他们最后的隐层输出。图 3 展示了我们的方案，公式如下：

$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G,$$

$$\phi^{MLP} = a_L(W_L^T(a_{L-1}(...a_2(W_2^T\begin{bmatrix}\mathbf{p}_u^M \\ \mathbf{q}_i^M\end{bmatrix}+\mathbf{b}_2)...))+\mathbf{b}_L),$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T\begin{bmatrix}\phi^{GMF} \\ \phi^{MLP}\end{bmatrix}), \quad (12)$$

这里的 p_u^G 和 p_u^M 分别表示 GMF 部分和 MLP 部分的用户嵌入；同样的，q_i^G 和 q_i^M 分别表示项目的嵌入。如之前所讨论的，我们使用 ReLU 作为 MLP 层的激活功能。该模型结合 MF 的线性度和 DNNs 的非线性度，用以建模用户-项目之间的潜在结构。我们将这一模式称为“NeuMF”，简称神经矩阵分解（Neural Matrix Factorization）。该模型的每个模型参数都能使用标准反向传播计算，由于空间限制这里就不再展开。

3.4.1 预训练

由于 NeuMF 的目标函数的非凸性，使得基于梯度的优化方法只能找到局部最优解（这也是训练一般神经网络所面临的问题）。研究表明，初始化（initialization）在深度学习模型的收敛性和性能的方面起到了重要的作用[7]。由于 NeuMF 是 GMF 和 MLP 的组合，我们建议使用 GMF 和 MLP 的预训练模型来初始化 NeuMF。

我们首先训练随机初始化的 GMF 和 MLP 直到模型收敛。然后，我们用它们的模型参数初始化 NeuMF 相应部分的参数。唯一的调整是在输出层，在那里我们将两者用权重连接起来：

$$h \leftarrow \begin{bmatrix} \alpha \mathbf{h}^{GMF} \\ (1-\alpha)\mathbf{h}^{MLP} \end{bmatrix}, \quad (13)$$

这里 h^{GMF} 和 h^{MLP} 分别表示 GMF 和 MLP 模型预训练的 h 向量； α 是一个超参数，用来权衡两个预训练模型（的比重）。

对于从头开始训练的 GMF 和 MLP ，我们采用自适应矩估计（Adam，Adaptive Moment Estimation）[20]，它通过对不频繁的参数进行频繁和更大幅度的更新来适应每个参数的学习速率。Adam 方法在两种模型上的收敛速度都比普通 SGD 快，并缓解了调整学习率的痛苦。在将预先训练的参数输入 NeuMF 之后，我们用普通 SGD 而不是 Adam 进行优化。 这是因为 Adam 需要保存更新参数的动量信息。因为我们用预先训练的模型参数初始化 NeuMF，并且放弃保存动量信息，不适合用基于动量的方法进一步优化 NeuMF。

4. 实验

在本节中，我们指导实验的进行，用以回答以下研究问题：

R Q 我们提出的 NCF 方法是否胜过 state-of-the-art 的隐性协同过滤方法？

R Q 我们提出的优化框架（消极样本抽样的 log loss）怎样为推荐任务服务？

R Q 更深的隐藏单元是不是有助于对用户项目交互数据的学习？

接下来，我们首先介绍实验设置，其次是回答上述三个问题。

4.1 实验设置

Database 我们尝试了两个可公开访问的数据集： MovieLens 和 Pinterest 两个数据集，它们的特征总结在表 1 中。

Table 1: Statistics of the evaluation datasets.

Dataset	Interaction#	Item#	User#	Sparsity
MovieLens	1,000,209	3,706	6,040	95.53%
Pinterest	1,500,809	9,916	55,187	99.73%

1. **MovieLens** 这个电影评级数据集被广泛地用于评估协同过滤算法。我们使用的是包含一百万个评分的版本，每个用户至少有 20 个评分。 虽然这是显性反馈数据集，但我们有意选择它来挖掘（模型）从显式反馈中学习隐性信号[21]的表现。为此，我们将其转换为隐式数据，其中每个条目被标记为 0 或 1 表示用户是否已对该项进行评级。
2. **Pinterest** 这个隐含的反馈数据的构建[8]用于评估基于内容的图像推荐。原始数据非常大但是很稀疏。 例如，超过 20%的用户只有一个 pin（pin 类似于赞一下），使得难以用来评估协同过滤算法。 因此，我们使用与 MovieLens 数据集相同的方式过滤数据集：仅保留至少有过 20 个 pin 的用户。处理后得到了包含 55,187 个用户和 1,580,809 个项目交互的数据的子集。 每个交互都表示用户是否将图像 pin 在自己的主页上。

评估方案。为了评价项目推荐的性能，我们采用了 leave-one-out 方法评估，该方法已被广泛地应用于文献[1,14,27]。即：对于每个用户，我们将其最近的一次交互作为测试集（数据集一般都有时间戳），并利用余下的培训作为训练集。由于在评估过程中为每个用户排列所有项目花费的时间太多，所以遵循一般的策略[6,21]，随机抽取 100 个不与用户进行交互的项目，将测试项目排列在这 100 个项目中。排名列表的性能由命中率（HR）和归一化折扣累积增益（NDCG）[11]来衡量。 没有特别说明的话，我们将这两个指标的排名列表截断为 10。如此一来，HR 直观地衡量测试项目是否存在于前 10 名列表中，而 NDCG 通过将较高分数指定为顶级排名来计算命中的位置。我们计算了每个测试用户的这两个指标，并求取了平均分。

对比方法。我们 NCF 方法（GMF，MLP 和 NeuMF）和下列方法进行了比较：

-Item 按项目的互动次数判断它的受欢迎程度，从而对项目进行排名。 这对基于评估推荐性能来说是一种非个性化的方法[27]。

-ItemKNN [31]。这是基于项目的标准协同过滤方法。我们遵循[19]的设置来适应隐含数据。

-BPR [27]。该方法优化了使用公式(2)的 MF 模型，该模型具有成对排序损失，BPR 调整它使其可以从隐式反馈中学习。它是项目推荐基准的有力竞争者。我们使用固定的学习率，改变它并报告了它最佳的性能。

-eALS [14]。这是项目推荐的 state-of-the-art 的 MF 方法。 它优化了公式(5)的均方误差，将所有未观察到的交互视作消极实例，并根据项目流行度对它们进行不均匀的加权。由于 eALS 显示出优于 WMF[19]均匀加权方法的性能，我们不再进一步展示报告 WMF 的性能。

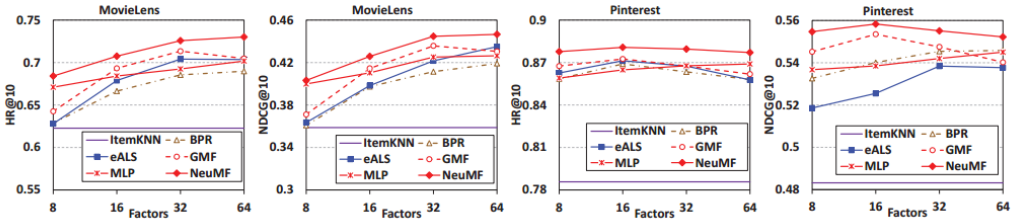
我们提出的方法旨在建模用户和项目之间的关系，我们主要比较用户-项目模型。因为性能差异可能是由个性化的用户模型引起的（因为它们是项目-项目模型），所以我们忽略了与项目-项目模型的比较，项目-项目模型有 SLIM[25] 和 CDAE[44]。

参数设置。

我们基于 Keras 实现了我们提出的方法。为了确定 NCF 方法的超参数，我们为每个用户随机抽取一个交互作为验证数据并调整其超参数。通过优化公式(7)的对数损失(log loss)来学习所有 NCF 模型，其中我们对每个积极实例进行四个消极实例的采样。对于从头开始训练的 NCF 模型，我们采用高斯分布随机初始化模型参数（平均值为 0，标准差为 0.01），用小批次 Adam[20]优化模型。

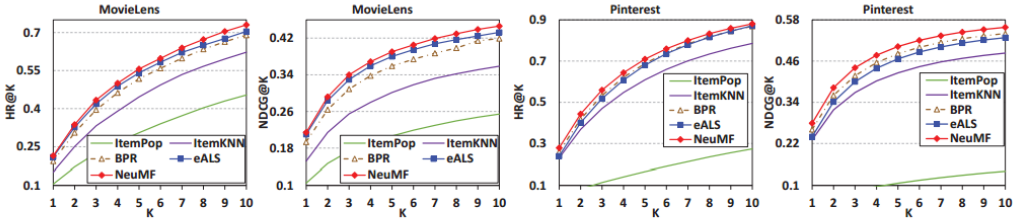
我们分别测试了批次大小[128,256,512,1024]，学习速率为[0.0001,0.0005,0.001,0.005]。 由于 NCF 的最后一层隐藏层决定了模型的性能，所以我们将其作为预测因子（predictive factors），并分别使用[8,16,32,64]的因素大小进行了模型评估。值得注意的是，大的因子可能导致过拟合，降低性能。 没有特别说明，我们采用了三层隐藏的 MLP；例如，如果预测因子的大小为 8，则神经 CF 层的结构为 32→16→8，嵌入大小为 16。对于预训练过的 NeuMF，α 设置为 0.5，允许预训练的 GMF 和 MLP 对 NeuMF 的初始化做出同样的贡献(相同权重)。

4.2 性能比较(RQ1)



(a) MovieLens — HR@10 (b) MovieLens — NDCG@10 (c) Pinterest — HR@10 (d) Pinterest — NDCG@10

Figure 4: Performance of HR@10 and NDCG@10 *w.r.t.* the number of predictive factors on the two datasets.



(a) MovieLens — HR@K (b) MovieLens — NDCG@K (c) Pinterest — HR@K (d) Pinterest — NDCG@K

Figure 5: Evaluation of Top-K item recommendation where K ranges from 1 to 10 on the two datasets.

图 4 显示了 HR@10 和 NDCG@10 相对于预测因素数量的性能。对于使用 MF 的方法 BPR 和 eALS，预测因子的数量等于潜在因素的数量。对于 ItemKNN，我们测试了不同的邻居大小并报告了最佳性能。由于 ItemPop 的性能较差，所以在图 4 中略去，以更好地凸显个性化方法的性能差异。

首先，我们可以看到，NeuMF 在两个数据集上都取得了最好的表现，远远超过了最先进的方法 eALS 和 BPR(平均而言,相对于 eALS 和 BPR 的相对改善分别为 4.5%和 4.9%）)。对于 Pinterest 而言，即使是 8 的小预测因子，NeuMF 也远远优于在 64 维潜在因子下的 eALS 和 BPR。这展现了 NeuMF 通过融合线性 MF 和非线性 MLP 模型的高水平的表达能力。其次，另外两种 NCF 方法：GMF 和 MLP，也表现出相当强劲的性能。其中，MLP 略逊于 GMF。请注意，MLP 可以通过添加更多的隐藏层来进一步改进（参见第 4.4 节），这里我们只展示使用三层的性能。对于小数目的预测因子，GMF 在两个数据集上都优于 eALS;虽然 GMF 在高纬因子的情况下受到过拟合的影响，但其获得的最佳性能优于 eALS（或差不多）。最后，GMF 显示出与 BPR 一致的进步，说明了在推荐任务的分类问题中使用 log loss 的有效性，因为 GMF 和 BPR 学习相同的 MF 模型但具有不同的目标函数。

图 5 显示了 Top-K 推荐列表的性能，排名位置 K 的范围为 1 到 10。为了使图像更加清晰，我们仅仅展示了 NeuMF 的性能，而不是所有三种 NCF 方法。可以看出，NeuMF 表现出与其他方法在位置上的一致性改进，我们进一步进行单样本配对的 t 检验，验证所有的改善对 $p < 0.01$ 有统计学意义（这里有点迷）。对于基准方法，eALS 在 MovieLens 上的性能优于 BPR 大约 5.1%，而在 NDCG 方面则逊于 BPR。 这与[14]发现吻合，它指出：由于 BPR 学习和感知成对排名，它可能在排名上表现出强劲的性能。 基于相邻（用户或者项目）的 ItemKNN 表现则逊于基于模型的方法。 而 ItemPop 表现最差，表明用户对个人喜好的建模是必要的，而不是仅向用户推荐热门项目。

4.2.1 预训练的作用

为了展示 NeuMF 预训练的效果,我们比较了 NeuMF 两种版本的性能,并且没有预训练。对于没有预训练的 NeuMF，我们使用了 Adam 随机初始化。如表 2 所示，在大多数情况下，具有预训练的 NeuMF 表现出了更好的性能；只有对于具有 8 的小预测因子的 MovieLens，

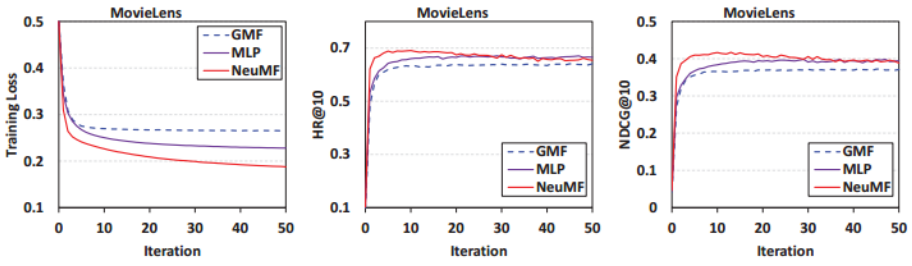
预训练方法执行稍差一些。 对于 MovieLens 和 Pinterest，NeuMF 与预训练的相对改进分别为 2.2%和 1.1%。 这个结果证明了我们的预训练方法对初始化 NeuMF 是有用的。

Table 2: Performance of NeuMF with and without pre-training.

	With Pre-training		Without Pre-training	
Factors	HR@10	NDCG@10	HR@10	NDCG@10
MovieLens				
8	0.684	0.403	**0.688**	**0.410**
16	**0.707**	**0.426**	0.696	0.420
32	**0.726**	**0.445**	0.701	0.425
64	**0.730**	**0.447**	0.705	0.426
Pinterest				
8	**0.878**	**0.555**	0.869	0.546
16	**0.880**	**0.558**	0.871	0.547
32	**0.879**	**0.555**	0.870	0.549
64	**0.877**	**0.552**	0.872	0.551

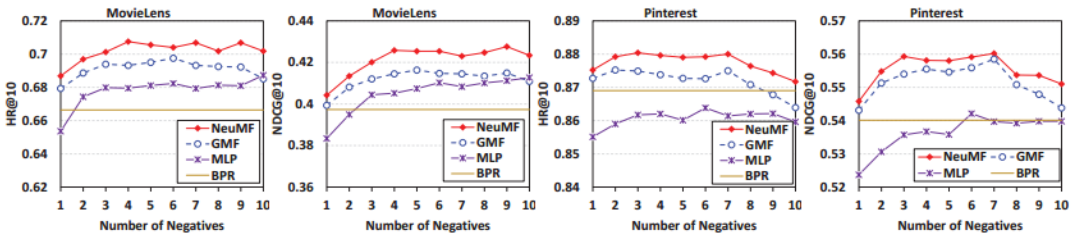
4.3 对消极采样使用 Log Loss（RQ2）

为了处理隐性反馈的一类性质，我们将推荐作为一个二分类任务。我们将 NCF 视为概率模型，使用对数损失（log loss）对其进行了优化。图 6 显示了 NCF 方法在 MovieLens 上每次迭代的训练损失（所有实例的平均值）和推荐性能。在 Pinterest 上的结果显示相同的趋势，由于空间限制而被省略。首先，我们可以看到，随着迭代次数的增多，NCF 模型的训练损失逐渐减少，推荐性能得到提高。最有效的更新发生在前 10 次迭代中，更多的迭代可能使模型过拟合（例如，虽然 NeuMF 的训练损失在 10 次迭代之后持续下降，但其推荐性能实际上降低）。其次，在三种 NCF 方法中，NeuMF 达到最低的训练损失，其次是 MLP，然后是 GMF。推荐性能也显示出与 NeuMF> MLP> GMF 相同的趋势。上述实验结果为优化从隐性数据学习的对数损失的合理性和有效性提供了经验证据。



(a) Training Loss (b) HR@10 (c) NDCG@10

Figure 6: Training loss and recommendation performance of NCF methods *w.r.t.* the number of iterations on MovieLens (factors=8).



(a) MovieLens — HR@10 (b) MovieLens — NDCG@10 (c) Pinterest — HR@10 (d) Pinterest — NDCG@10

Figure 7: Performance of NCF methods *w.r.t.* the number of negative samples per positive instance (factors=16). The performance of BPR is also shown, which samples only one negative instance to pair with a positive instance for learning.

对于成对目标函数的逐点对数损失的优势[27,33]是对于消极实例的灵活采样率。不像成对目标函数只能将一个采样消极实例与一个积极实例配对,逐点损失我们可以灵活地控制采样率。

为了说明消极采样对 NCF 方法的影响,我们在不同的消极实例采样比下展示了 NCF 方法的性能。如图 7 所示。可以清楚地看到，每个积极实例只有一个消极实例不足以达到模型的最佳性能,而抽样更多的消极实例则是有益的。将 GMF 与 BPR 进行比较,我们可以看出,采样率为 1 成的 GMF 的性能与 BPR 相当,而 GMF 在采样率较高的情况下性能明显高于 BPR。这表明了相较于成对 BPR 损失，逐点对数损失的优势所在。对于两个数据集，最佳采样率约为 3 比 6。在 Pinterest 上，我们发现当采样率大于 7 时，NCF 方法的性能开始下降。这说明，设置过大的采样率可能会对性能产生不利影响。

4.4 应用深度学习是否真的有效?(RQ3)

由于使用神经网络学习用户-项目之间交互函数的工作很少，因此，使用深层网络结构是否有利于推荐任务是值得思考的。 为此，我们进一步研究了具有不同隐藏层数的 MLP。结果总结在表 3 和表 4 中。MLP-3 表示具有三个隐藏层（除了嵌入层）的 MLP 方法，其他类似的符号具有相似的意义。 我们可以看到，即使具有相同能力的模型，堆叠更多的层也有利于性能的提升。 这个结果非常令人鼓舞，它表明使用深层模型进行协同推荐的有效性。我们将性能提升归功于堆叠更多非线性层所带来的高非线性度。为了验证这一点，我们进一步尝试堆叠线性层,使用恒等函数作为激活函数。性能比使用 ReLU 单元差很多。

Table 4: NDCG@10 of MLP with different layers.

Factors	MLP-0	MLP-1	MLP-2	MLP-3	MLP-4
MovieLens					
8	0.253	0.359	0.383	0.399	**0.406**
16	0.252	0.391	0.402	0.410	**0.415**
32	0.252	0.406	0.410	**0.425**	0.423
64	0.251	0.409	0.417	0.426	**0.432**
Pinterest					
8	0.141	0.526	0.534	0.536	**0.539**
16	0.141	0.532	0.536	0.538	**0.544**
32	0.142	0.537	0.538	0.542	**0.546**
64	0.141	0.538	0.542	0.545	**0.550**

对于没有隐藏层（即，嵌入层直接映射到预测结果）的 MLP-0，性能非常弱，不过比非个性化 ItemPop 更好。这验证了我们在 3.3 节中的观点：简单地连接用户和项目的潜在向量不足以对其特征的相互作用进行建模，因此需要使用隐藏层进行变换。

5. 相关工作

关于推荐的早期文献主要集中在显性反馈[30,31],而最近工作的重心越来越多地转向隐性数据[1,14,23]。具有隐性反馈的协同过滤（CF）任务通常被转化为项目推荐问题，其目的是向用户推荐一个简短的项目列表。 与被广泛采用的通过显性反馈进行评级预测相反，解决项目推荐的问题更具实践性，但同时也富有挑战性[1,11]。 一个关键的先验是对丢失的数据进行建模，这些数据在显式反馈的工作中常常被忽略[21,48]。为了建立具有隐性反馈的项目推荐的潜在因素模型，早期工作[19,27]使用均匀加权，其中提出了两种策略，即将所有缺失数据作为消极实例[19]或从缺失数据中抽样作为消极实例[27]。

最近，He 等[14]和 Liang 等人[23]提出了专门的模型来减少丢失的数据，而 Rendle 等人[1]为基于特征的因式分解模型开发了隐式坐标下降解法（iCD），实现了当前最高水平的项目推荐。下面我们将讨论使用神经网络的推荐工作。

早期的先驱工作是由 Salakhutdinov 等人[30]提出的一种两层限制玻尔兹曼机(RBM)来模拟用户对项目的明确评级。这项工作后来被扩展到对等级的序数性质进行建模[36]。最近，自动编码器（autoencoders）已成为构建推荐系统的一般选择[32,22,35]。基于用户的 AutoRec[32]的想法是学习隐藏的结构，根据用户的历史评级作为输入，可以重建用户的评分。在用户个性化方面，这种方法与项目-项目模型[31,25]具有相似之处，它表示用户作为其评分项目。为了避免自动编码器学习恒等函数，并且不能将其推广到不可观察的数据，它已经应用了去噪自动编码器(DAE)来学习人为损坏的输入[22,35]。最近，Zheng 等人[48]提出了一种用于 CF 的神经自回归（neural autoregressive ）方法。虽然以前的努力证明了神经网络解决 CF 的有效性，但大多数重点放在明确的评级上，并仅对观察到的数据进行建模。因此，他们不易于从只有积极样例的隐性数据中学习用户的偏好。

虽然最近的工作[6,37,38,43,45]已经探索了基于隐性反馈的深入学习模型，但它们主要使用 DNN 来建模辅助信息，例如文字描述[38]，音乐的声学特征[37,43]，用户的跨域行为[6]以及知识库中的丰富信息[45]。然后由 MF 和 CF 相结合 DNN 学习特征。与我们工作最相关的工作是[44]，它为 CF 提供了一个隐性反馈的协同去噪自动编码器（CDAE）。与基于 DAE 的 CF [35]相反，CDAE 还将用户节点插入自动编码器的输入端，以重建用户的评级。如作者所展示的，当使用恒等函数作为激活函数作用在 CDAE 的隐藏层时，CDAE 相当于 SVD++模型[21]。这意味着虽然 CDAE 虽然是 CF 的神经建模方法，但是它仍然使用线性内核（即内积）来模拟用户-项目交互。这可能部分解释了为什么使用深层 CDAE 不会提高性能（参见[44]第 6 节）。与 CDAE 不同，我们的 NCF 采用双通道结构，建立多层前馈神经网络的模型模拟用户项目交互。这允许 NCF 从数据中学习任意函数，比固定内部函数有更强大的表达能力。

顺着相似的研究方向,学习两个实体的关系已经在知识图的文献中得到了深入的研究[2,33]。 许多有关的机器学习方法[24]已经被设计出来。 与我们提出的方案最相似的是神经张量网络（NTN，Neural Tensor Network）[33]，它使用神经网络来学习两个实体的相互作用并表现出很强劲的性能。这里我们专注于=它与 CF 的不同设置问题。 尽管将 MF 与 MLP 结合在一起的 NeuMF 的想法是受到了 NTN 的一定程度的启发，但是我们的 NeuMF 比 NTN 更灵活和通用，它允许 MF 和 MLP 学习不同的嵌入式组合。

最近，谷歌公布了他们的通用深度学习方法的 App 推荐[4]。 深层组件类似地使用特征嵌入的 MLP，据称具有很强的泛化能力。 虽然他们的工作集中在结合用户和项目的各种特征，但我们的目标是使用 DNN 探索纯粹的协同过滤系统的。我们显示了对于建模用户-项目交互，DNN 是一个有前途的选择，以前没有对此进行调查。

6. 总结和展望

在这项工作中，我们探索了用于协同过滤的神经网络结构。我们设计了一个通用框架 NCF，并提出了三种实例：GMF，MLP 和 NeuMF，以不同的方式模拟用户-项目交互。我们的框架简单而通用；它不仅限于本文提出的模型，它对深入学习推荐方法的也具有指导意义。这项工作补充了主流浅层协同过滤模型，为深入学习推荐研究开辟了新途径。

在将来的工作中，我们将研究 NCF 模型在成对学习中的应用，并将 NCF 扩展到建模辅助信息，如用户评论[11]，知识库[45]和时间信号[1]。现有的个性化模式主要集中在个人身上，为用户群体开发模型将会是一个有趣的发展，这有助于社会团体的决策[15,42]。此外，我们特别感兴趣的是建立多媒体项目的推荐系统，这是一个有趣的任务，但在推荐

社区中受到相对较少的关注[3]。多媒体项目（如图像和视频）包含更丰富的视觉语义[16,41]，可以反映用户的兴趣。 要构建多媒体推荐系统，我们需要开发有效的方法来学习多视图和多模态数据[13,40]。 另一个新出现的方向是探索循环神经网络和散列方法（hashing methods）[46]的潜力，以提供有效的在线推荐[14]。

鸣谢

作者感谢匿名评审者的宝贵意见,给作者对推荐系统的思考和论文的修订提供了极大的帮助。

四、外文原文

毕业论文（设计）文献综述和开题报告考核

导师对开题报告、外文翻译和文献综述的评语及成绩评定：

成绩比例	文献综述（10%）	开题报告（15%）	外文翻译（5%）
分 值			

导师签名_____

年　　月　　日

学院盲审专家对开题报告、外文翻译和文献综述的评语及成绩评定：

成绩比例	文献综述（10%）	开题报告（15%）	外文翻译（5%）
分 值			

开题报告审核负责人（签名/签章）_____

年　　月　　日