 单位代码

学 号



**硕士学位论文**

基于CV水平集图像分割研究

论文作者：高名衍

指导教师：唐雁教授

学科专业：计算机应用技术

研究方向：网络应用与web智能

提交论文日期： 年 月 日

论文答辩日期： 年 月 日

学位授予单位：西南大学

中 国 • 重 庆

年 月

**独创性声明**

学位论文题目：

本人提交的学位论文是在导师指导下进行的研究工作及取得的研究成果。论文中引用他人已经发表或出版过的研究成果，文中已加了特别标注。对本研究及学位论文撰写曾做出贡献的老师、朋友、同仁在文中作了明确说明并表示衷心感谢。

学位论文作者： 签字日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解西南大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权西南大学研究生院（筹）可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书，本论文：□不保密，□保密期限至 年 月止） 。

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期：

目 录

[摘要 I](#_Toc9526664)

[Abstract III](#_Toc9526665)

[第1章 绪论 1](#_Toc9526666)

[1.1研究背景 1](#_Toc9526667)

[1.2国内外研究现状 2](#_Toc9526668)

[1.3主要工作 4](#_Toc9526669)

[1.4论文结构 5](#_Toc9526670)

[第2章 相关理论及技术 7](#_Toc9526671)

[2.1协同过滤推荐技术 7](#_Toc9526672)

[2.2浅层推荐模型 8](#_Toc9526673)

[2.2.1传统矩阵分解推荐模型 8](#_Toc9526674)

[2.2.2基于神经网络的矩阵分解模型 11](#_Toc9526675)

[2.3深层推荐模型 12](#_Toc9526676)

[2.3.1 多层感知机推荐模型 12](#_Toc9526677)

[2.3.2深度矩阵分解推荐模型 14](#_Toc9526678)

[2.3.3深度神经协同过滤模型 16](#_Toc9526679)

[2.4注意力推荐模型 18](#_Toc9526680)

[2.4.1注意力机制 18](#_Toc9526681)

[2.4.2注意力因子机模型 20](#_Toc9526682)

[2.5本章小结 22](#_Toc9526683)

[第3章 融合辅助信息的多交互深度矩阵分解模型 23](#_Toc9526684)

[3.1模型介绍 23](#_Toc9526685)

[3.1.1模型输入 24](#_Toc9526686)

[3.1.2多交互网络结构 25](#_Toc9526687)

[3.1.3激活函数 26](#_Toc9526688)

[3.1.4模型输出 26](#_Toc9526689)

[3.2学习过程 26](#_Toc9526690)

[3.2.1目标函数 26](#_Toc9526691)

[3.2.2模型训练 27](#_Toc9526692)

[3.2.3模型优化 27](#_Toc9526693)

[3.3实验 28](#_Toc9526694)

[3.3.1实验环境和实验数据 28](#_Toc9526695)

[3.3.2评价标准 28](#_Toc9526696)

[3.3.3实验内容 29](#_Toc9526697)

[3.3.4对比实验 30](#_Toc9526698)

[3.3.5参数分析 31](#_Toc9526699)

[3.4本章小结 33](#_Toc9526700)

[第4章 融合注意力机制的多交互神经矩阵分解模型 35](#_Toc9526701)

[4.1模型介绍 35](#_Toc9526702)

[4.1.1模型输入 36](#_Toc9526703)

[4.1.2浅层/深层推荐结构 37](#_Toc9526704)

[4.1.3注意力网络 38](#_Toc9526705)

[4.1.4模型输出 38](#_Toc9526706)

[4.2学习过程 39](#_Toc9526707)

[4.2.1目标函数 39](#_Toc9526708)

[4.2.2模型训练 39](#_Toc9526709)

[4.2.3模型优化 40](#_Toc9526710)

[4.3实验 40](#_Toc9526711)

[4.3.1实验环境和实验数据 40](#_Toc9526712)

[4.3.2评价标准 41](#_Toc9526713)

[4.3.3实验内容 42](#_Toc9526714)

[4.3.4对比实验 43](#_Toc9526715)

[4.3.5参数分析 44](#_Toc9526716)

[4.4本章小结 46](#_Toc9526717)

[第5章 总结与展望 47](#_Toc9526718)

[5.1总结 47](#_Toc9526719)

[5.2展望 48](#_Toc9526720)

[参考文献 49](#_Toc9526721)

[致谢 53](#_Toc9526722)

[攻读硕士学位期间公开发表的论文 55](#_Toc9526723)

**基于CV水平集图像分割研究**

**计算机软件与理论 硕士研究生 高名衍**

**指导老师 唐雁 教授**

# 摘要

个性化推荐技术主要根据用户需求和偏好等信息，通过相关推荐算法获取用户对项目的偏好特征，并向用户推荐其可能感兴趣的项目；在缓解“信息过载”问题上起到了重要的作用，被广泛应用于各个领域，备受广大研究者关注。

其中，基于协同过滤的推荐技术应用和研究最为广泛，尤其是矩阵分解(MF)和SVD++等算法在Netfix大赛中取得良好成绩，将基于矩阵分解的协同过滤模型研究推上了高潮。但是，该模型仅仅考虑用户与项目的评分数据，存在的数据稀疏性和冷启动问题会很大程度上影响到模型的推荐性能；作为一个浅层推荐模型，也无法获取到用户与项目更深层次的隐藏特征与交互信息去进行更精准的推荐。

随着深度学习技术的不断发展和突破，也为推荐模型的研究带来新的机遇和挑战。基于深度学习的推荐模型利用多层神经网络结构，能够非线性地学习用户与项目的交互信息，获得更深层次、更抽象的隐藏特征表示，表现出较好的推荐效果。但是，现有模型大多基于矩阵分解的思想，使用单一的评分信息在遇到数据稀疏性问题时会降低模型的推荐性能；只对多层神经网络最后的隐表示进行交互，没有考虑到每层网络所学习到的特征表示都是非常重要的；模型训练时对用户和项目所有的特征交互都采用相同的权重，但对于不同的特征交互，其重要程度不同。

本文针对上述问题展开研究，融合辅助信息缓解稀疏性问题，构建新的网络结构有效利用每层网络所学习到的隐表示，引入注意力机制去判断特征交互的重要程度，将浅层模型和深层模型相结合进一步提升模型的推荐性能。主要工作如下：

（1）针对评分数据的稀疏性问题和只利用最后的隐表示进行推荐，提出融合辅助信息的多交互深度矩阵分解模型。在基于矩阵分解的深度学习推荐模型基础上，使用用户和项目的评分信息和更多的辅助信息（用户/项目属性，评论，标签等）作为模型的输入数据，辅助信息中包含用户与项目一定的隐藏偏好特征，不但能缓解模型的数据稀疏性问题，还能从辅助信息中抽取更多用户和项目的属性特征；通过构建并行的多层非线性网络分别学习用户和项目的隐表示，并且对每一层网络所学习到的用户和项目隐表示都进行一次点积操作，利用网络层不同的学习能力获得不同层的交互结果；聚合所有层的交互结果作为所提模型的最终结果并进行评分预测。在Movielens latest small和Movielens latest 20m数据集上进行对比实验并对该模型相关的参数进行对比分析，实验结果表明，所提出的推荐模型能更准确地进行评分预测。

（2）针对不同特征交互具有不同重要程度，结合浅层模型与深层模型进行推荐，提出融合注意力机制的多交互神经矩阵分解模型。基于工作(1)所提出的多交互网络结构，获得每一层网络所学习到的用户和项目交互结果；将多层神经网络学习到的用户和项目隐表示作为输入，增加一个注意力网络（Attention）去学习一个注意力权重矩阵，用来判断用户和项目特征交互的重要程度，将多交互网络结构学习到的隐表示与注意力权重矩阵加权得到一个深层模型的推荐结果；然后基于浅层的矩阵分解模型思想，与上述的深层模型共享同一嵌入层，获得用户和项目的隐因子表示，直接采用点积操作得到一个浅层模型的推荐结果；最后结合线性的浅层模型与非线性的深层模型，对浅层推荐模型和深层推荐模型的结果进行加权得到所提模型最终结果并进行Top\_N排序推荐。在Movielens 1m和Pinterest数据集上进行对比实验并对该模型相关的参数进行对比分析，实验结果表明，所提出的推荐模型能更精确地为用户进行合适的项目推荐。

**关键词：深度学习；辅助信息；多层交互；矩阵分解；注意力机制**

**Research on Recommendation Model Based on**

**Deep Learning**

***Major:*** Technology of Computer Application

***Supervisor:*** Prof. Yan Tang

***Author:*** Tonghuan Li

# Abstract

The personalized recommendation technology mainly obtains the user's preference characteristics for items by recommendation algorithm and recommends items to users according to user's demands, preferences or other information. It plays an important role in alleviating the problem of "information overload". It has also applied to various fields and attracted much attention of researchers.

In personalized recommendation system, the recommendation model based on collaborative filtering was used widely, especially matrix factorization (MF), SVD++ have made many achievements in Netfix competition, and pushed the research on collaborative filtering model based on matrix factorization to climax. However, the model just only considers the rating information between users and items, and the problem of data sparsely and cold start will greatly affect the recommendation performance of the model. In addition, as a shallow model, it can’t to learn the deeper features of users and items, and the interaction information between users and items to make more accurate recommendation.

With the development and breakthrough of deep learning technology, it has brought new opportunities and challenges for the research of recommendation model. The recommendation model based on deep learning utilizes the structure of multi-layer neural network, which can learn the interactive information between users and items non-linearly, obtain deeper and more abstract hidden feature representations, and show better result. But, most of models are based on the idea of matrix factorization, using a single rating information will reduce the recommendation performance of the model when encountering the problem of data sparsely; only the last latent representation of multi-layer neural networks is interacted, without considering the feature representation learned by each network is also important. In model training, the same weight is applied to all feature interactions between users and items, but for different feature interactions, the importance is different.

In this paper, research on the above problems, combines auxiliary information to alleviate the sparsely problem, constructs a new network structure to effectively utilize the latent representations learned by each layer of network, adding the attention mechanism to judge the importance of different feature interaction, and further improves the recommendation performance of the model by combining shallow model and deep model. The main work of this paper is as follows:

(1) Considering the problem of the rating data sparsely and only using the last latent representation to recommend, we proposed a Multi-interactive Deep Matrix Factorization Model Based on Auxiliary Information. On the basis of deep learning recommendation model based on matrix factorization, we further utilize the rating data and fuse more auxiliary information (user/item attributes, comments, tags, etc.) as the model input, which contains some hidden preference features of users and items, it can not only alleviate the problem of data sparsely, but also extract more attributes of users and items from auxiliary information. Then, we construct parallel multi-layer non-linear network to learn latent representation of users and items respectively, and utilizing dot product operation for latent representation of users and items learned by each layer of network, considering different learning abilities of network layer to obtain interactive results of different layers. Finally, aggregate the interaction results of all layers as the model final result and predict the score. After compared the experiment on Movielens latest 100K dataset and analyzed the model related parameters, the experimental results have proved that the proposed recommendation model can accurately predict the score.

(2) Considering the importance of different interaction features, and further combine the shallow model and deep model to recommend, we proposed an attention multi-interactive neural matrix factorization model. Firstly, based on the multi-interactive network structure proposed by work (1), to obtain the interactive results of users and items learned by each layer of network; using the latent representations learned by multi-layer neural networks as input. Adding an attention network to learn an attention weight matrix, which is used to judge the importance of interaction between user and item features; utilizing the latent representations that learned from the multi-interactive network and the attention weight matrix, to obtain a result of deep model. Secondly, based on the idea of shallow matrix factorization model, we share the same embedding layer with the deep model mentioned above, obtain the latent factor representation of users and items, and directly use the dot product operation to obtain a result of shallow model. Finally, combine the linear shallow model and non-linear deep model, weighted the shallow result and the deep result to obtain the final results of the proposed model and make Top\_N rank recommendation. After compared the experiment on Movielens 1m and Pinterest datasets, and analyzed the model related parameters. The experimental results have proved that the proposed recommendation model can more accurately to recommend items for users.

**Keywords: deep learning; auxiliary information; multi-interaction; matrix factorization; attention mechanism;**

# 

# 第1章 绪论

## 1.1研究背景

科学技术的迅猛发展以及互联网的日益普及，提高了人们信息交互的效率，而图像作为信息的重要载体，满足了人们获取信息的需求，为人们的生活提供了极大的方便[1]，例如，医学图像中的核磁共振图像(MRI)能够辅助医生找到病灶，人脸、指纹图像能够应用于安保联防以及智能支付，数字图像应用于车牌号码识别，视频图像跟踪应用于农业产业自动化以及社交媒体上传图片记录生活中美好瞬间[2]等等。从以上这些方面可以看出图像渗透到人们生活的方方面面，因此，如何通过图像处理技术得到我们所需的信息就显得尤为重要。



图1-1图像处理工程结构

图像处理技术范围很广，是一门多学科交叉的技术，涉及到学科包括生物医学、数学、信息论、计算机科学等多种学科，其技术内容包含了图像分割、图像压缩、图像复原重建、图像分类识别、图像合成等。其中，图像分割是图像处理技术的一个重要环节，如图1-1所示，目的是将图像中目标区域提取出来，为后续的图像处理打下良好的基础。图像分割定义如下：

我们将图像看作各个不同的子区域的结合成的二维空间区域[3]，符合如下关系式：

1. 。



1. 是一个连通集， 。



1. ，对于所有和，



1. ，



1. ，对于任何 和 的邻接区域



其中和表示并集和交集，表示空集， 表示 集合像素点的属性。



图像分割理论上要完全满足上述五个条件，这极大增加了图像分割算法的难度和复杂性，如何攻克这些难题一直是图像分割研究方向的重点。

通过以上分析以及图1-1所展示的内容，可以看出图像分割是呈现目标区域的基础，对其它特征分析起着至关重要的作用，并且，排出了其它背景因素的干扰，为图像进一步分析提供了可能。

图像分割的应用范围非常广泛，例如，人脸识别过程中找出人像区域，医学影像中找到肿瘤、血管位置以及相应器官组织大小的测量，卫星遥感图像中找到道路、河流、森林路径区域，目标检测中定位追踪目标，交通控制系统中找到车牌等。图像分割应用范围之广、应用场景之复杂，也为图像分割的要求提出了严峻挑战，而图像采集以及传播过程中，难免会引入噪声、像素缺失等问题，使图像变得模糊、灰度不均匀，那么如何克服这些不利因素，更加精准快速的得到目标区域是图像分割领域亟待解决的问题。

## 1.2国内外研究现状

图像分割自成像技术出现以来就开始发展，计算机图像分割算法中最早出现在二十世纪七十年代，多年来图像分割算法在国内外专家学者的研究下发展迅速，各种经典算法层出不穷，伴随着计算机硬件运算能力的改进、人工智能时代的来临，涌现出了大量优秀的研究成果，但众所周知，图像类别多样、场景复杂、图像中存在的干扰因素也多种多样，所以学界并没有一个万能的图像分割理论适用于所有类型的图像，我们按照公认的分类方式，将国内外经典的、效果显著的图像分割算法分为如下几类：

(1)基于阈值的分割方法

阈值分割法比较简单，主要是利用灰度直方图达到分割目的[]，其基本原理是统计出待处理图像每个灰度值的分布规律映射到灰度直方图上，之后，选取其中的某几个阈值来划定图像不同的区域，从而将图像中的目标区域确定下来，达到分割的作用。可见，如何找到最优阈值是该算法性能提升的关键所在。阈值分割法的研究思路主要是从全局阈值和局部阈值两个方向展开。全局阈值法利用待处理图像全局信息统计灰度直方图，然后选出一个或几个阈值进行切分；局部阈值法是先将大图切分成小图，再对小图进行全局阈值法选取最优阈值。总的来说，阈值分割法简单高效，但在应对灰度不均匀、场景复杂的图像时，该方法分割效果较差。常见的阈值分割法有：最大熵统计阈值法、最小误差法等[]。

(2)基于边缘检测的分割方法

边缘检测法也是图像分割领域经典的分割方法。观察图像，我们可以发现在图像不同区域的交界处会发生灰度值的突变，这就表明图像像素区域已从背景区域进入目标区域。边缘检测法就是采用这样的思想，找到灰度不连续的地方，这种图像灰度值的阶跃性可以通过导数来检测，因此，常用如下微分算子边缘位置进行检测[]：Roberts算子、Sobel算子、Laplace算子以及Kirsh算子等。每个算子各具特点，针对一定类型图像的处理效果较好，Laplace算子的特点是各向同性；Prewitt算子、Sobel算子等对噪声图像以及灰度较均匀的图像分割效果较好；Roberts算子适用于前景背景灰度像素差别大并且噪声较少的图片。总的来说，边缘检测法的优势在于检测速度快、准确，但是对图像的要求比较高，噪声以及灰度不均匀等因素会严重影响该类型算法的准确度。

(3)基于区域的分割方法

区域分割法的主要思路是依据图片的特征划将图片划分为不同的子图，其主要的算法是：区域生长法，区域分裂法以及区域合并法。区域生长法[]的思想是将大图切成小图，找到各个其中相似性较高的子图进行合并，合并完成后便得到不同特征的区域达到了分割的目的。区域生长法的关键点在于切分成多少子图，依据怎样的特征来合并这些区域以及这些特征相似性的准则。该算法的优势在于简单易行，而缺点是需要手工调整种子数。区域分裂合并法的主要思想仿照了四叉树数据结构，将原始图像定义为TreeRoot，将大图切分成四个子图作为四叉树的第一层，然后再对这四个子图进行细分，若其中的子图满足一致性条件则不再细分，若不满足，则进一步将其划分为四个子图，若子图中存在某几块具有一致性特点，则可以讲它们合并，直到所有子图都满足灰度一致性的条件。该方法对复杂图片的分割效果较好，但是计算量大，可能会对前景背景分界区造成破坏。

(4)基于多学科交叉的分割方法

正如1.1小节所论述的内容，图像分割技术是多学科交叉的技术，在图像分割中融合其它学科思想也是提高分割性能的一种思路，随着计算机硬件算力的提升，越来越多的新型算法涌现了出来，例如，基于数学形态学的分割方法，基于人工神经网络的分割方法，基于模糊集理论的分割方法等等，方法不一而足。这些算法各有利弊，例如，数学形态学的方法对噪声比较敏感[]，人工神经网络分割方法比较准确，但运算周期长、依赖硬件设备。总的来说，我们很难找到一种普适性的图像分割算法适用于所有图像。

(5)基于主动轮廓模型的图像分割方法

基于主动轮廓模型的方法是对传统计算机视觉分层模型理论的一种颠覆式创新，通过构建能量函数，利用局部极值点与模型的交互，并不断迭代其中的能量项来达到满足图像处理成果的目的。在这种思想的引导下，Kass等[]于1987年提出了Active Contour Model，即主动轮廓模型，又称Snake模型。多年来，学者们对其进行研究和创新，提出了各种改进方法，主要分为两类：参数主动轮廓模型和几何主动轮廓模型。

参数主动轮廓模型主要是通过调整和变化曲线表达式的参数改变曲线形状的目的，其主要形变推动力是依靠图像数据得到的能量泛函偏微分方程以及图像的先验模型。但是，该模型的缺点也显而易见，曲线的拓扑结构变化难以处理，参数变化的计算极其复杂[]，按照这样的思路进行优化就更加难以推进。

为了有效解决了上述问题，几何主动轮廓模型应运而生,它与水平集方法相结合，将高维曲面的零水平集作为分割曲线，不断演化得到最终的目标区域分割线，其曲线拓扑结构的变化也非常的简单自然，可以说，水平集方法有效推动了无参数几何主动轮廓模型的发展。水平集方法创始于1988年，由美国加州大学Osher教授和Sethian教授首次提出[]，该方法的主要思想是构建出高维的函数曲面，通过该曲面不断的演化找到低维的目标区域，其演化的主要方式是通过水平集函数的偏微分方程对其进行数值上的实现，这样就可以避免通过调参这样繁琐复杂的方式对曲面方程进行调整。水平集方法创立之后便由Caselle[]等人应用到了活动轮廓模型当中，两者结合共同构成了基于水平集的几何活动轮廓模型，该方法集成了水平集算法的优点，摒弃了分割曲线在演化过程中繁琐的调参计算，通过高维曲面向低维曲面映射的思路，巧妙的处理了分割曲线拓扑结构变化的问题。该思想于1993年提出并延续发展至今，它以强大的数学理论、良好的信息包容性、平滑灵活的分割曲线等优势，逐渐超越参数活动轮廓模型成为活动轮廓模型研究热点，同时，也成为了图像分割理论中非常重要的一个分支。近年来，基于水平集方法的研究理论层出不穷，是图像分割领域热门研究课题。学界对于基于水平集的分割方法分为如下两类：

(a)基于边界的轮廓模型

该模型的主要根据边界区域的陡峭程度来调整曲线的演化速度，当图像边界区域灰度变化较小，曲线演化速度就非常快，当图像边界区域的灰度变化处于突变状态时，曲线演化速度就变慢甚至停止演化。Caselles和Kimmel[]依据能量最小化的思路，创造性的将水平集方法首次引入活动轮廓模型，提出了Geometric Active Contour Model，成为了边界轮廓模型的经典。但该模型针对边界清晰的图像分割效果较好，如果图像灰度不均匀、目标区域不是特别明显的话，会出现误分割的现象。Siddiqi等人[]在上述模型的基础上进行调整，增加了面积约束项，能够有效解决图像边界存在的较小缝隙，但如果图像目标区域边界缝隙较大，也会造成该模型分割失败。Liu[]等人将GAC模型与图像的局部区域信息相结合，并重构边界停止函数因子，提高了对目标区域边界的识别能力以及抗躁能力。

总的来说，基于边界的轮廓模型依然是水平集分割方法研究的热门方向，近年来还是有很多好的改进方法被提出，其分割效果优劣的关键在于目标区域的边界是否清晰，如果没有明确的目标区域边界，会严重影响分割效果，而且如果图像中存在噪声，对该模型的分割效果影响也较大。

(b)基于区域的轮廓模型

区域轮廓模型结合了图像的区域统计信息以及边界梯度信息，避免过度依赖目标区域边界而引起的上述问题。区域轮廓模型又可以细分为两类：全局区域和局部区域[]。全局区域轮廓模型中最著名的当属由Mumford和Shah两人提出的Mumford-Shah模型[]，该模型最大的缺陷就是计算复杂度太高，数值求解非常繁琐。为了解决这个问题，Chan和Vese两人在2001年提出了C-V模型[]，相当于简化版的M-S模型，它计算简单、复杂度低，曲线收敛速度快、效果好，充分利用了图像局部梯度以及同质区域的全局信息，堪称最经典的全局模型。但该模型非常依赖初始轮廓曲线位置，若初始轮廓线选择的区域不好，将会影响分割结果。并且，待检测的图像往往前景背景灰度不均匀，这也会造成分割的失败。为了解决这样的问题，研究者采取了将图像局部信息引入模型的思路。Li[]等人将活动轮廓模型与局部邻域思想相结合，提出了Local Binary Fitting Model，即LBF模型。该模型较好的处理了灰度不均匀图像的分割问题，之后进一步改进提出了RSF模型。Li提出的模型运用核函数进行卷积运算，使得计算非常复杂，且没有很好的解决初始轮廓曲线对分割结果带来的影响。Zhang[]等人提出了Local Image Fitting Model，即LIF模型，克服了计算复杂度高的问题，但曲线演化速度仍待提升，还需要解决初始轮廓线的问题。Wang[]等人提出了Local Gaussian Distribution Fitting Energy Model，即LGDF模型，该模型利用高斯分布来表示图像局部灰度值，同时定义了局部高斯适应能量项与水平集函数相结合。该方法计算量较大，对初始轮廓线敏感。袁建军等人[]提出了亮度信息和梯度信息相结合的活动轮廓模型，贾迪野等人[]对邻域像素点进行扫描处理，优化Mumford-Shah模型，减少了演化方程计算复杂度。王慧斌等人[]引入了图像纹理特征信息，提出多区域活动轮廓模型，有效提高了分割效率。

当前，国内外主要研究水平集图像分割方法的团队有：Stanley Osher教授和James A.Sethian教授领导的加州大学伯克利分校团队；Tony Chan教授领导的加州大学洛杉矶分校的团队，李纯明教授所在的美国Vanderbilt University图像科学部，以及上海交通大学图像处理与模式识别研究所，重庆大学光电技术及系统教育部重点实验室等

## 1.3主要工作

目前，广大学者对于基于CV模型的水平集图像分割算法进行了大量研究，提出的各种改进算法也各具成效，但总体来说，依然面临如下几点不足之处：(1)初始轮廓线问题一直都是影响水平集图像分割效果的重大问题，根据大量的实验研究发现同样的模型，如果初始轮廓线不同，会造成分割出的结果千差万别。(2)算法复杂度高、计算过程时间长，尤其是结合图像局部信息的一类算法，计算过于复杂，算法迭代次数较多。(3)大部分模型只是对能量函数进行改进，往往只能解决某一种问题，如果考虑了局部信息，分割结果容易陷入局部极小值，如果考虑全局信息则容易受到图像灰度不均匀的影响。(4)图像灰度不均匀以及噪声问题会影响分割的准确度。针对上述问题，本文做了如下两点工作：

(1)提出了高斯混合模型与CV模型相结合的GMM-CV模型，利用高斯混合模型稳定快速的像素识别能力对图像进行目标前景和背景的检测，得到初始轮廓区域，然后通过该先验信息计算出目标前景灰度均值 和背景灰度均值融入到能量泛函公式中，与原始的CV能量项进行权重分配，并引入距离惩罚项和长度惩罚项，共同构成全新能量函数 。该方法克服了初始轮廓曲线选择不当对分割结果的影响，并将高斯混合模型处理得到的先验信息融入到能量泛函中，进一步优化了分割的准确程度。我们利用数据集BSDS500数据集对GMM-CV模型进行了测试，实验结果表明GMM-CV模型取得了良好的实验成果。



(2)本文在(1)的基础上做了大幅度改进，综合了图像全局信息和局部信息提出了多尺度RSF-CV模型，首先考虑到噪声因素对图像分割的影响，将KMeans聚类方法引入到模型中对图像进行前景和背景检测，替换掉之前的高斯混合模型，因为KMeans算法的优势可以得到两个聚簇值 和 ，更加精准求出了前景和背景平均灰度值。能量泛函公式采用改进的CV模型Region-Scaling Fitting Model，简称RSF模型，将图像的局部信息考虑在内以降低图像灰度不均匀、噪声因素带来的影响，然后再加入CV能量项将图像全局信息纳入公式中以防止分割曲线陷入局部极小值，共同构成 能量泛函公式。为了能够在图像的不同区域有效调节CV全局能量项和RSF局部能量项的比重，也就是和的值，本文参考了LBP算法的思想，当分割曲线处在背景区域时，曲线的演化以全局能量项为主，当分割曲线处在目标区域附近时，曲线的演化以局部能量项为主。最后，再加入长度惩罚项和距离惩罚项来保证分割曲线拓扑结构的平滑自然。我们用BSDS500数据集和STARE数据集对新模型进行测试，结果显示该模型的分割效果良好，实验结果优于其它模型。



## 1.4论文结构

本文共分成五个章节，每个章节内容如下：

**第1章**：绪论。主要概述了图像分割的研究背景及意义，总结并归纳国内外研究进展并主要分析了水平集图像分割算法的研究成果，介绍了本文所做的主要工作内容，最后简单阐明了全文架构。

**第2章：**相关理论及技术。主要是对本文所涉及到的模型算法以及理论思想进行介绍，主要包括水平集理论以及经典的CV模型和RSF模型，图像前景区域检测的高斯混合模型、KMeans聚类算法，权重调整用到的LBP算法。

**第3章**：提出GMM-CV模型。对该模型的整体结构和算法思想进行了详细说明，并展示了算法流程以及图像分割的结果，在公开数据集上进行了测试，并对比了其它水平集图像分割模型。

**第4章**：提出了多尺度RSF-CV混合模型。与第3章结构类似，主要对模型进行了详细介绍，并展示了算法流程以及图像分割的结果，在公开数据集上测试了该模型并与其它图像分割模型进行了对比分析。

**第5章：**总结与未来展望。将现阶段所做的工作和研究成果进行总结，分析和讨论了模型中存在的问题，阐述了优化的思路和方向，并提出了对未来研究工作的展望。

# 第2章 相关理论及技术

## 2.1曲线的演化

曲线的演化是通过单位法矢量和曲率来描述曲线拓扑结构随时间变化的，其中单位法矢量表示各点运动的方向，而曲率则表示曲线的弯曲程度，从图2-1中，可以看出二者的几何关系是互相垂直的[]。这样，曲线就可以在二维图像中受某些驱动力的作用下，随时间逐渐扩张或收缩到我们所想要的目标边界。



图2-1某点的切线和法线

如图2-1所示，定义一条曲线 随时间发生形变，参数 为曲线参数，参数 为曲率，曲线的切线和法线分别用和来表示。



(2-1)



(2-2)



(2-3)

曲线上的点可以通过法向量和切向量的组合来表示，将时间引入到曲线表达式中，对时间求导可得到(2-3)所示公式，其中，和分别表示切向和法向的速率。但是，切向速率并不能引起曲线形状的变化，因此可简化方程为：

(2-4)



其中，和分别表示演化的方向和速度，的正负则表示曲线运动方向是向内收缩和向外扩张。按照的不同取值，可以进一步将曲线演化分为曲率演化和常量演化：若，则为曲率演化，方程表达式为：



(2-5)

其中为系数，表示曲率，反映了曲线弯曲程度。曲线各部分的演化速度根据弯曲程度而变化不一，因此，在曲率的推动下，整条曲线会随着时间逐渐演化近似于圆形；若，表示常系数，曲线按照常量方式演化，方程表达式为：



(2-6)

常量演化机制的主要作用就是避免曲线出现合并或分裂的情况[]。综上所述，常量演化和曲率演化共同作用于曲线的拓扑结构变化。但是，研究学者发现按照这样的方式进行演化，会出现难以攻克的问题，例如，通过参数来引导曲线变化，其数值计算非常复杂；曲线在演化过程中出现分裂或合并，会使造成曲线的根本性改变，而我们难以预估曲线在演化过程中是否会出现这样的情况；曲线演变过程中误差的积累会影响精度。研究者们尝试改变思路应对这些问题，1988年，OSher和Sethian提出了水平集方法，成为了攻克这些问题的利器。

## 2.2水平集方法

### 2.2.1水平集理论主要思想

OSsher和Sethian[]两位学者在研究热力学方程时，为了能够描述火苗外部轮廓线飘忽不定的形态而提出了水平集方法。该方法的主要思想是利用高维泛函来描述低维曲线的演化，而低维曲线的表达式则映射到高维泛函的零水平集处，这样就将之前数值求参转变为求解PDE问题[]。这样的描述看似更加复杂，但在曲线演化时简化数值计算，解决了长久以来的问题。

我们将二维平面曲线映射到高一维的水平集函数中 ，给定时间得到，并且将时刻的零水平集定义为演化曲线，即，然后进行复合链式求导，得到：



(2-7)

在公式中，表示的梯度，进一步假设曲线的弧长参数，根据之前的讨论在曲线切线方向的变化量为 ，可得：



(2-8)

由上述公式可知，与切线 相垂直，那么与曲线法线的方向是相同的，进一步假设水平集函数在曲线内部小于0，在曲线外部大于0，可得内部单位法向量 ，根据 ，得到：



(2-9)



其中，



(2-10)

公式(2-9)是Hamilton-Jacobi偏微分方程，综上可知，水平集方法的核心是求解出随时间变化的偏微分方程。

### 2.2.2水平集函数的求解

上一小节分析可以看出，偏微分方程求极值是水平集函数求解的核心，那么它主要的求解方法有限差分法、有限元法、谱法等。而图像作为二维网格数据，具有离散化的特点，使得有限差分法成为了偏微分方程求极值的首选方法。

假设有一副二维图像，大小为，水平集函数为，假定空间间距，即步长为，时间间隔为，在时刻，图像在像素点处的水平集函数为 ，那么(2-9)公式可以表示为：



(2-11)



根据(2-11)可以看出，在时刻网格点的速度为。为了避免Hamilton-Jacobi方程出现不连续或者导数不存在的情况，我们采用Upwind Finite Differential Method进行求解[]。水平集函数演化方程如下所示：

(2-12)



其中，



(2-13)



(2-14)

同时，还需要得到中心差分算子、向前差分算子、向后差分算子的一阶公式：



(2-15)



(2-16)

(2-17)



为了保证曲线正常演化，确保水平集函数收敛，就需要令时间步长满足如下Courant-Friedrichs-Levy条件：

(2-18)



其中，为点的最大移动速度。

## 2.3经典的水平集活动轮廓模型

### 2.3.1 Mumford-Shah模型

Mumford和Shah两位学者最早提出基于区域信息的活动轮廓模型，简称M-S模型，其能量泛函表示如下：

(2-19)



其中，第一项作为驱动项，使分割曲线不断靠近目标分割区域，第二项是正则项，保证曲线在演化过程中拓扑结构的平滑，第三项为长度项，限定了轮廓线的长度。该公式通过三个能量项共同约束了分割轮廓线的演化，确保轮廓线的平滑完整准确，但是，该模型的数值求解十分复杂且计算量大，之后，Chan和Vese两位学者在此基础上进行了简化，提出了基于水平集的C-V模型。

### 2.3.2 Chan-Vese模型

M-S模型虽然足够完美，但是复杂的求解过程令众多学者转而研究更加精简的模型，其中最为经典的就是结合了水平集方法的C-V模型[]。

我们将图像定义在上，定义活动轮廓曲线为，待分割的图像由两部分组成目标区域和背景区域，而演化曲线也会将图像分成曲线内部区域和，将能量泛函定义如下：

(2-20)



其中，和均大于0，通常取为1，和分别表示曲线内部和外部的平均灰度值，当演化曲线到达目标区域分割边界时，趋近于0，因为此时曲线各个像素点的灰度值与接近，曲线外部的各个像素点的灰度值与接近，因此，在曲线到达目标区域分割线的时候，取到最小值。我们再加入长度项构成Chan-Vese所提出的能量泛函模型：

(2-21)



其中，，。通过最小化上述公式，可得到参数。CV模型的优势在于速度函数与图像的梯度互不影响，轮廓检测既可以针对梯度有意义的图像也可以检测梯度无意义的图像。

根据上述分析可知，零水平集用来表示分割曲线，将水平集函数用表示，同时引入Heaviside函数来划分演化区域：



(2-22)

Dirac函数来限制零水平集周围的函数值：



(2-23)

能量泛函公式可进一步推导如下：

(2-24)



(2-25)



(2-26)

水平集函数的C-V模型能量泛函表示为：



(2-27)

因此图像水平集函数可以表示为：

(2-28)



保持水平集函数不变，将能量函数最小化，可以得到两个参数的表达式如(2-29)、(2-30)所示。上述提到的Heaviside函数、Dirac函数是一个概念性的函数，因此需要引入向其逼近的，具有表达式的函数，如(2-31)、(2-32)

(2-29)



(2-30)



(2-31)



(2-32)

我们将轮廓线的曲率表示如下：



(2-33)

保持和的值不变，求水平集函数的最小值，得到如下求导方程：



(2-34)

通过以上详尽的分析，可以得到水平集方程的数值解(2-29)、(2-30)、(2-34)。CV模型的数值解都是依据图像的全局信息得到，可以看出CV模型的特点就是图像的全局性。

最后对水平集函数进行离散化处理，采用的方法是有限差分法，假设时间间隔为，空间间隔为，可以得到如下迭代公式：

(2-35)



上述公式中，，表示在坐标处，、次迭代水平集函数值，、表示向前差分、向后差分算子：



(2-36)

以上公式就是CV水平集函数的离散形式。

### 2.3.3 RSF模型

RSF(Region-scalable Fitting)模型是在CV模型上的进一步改进，是Li[]等人为了解决仅考虑全局信息产生的灰度不均匀问题，提出的局部区域拟合模型。该模型将高斯核函数引入能量项，并通过高斯函数与数据拟合项结合的方式，共同驱动曲线的演化，同时，水平集函数中的各类正则项确保了曲线拓扑结构的平滑。我们对该模型进行分析，定义图像，图像区域为，对其中的某个像素点定义为，曲线将图像分为曲线内部和曲线外部。定义RSF的局部拟合能量泛函公式如下：



(2-37)

公式中，和为非负常数，表示轮廓线内部区域灰度均值，表示轮廓线外部区域灰度均值，为周围像素点，表示高斯核函数,引入作为尺度参数，表达式如下：

(2-38)



上述泛函针对的是某个像素点周围的情况，泛函的大小通过，两者的差值计算出来。那么整幅图像的能量泛函需要将所有像素点的局部能量泛函叠加起来，得到总的能量泛函表达式如下：

(2-39)



其中，表示轮廓线的平滑项，然后，按照2.3.2小节CV水平集模型的推到步骤，将水平集函数引入能量泛函中：



(2-40)

其中，以及。选取Heaviside函数和Dirac函数如下：



(2-41)



(2-42)

加入正则项以及(2-39)，(2-40)后，可得能量函数：

(2-43)



(2-44)



(2-45)

在上述公式中，为，为。根据梯度下降流方法，以及欧拉公式，可得

(2-46)



根据上式，可以得到



(2-47)

接着，固定的数值，我们可以再求出水平集函数关于时间的偏微分方程：



(2-48)

其中，为公式(2-40)Dirac函数，其中，，表达式为：

(2-49)



综上所述，就可以利用轮廓线附近的局部图像信息，驱动曲线演化到达最终的目标区域的边界，同时保证了轮廓线拓扑结构的平滑完整。

## 2.4其它相关算法

### 2.4.1高斯混合模型

高斯分布(也称正态分布)是概率论中非常重要的一种分布类型，具有广泛的应用，单一变量的高斯分布如图2-2所示表示均值，表示方差，由于图像特征的特殊性，无法将所有的特征均用同一个高斯分布去描述，因此，采用混合的高斯分布模型能够更加精准的描述图像的特征分布。



图2-2 高斯分布示意图

图像中的每个像素抽取出来的特征向量都有不同的类别，隶属于不同的标签，例如，像素属于目标区域，像素属于背景区域，不同区域之间相互独立，区域内不同像素之间的特征向量也是相互独立，同一区域内的像素服从同一高斯分布，所有区域的高斯分布加权组合得到该图像的整体高斯分布。

高斯混合模型图像分割的原理如下：图像的各个区域可用特征向量进行表示，例如图像纹理、像素点的灰度值等等，这些信息都能反映出图像中各个区域间的差异。假设图像某个区域为，那么它就隶属于高斯混合模型中第个高斯分量，当我们计算某个像素属于各个不同区域的概率，即可得到这些像素的分类，从而达到图像分割的目的。



图2.3高斯混合模型

如图2.3所示，第个像素的观测值用表示，个标记用表示，对应的概率密度为，则它的函数为[]：



(2-50)

像素集的模型是有限混合模型，为某像素属于标记的先验概率，，表示所有的参数，先验概率满足条件：



(2-51)

所有的子分布服从高斯分布，它所对应的就是高斯混合模型：



(2-52)

像素统计之间具有独立性，其中，像素集构成的联合条件密度为[]：

(2-53)



上述公式是在参数似然函数，对(2-51)取对数可以得到对数似然函数：



(2-54)

这里。对上述公式通过EM算法[]进行最大化似然估计，得到各个参数的值。高斯混合模型形式简单，参数少，计算复杂度也相对较小，本文将其作为轮廓线的初始化工具，有效提高了分割精确度。

### 2.4.2 KMeans聚类算法

KMeans聚类算法属于无监督聚类算法，该算法依据某一标准，再学习过程中按照数据的特性，将数据集分为不同类或者簇[]，该算法可用于图像分割。基本原理如下：

将数据集，划分为个分区，，每个分区表示一个类。每一个样本对象对应一个分区，每一个分区最少有一个样本，KMeans算法的执行流程如下：

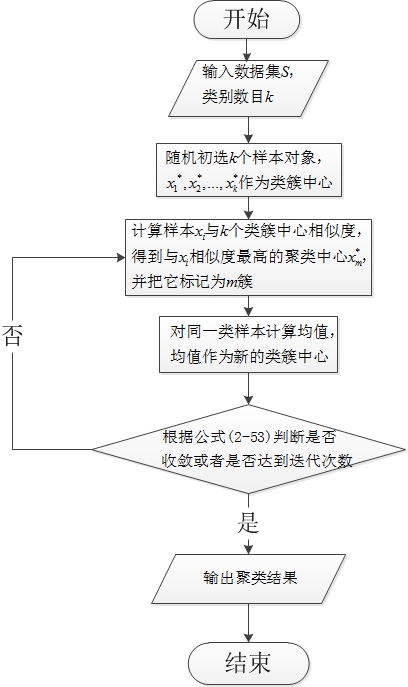


图2.4 KMeans算法流程图

经过上述算法流程后，我们可以得到最终的分类结果，以及对应的类簇中心，聚类结果评价以及迭代终止条件如公式(2-53)，如果聚类效果比较好，那么较小。



(2-55)

### 2.4.3 LBP算法

LBP(Local Binary Patterns)[]能够有效提取图像的特征，算法的主要思想是：在二维图像中，存在一个矩阵，将矩阵中心的灰度值作为阈值，邻域的八个像素点的灰度值与其进行比较，如果灰度值比阈值大，则标价为1，否则标记为2。如下图所示。



图2.5 LBP算法示意图

公式表示如下：

(2-56)



其中，表示矩阵的中心，该像素点的灰度值为，它周围像素点的灰度值为，我们定义符号函数如下：

(2-57)



本文采用了LBP算法的思想，充分考虑轮廓曲线邻域的灰度值分布，对能量项动态调参，在模型中将全局信息和局部信息进行了有效的结合。

## 2.5本章小结

本章主要介绍了曲线演化的理论以及曲线演化存在的参数化问题，详细介绍了水平集方法的主要思想和水平集函数的数值求解过程，总结了结合水平集方法的三个经典活动轮廓模型：Mumford-Shah模型、Chan-Vese模型、RSF模型，最后介绍了与本文模型相关的其它算法和模型：高斯混合模型、KMeans算法、LBP算法。

第3章 基于高斯混合模型与CV水平集理论的混合模型

根据2.2小节、2.3小节介绍的水平集理论及相关算法可知，基于水平集的图像分割方法，需要在算法的初始阶段确定一条轮廓线，该轮廓线在算法迭代过程中不断演化，最终演化至目标区域边界达到分割的目的。现如今不论是传统的CV水平集图像分割算法还是其它改进算法，在确定初始轮廓线时没有统一的方式和标准，通过下图3.1可以看到，不同的初始轮廓线会造成不同的分割结果，甚至有些会造成图像分割失败，因此，综合上一章节的理论知识以及实践总结，我们可以看到初始轮廓曲线的重要性。



图3.1不同初始轮廓线下的分割结果对比：(a)不同初始轮廓线

(b)传统CV模型的分割结果(c)LBF模型的分割结果

为了能够克服初始轮廓线不确定性所带来的问题，本文采用高斯混合模型对图像的目标区域进行预判，确定初始轮廓线。高斯混合模型属于无监督聚类模型，可有效应用于图像中像素的分类，而且其对各种形状分布的数据都具有良好的聚类效果，因此可适用于划分各种形状的目标区域。又因为水平集方法具有良好的信息包容性，我们将高斯混合模型确定的先验信息融合进CV水平集模型中，并引入先验信息中的图像灰度均值，共同作用于CV水平集图像分割模型，使实验结果更加精准。

## 3.1 GMMCV模型框架介绍



图3.2 高斯混合模型确定目标区域

根据2.4.1小节以及第三章介绍的高斯混合模型的优势，本文将CV水平集模型与高斯混合模型相结合，提出了GMMCV模型，从图3.2中我们也可以看到高斯混合模型能够很好的确定初始轮廓线，有效规避了不同轮廓线造成分割结果的差异，同时我们可以看到背景区域的图像纹理已经被消除，完全达到了灰度均匀，这更加有助于提高曲线演化的准确度，该模型的框架图如图3.3所示。

1. 模型输入：将待处理的图像输入模型中。

2. 高斯混合模型确定初始轮廓线：利用高斯混合模型进行图像像素的二分类，即确定图像的目标区域以及背景区域，区域的分界处为初始轮廓线。

3. CV水平集方法演化曲线：计算目标区域与背景区域的灰度均值，将两者信息融入到传统CV水平集模型进行改进，两个区域的平均灰度值也随着能量泛函共同迭代，达到更加精准的分割效果。

4. 模型输出：将图像分割最终结果输出。

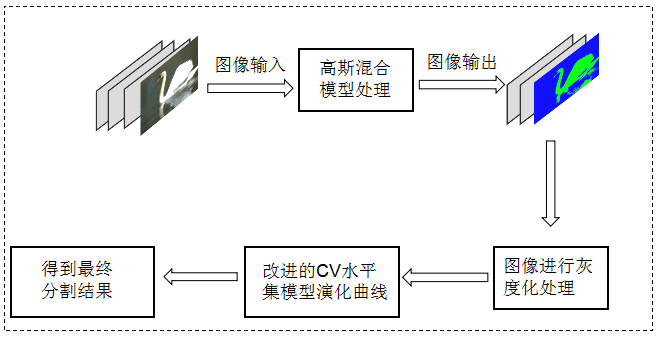


图3.3 GMMCV模型框架图

## 3.2改进的CV水平集模型

为了能够达到更加精准的分割效果，本文不仅采用高斯混合模型确定的初始轮廓线，同时，充分利用了目标区域与背景区域的灰度均值，改进了传统的CV水平集模型，使最终的分割结果更加精准。

### 3.2.1结合区域平均灰度值的CV模型

高斯混合模型对图像像素聚类后的效果，如图3.2所示，得到该结果后，我们对初始的目标区域以及背景区域计算灰度均值和，公式如下：



(3-1)

其中， 表示目标区域内部像素点的灰度值总和，表示目标区域内部像素值的个数， 表示背景区域内部像素点的灰度值总和，表示背景区域像素点的个数。之后，我们将这两个数值融入到CV模型中得到新的能量泛函公式：



(3-2)

上述公式中，表示水平集函数，和分别表示曲线内部和外部的平均灰度值，表示像素点在坐标的灰度值，为了能够平衡目标区域、背景区域能量项和曲线轮廓内部、外部区域能量项在函数中的权重，我们加入了权重参数，两个参数的数值均为0.5，公式中均取值为1，表示Heaviside函数，定义如下：



(3-3)

因为Heaviside函数特殊性，没有具体的函数表达式，引入水平集函数后，选取Heaviside函数表达式如下：



(3-4)

### 3.2.2能量泛函求解

上一小节已经介绍了加入先验信息后优化的CV水平集模型，为了保证曲线在演化过程中的稳定，防止水平集函数在曲线演化过程中与符号距离函数偏差越来越大，我们在能量泛函公式加入正则项：



(3-5)

同时，为了保证曲线在演化过程中拓扑结构的平滑自然加入长度惩罚项：



(3-6)

得到最终的能量泛函公式为：

(3-7)



通过变分法以及标准梯度下降流法，可进一步得到水平集函数对时间的求导函数：



(3-8)

其中， 为Dirac函数[]，公式如下：



(3-9)



因为图像是离散化网格类型的数据，为了能够准确对水平集函数进行数值求解，我们采用2.3.2小节介绍的有限差分法进行计算，算法流程如表3-1所示。

表3-1 GMMCV模型算法流程

|  |
| --- |
| GMMCV模型算法流程 |
| **输入：**  输入图像  **过程：**  # # # # # # # # # # # # # # # #高斯混合模型确定初始轮廓线# # # # # # # # # # # # # # #  1：初始化分类数目k以及各个分模型参数  2：EM算法迭代计算，求得各个参数直至收敛  3：对像素点标记颜色，得到聚类结果  # # # # # # # # # # # # # # # #改进CV水平集模型演化轮廓线# # # # # # # # # # # #  4：设定初始参数值  5：根据高斯混合模型进行水平集函数初始化  6：依据(3-1)计算和的值  7：依据水平集演化的偏微分方程，更新水平集函数  8：判断是否满足收敛条件，取0.01，否则返回步骤6  **输出：**割后图像 |

## 3.3实验

### 3.3.1实验环境和实验数据

实验硬件环境为:Intel(R) Core(TM) CPU I5-3370 @1.80GHz(8核) CPU、12GB RAM;软件环境为：Windows 7旗舰版操作系统、Matlab 2014a 、Opencv 3.1.0。实验参数若无特殊说明选取如下：Heaviside函数和Dirac函数中为1.0，时间步长取0.01，取0.002\*255\*255，取1.0。

实验数据：采用Berkely Computer Vision Group提供的数据集BSDS500[]，该数据集主要用于图像分割或者轮廓检测方面的研究，包含200张训练图片，200张测试图片，100张验证图片，数据集目标区域的结果由人工进行标注，包括轮廓信息和分割信息，数据集中的图像类型属于自然场景图像。

### 3.3.2评价标准

为了能够展示本文算法的有效性，我们采取以下四种量化指标对实验结果进行评测：



(3-10)



(3-11)



(3-12)

上述公式中，表示图像的真实区域值，表示图像的实验结果，数值越大说明模型分割的效果越好，数值越小说明模型分割的效果越好，则表示分割出的区域与真实区域的相似度。除此之外，本文还采用RMSE(Root Mean Square Error)来验证实验结果，公式如下所示，模型结果像素点的坐标用表示，真实区域坐标用表示，RMSE数值越小，表明实验结果与真是结果偏差越小。

### 3.3.3实验内容

为了验证本文GMMCV模型的有效性，我们在实验中对比了经典的CV水平集模型、引入了图像局部信息的LBF模型和RSF模型、简化计算复杂度的LIF模型、针对灰度不均匀问题的TPFP模型[]。

实验一：在公开数据集BSDS500上进行测试，将本文模型与上述其它模型进行对比，实验结果通过3.3.2小节介绍的评价标准进行展示。

实验二：在灰度不均匀的图像进行实验，选取其它模型最优的初始轮廓曲线所得实验结果与本文GMMCV模型进行对比，通过直观的分割结果，可以看出实验效果的好坏。

### 3.3.4实验结果





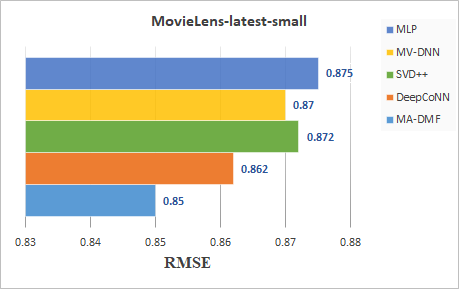








实验中，由于模型学习时增加了批处理操作，我们可以选择一个更小的正则化参数和更大的学习率，采用Mini-BGD方法对损失函数进行优化。模型的相关参数设置如下，模型能取得较好的推荐性能： batch-size为512、隐因子个数为64、学习率为0.1、隐藏层网络结构为[128,64,32,16]、L2正则化参数为0.00003、嵌入层和深度神经网络层中分别采用均值为0方差为[0.01/n2]和[1/n2]的高斯分布去初始化权重参数(n为嵌入层向量维度和隐藏层神经元个数)、用小批次梯度下降的方法对损失函数进行优化。采用RMSE指标与其他推荐模型对比实验，实验结果如图3-2所示。结果表明，本文所提MA-DMF模型在MovieLens latest的small和20m两个数据集上均能更准确地进行评分预测，相比上述的其他模型具有更好的推荐性能。



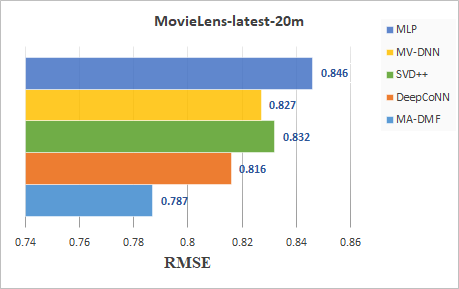


图3-2 推荐模型RMSE对比

### 3.3.5参数分析

在基于深度学习的推荐模型中，模型相关的参数会在一定程度上影响整体模型的推荐性能，为了进一步探索MA-DMF模型的相关参数对整体推荐性能的影响，我们将从隐藏层结构（网络层数）、隐因子个数、批标准化操作(Batch Normalization，BN)和正则化系数（λ）几个方面展开研究讨论，在MovieLen-latest-small数据集上进行对比实验。

在基于深度学习的模型中，隐藏层的神经网络层数是非常重要的一个参数，也是深度神经网络中的一个关键部分。在提出的MA-DMF模型中，我们认为隐藏层网络中的每一层都拥有不同的学习能力，所以每一层所学习到的特征表示都被认为是重要的。针对以上考虑，我们尝试去证明深度神经网络的网络层结构对模型存在一定影响，通过实验对比进一步探索了隐藏层中神经网络层数对模型性能的影响，实验结果如图3-3所示。Layer-5指的是模型隐藏层的网络层数为5。MA-DMF模型的其他超参数设置为：隐因子个数为64，L2正则化参数为0.00003，学习率为0.1，batch-size为512进行对比实验。通过实验对比，可以观察到：本模型的隐藏层的层数小于等于4之前，通过增加其网络层数目可以在一定程度上提升模型的推荐性能；但是，当隐藏层的网络层数目大于等于5之后，模型的推荐性能反而会降低。这表明一定数目的隐藏层层数能够让模型具有更优的推荐性能，但更深层次的网络反而会浪费大量的计算资源，导致一个更差的推荐效果。

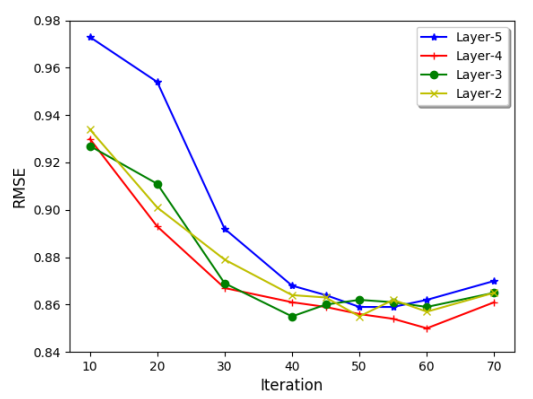


图3-3网络层数对比实验

在前文中提到过，我们需要将用户和项目的向量映射到一个隐藏空间中，获得用户和项目的一个隐特征表示，再将其送入深度神经网络中去学习更深层次的特征表示，所以隐因子的个数对整个模型结构也有重要的影响。为了选择一个合适的隐因子个数，针对不同的隐因子个数[16,32,64,128]进行了对比实验，实验结果如图3-4所示。我们设置隐藏层的层数为4、L2正则化参数为0.00003、学习率为0.1、batch-size为512。通过实验对比，可以观察到：当设置模型的隐因子个数为64时，能获得一个较好的推荐结果。这也表明适当增加隐因子个数能够提升模型的推荐性能；但是，当模型隐因子个数超过一定数量，不仅需要占用大量的内存和计算资源，还会导致一个更差的推荐效果。

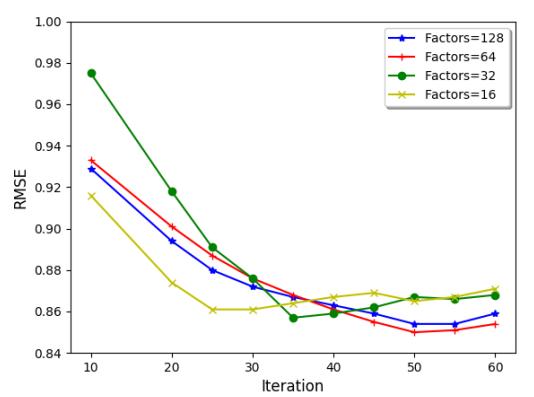


图3-4 隐因子个数对比实验

为了验证批处理（BN）的有效性，设置因子个数为64，隐层结构为[128,64,32,16]、L2正则化参数为0.00003、学习率为0.1、batch-size为512，进行了对比实验，结果如图3-5所示。通过实验对比，可以观察到：在MA-DMF模型中添加批处理操作能够有效地对深度神经网络层的输入进行归一化，保证输入数据分布的稳定性，能在一定程度上提升模型的推荐性能。

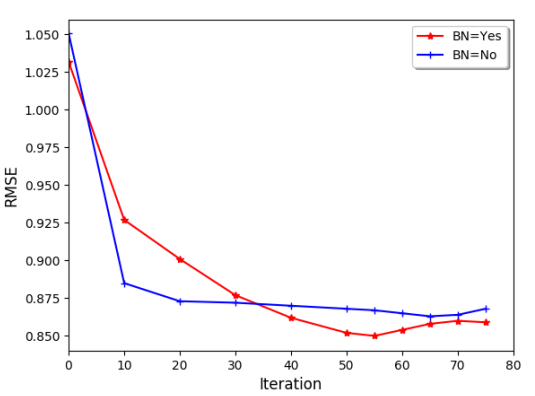


图3-5 批标准化(BN)作用对比实验

在对模型进行优化时，我们针对评分预测的任务构建了相应的损失函数，并增加了L2正则化操作去防止模型过度拟合。为了选择一个合适的正则化系数，探索不同正则化系数对模型的影响效果，我们通过设置不同的正则化系数去优化模型的损失函数，其对比实验结果如图3-6所示。设置因子个数为64、batch-size为512、学习率为0.1、隐藏层的网络结构为[128,64,32,16]。通过实验对比，可以观察到：我们设置正则化系数λ=0.00003时，预测结果与实际评分相差最小，模型表现的推荐效果最好。这也表明正则化系数能一定程度上影响模型的性能，而且在采取BN操作之后，可以选择一个更小的正则化系数去训练模型。

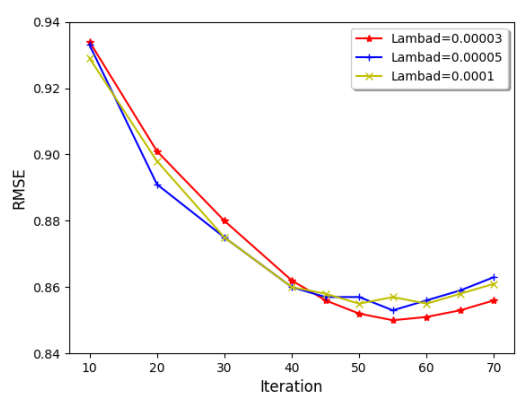


图3-6 正则化参数λ对比试验

## 3.4本章小结

通过分析，大多数深度学习推荐模型存在数据稀疏性问题，而且只利用用户和项目最后的隐表示进行交互。我们进一步融合辅助信息并重新构建隐层网络，提出了融合辅助信息的多交互深度矩阵分解模型（MA-DMF），对该模型的整体结构和学习过程进行了介绍说明。在MovieLen latest数据集与其他推荐模型进行相关实验对比，在RMSE指标上的结果证明所提出的MA-DMF模型能更准确地进行评分预测。最后针对模型相关参数进了分析，通过实验去验证并说明了相关参数对模型整体推荐性能的影响。

# 第4章 融合注意力机制的多交互神经矩阵分解模型

自从注意力机制被引入神经网络的建模中，已经在许多领域中被广泛应用，例如：机器翻译[58]，信息检索[59]和计算机视觉[60]等。目前大多基于深度学习的推荐模型对深度神经网络中学习到的用户/项目隐特征表示都做了统一对待，对所有可能的特征交互都使用相同的权值，但并非所有特征都是相关的，在不同特征之间进行交互时，其重要程度是不一样的；而且大多数模型只利用了深度神经网络的非线性学习特性，没有过多地尝试将线性的浅层推荐模型与非线性的深层推荐模型相结合。

针对以上问题，受注意力机制和混合推荐模型启发，提出一种融合注意力机制的多交互神经矩阵分解模型 (Attention Multi-interactive Neural Matrix Factorization Model, AM-NMF)基于上文提出的多层交互网络结构，进一步融合注意力网络获得关于用户和项目特征交互的权重矩阵，用来判别不同特征交互的重要程度；将传统的浅层推荐模型和融合注意力的深层推荐模型相结合，提升整体模型的推荐性能。

## 4.1模型介绍

融合注意力机制的多交互神经矩阵分解模型（AM-NMF）结构如图4-1所示。以下将分别介绍模型的输入、浅层/深层推荐结构、注意力网络和模型输出结果。

1. 模型输入：考虑隐式反馈信息，通过One-hot编码用户和项目的ID信息作为模型的输入数据。

2. 浅层/深层推荐结构：将整体模型理解成一个浅层矩阵分解推荐模型和一个深层注意力多交互推荐模型的结合，它们共享同一个嵌入层向量，获得不同的推荐结果。

3. 注意力网络：将多层神经网络的输出的用户和项目的隐表示作为输入向量，通过一个标准的注意力网络去获得一个注意力权重矩阵，与深层多交互网络学习到的隐表示进行加权，获得一个深层模型的推荐结果。

4. 模型输入：将得到的浅层模型推荐结果和深层模型推荐结果加权作为AM-NMF模型最终的推荐结果。

### 4.1.1模型输入



图4-1 融合注意力机制的多交互神经矩阵分解模型（AM-NMF）

模型最底部为输入层，我们使用一个用户和一个项目作为输入特征，通过One-hot编码将其转换为二进制稀疏向量。这里的输入同样可以使用相关内容特征来表示用户和项目，也能一定程度解决冷启动的问题。

在AM-NMF模型中，我们考虑隐式反馈去构建用户-项目交互矩阵，假设M和N分别表示用户和项目的数量，则其交互矩阵Y∈RM×N定义为：

 (4-1)

其中可以理解为一个标签，1表示用户u和项目v相关（用户u对项目v存在评分），否则为0。但是1并不意味用户u真的对项目v感兴趣，也不能表明用户u不喜欢v，可能是用户根本不知道这个项目的存在或者丢失了一部分数据。

所以，隐式反馈中的推荐问题可以被认为是去估算矩阵**Y**中未观察条目的评分（所获得的评分用来评估项目的排名），推荐模型假定这些缺失的数据可以通过底层模型来生产，形式上可以被抽象为学习函数，其中表示用户与项目交互的预测分数，表示模型参数，f表示将模型参数映射为预测评分的交互函数。下面，我们将主要介绍AM-NMF模型的主要结构，以及如何学习并参数化交互函数f从而估算。

### 4.1.2浅层/深层推荐结构

整个AM-NMF模型可以看作是一个传统浅层推荐模型和一个融合注意力机制的多交互深层模型。

在输入层之上为嵌入层，通过一个完全连接层将输入层中稀疏向量中的非 0 特征映射为一个稠密的向量表示，帮助去学习并获得用户与项目的嵌入向量pu，qv（类似隐因子模型中的隐因子向量）。在本模型中，其中的浅层模型和深层模型共享同一个嵌入层，利用相同的用户和项目隐因子向量去学习和推荐。

传统的浅层推荐模型：对用户和项目嵌入后的隐因子向量直接采用点积操作，获得一个浅层模型的推荐结果。

 (4-2)

深层的推荐模型：将用户和项目的嵌入向量作为输入向量送入多层交互神经网络中去学习更深层次、更抽象的特征表示。这里我们没有使用通用的深度神经网络结构，我们认为隐藏层中的每一层所学习到的用户-项目隐表示都是非常重要的，为了避免漏掉重要的特征表示，我们对采用上文提出的多交互神经网络，对每一层所学习到的隐表示做一个交互操作，最后聚合所有层的交互结果作为一个输出结果。

 (4-3)

 (4-4)

**......**

 (4-5)

 (4-6)

 (4-7)

 (4-8)

其中，分别表示多交互神经网络中第i层的用户和项目隐特征表示；，分别表示网络中第i层的权值矩阵和偏置矩阵；表示第i层的激活函数，⊙表示向量对应元素相乘；表示第i层用户与项目的交互结果，表示聚合后的输出结果。在神经网络的激活函数选择中，我们使用 Sigmod、Tanh 、 ReLU 三种激活函数进行了对比实验，结果表明选择Tanh 函数模型能表现出更好的推荐性能。

 (4-9)

 (4-10)

 (4-11)

### 4.1.3注意力网络

在深层的多交互的网络结构之上为一个标准的注意力网络，通过多交互神经网络的最后一层所获得的用户和项目隐表示输出为pun，qvn，将其作为注意力网络的输入向量。注意力交互也是该模型的核心部分，通过注意力网络去学习一个注意力向量，发现用户和项目交互时不同特征的重要程度和影响力是不一样的。假设最后一层的神经元个数为K，则用户和项目最后输出的隐向量维度和注意力向量的维度都为K。注意力向量计算如下：

 (4-12)

 (4-13)

将多交互神经网络的聚合结果与注意力向量进行点积操作，得到一个深层模型的推荐结果：

 (4-14)

其中表示用户u对项目v的注意力向量。假设注意力向量为K维，，则；表示F中第k个因子。这也表示在pu和qv之间的每个隐特征因子进行交互时，都有一个注意力权重去发现项目v的这k个因子对用户u的重要程度。因此，注意力权重对每一个用户-项目交互都是独一无二的。

### 4.1.4模型输出

最后，将传统浅层模型的推荐结果和融合注意力机制的深层推荐结果进行加权操作，通过输出层的非线性函数将其映射为模型最终的输出结果：

 (4-15)

其中F表示输出层的激活函数；wi表示一个权重，判断两个模型表示结果的重要程度。此处我们采用Sigmod作为激活函数，设置w1和w2为0.5。

## 4.2学习过程

本节将对AM-NMF模型的学习过程进行介绍，分别从目标函数选择、模型训练和模型优化等几个方面进行说明。

### 4.2.1目标函数

在深度学习的模型学习中，可以通过常用的两种损失函数：逐点损失(point-wise lose)和成对损失(pair-wise loss)去优化相关的目标函数，然后去估计相关的模型参数。在上文的MA-DMF模型中，我们从显式反馈出发：即考虑逐点损失，最小化预测分数和真实分数之间的均方误差(square loss)。但是在AM-NMF模型中，考虑到隐式反馈的性质，我们这里可以把推荐看作一个二分类任务，即：将AM-NMF模型视为一个概率模型，使用二进制交叉熵损失函数(对数损失)[26]进行优化。这里我们采用成对损失：已观察到的条目应该比未观察到的条目排名更高，所以需要最大化已观察项和未观察条目之间的间隔。我们定义AM-NMF模型的目标函数如下：



 (4-16)

公式中y表示交互矩阵Y中观察到的条目（例如：用户对电影有明确的评分），则表示负实例（可以将未观察到的所有用户-项目交互样本都作为负实例，也可以采取抽样的方式标记一部分交互样本作为负实例）。表示已观察到的真实数据，表示未观察到的模型预测值。

### 4.2.2模型训练

在优化目标函数时，我们在Tensorflow框架下，采用Adam方法去训练神经网络模型。Adam算法可以自适应地去调整每个参数的学习速率（频繁的参数执行较小的更新，不频繁的参数执行较大的更新）。与SGD算法相比，使用Adam方法能让模型有更快的收敛速度，并减轻了调整学习速率的痛苦。

在模型训练时，我们考虑使用小批次迭代去加速整个模型的训练过程，采用一定数量的负采样（negative sampling）去提高训练速度。在模型的每一次训练中，我们都会抽取一定数量用户以前从未交互过的项目作为负实例，将负实例与正实例一起输入到模型中训练并进行参数更新。

### 4.2.3模型优化

在优化AM-NMF模型时，我们同样在深层结构对应的隐藏层中增加批标准化的操作，在模型进行迭代训练时对每一层网络的输入数据都进行归一化处理，保证输入数据分布的稳定性。

另外，为了防止模型过度拟合，我们在深度神经网络训练时增加Dropout操作[61]。设置p=0.5去随机“删除”网络中的一些神经元，即让这些神经元在训练时以0.5的概率去停止激活，不会过度依赖一些局部的特征，增加整体模型的泛化能力。

## 4.3实验

### 4.3.1实验环境和实验数据

实验硬件环境为:Intel(R) Xeon(R) CPU E7-4820 @2.00GHz(8核) CPU、64GB RAM的刀片服务器;

实验软件环境为：CentOS Release 6.6操作系统、Pycharm2017 、Tensorflow 1.3.0 CPU;

实验数据：采用最近经常用于隐式反馈的推荐数据集：MovieLens 1m数据集和Pinterest数据集。MovieLens 1M[53]是一组由MovieLens用户提供的电影评分数据。这些数据中包括6040名用户对3706部电影的近1000000条评分，每个用户至少有20条评分，另外包含一些电影的属性数据（电影年代、类型）和用户相关的属性数据（性别、职业、年龄）。Pinterest[54]是美国图片社交网站Pinterest中用户对喜欢的图片进行pin操作的数据，是由X.Geng等人在2015年抓取百万用户个人资料及其相关的图片，构建为一个用于推荐系统的隐式反馈数据，用来评估基于内容的图片推荐。2017年何向南等人[26]考虑到其数据过于庞大和稀疏，经过处理，只保留至少有20次交互（pin操作）的用户，得到了新的Pinterest数据用于评估基于深度学习推荐模型，其包含了55187名用户对9916个图片的近1500000条交互信息。

表4-1 实验数据集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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% |

由于在模型训练中采用了负反馈，我们对上面的两个数据集都进行了预处理，将数据集中所有的用户没有交互的项目找出并构建负反馈实例。在模型训练时，每一个交互都会与多个负实例进行配对，因此训练实例数量是大于交互次数的。

### 4.3.2评价标准

为了评估模型项目推荐的性能，在本文中采用leave-one-out方法进行评估，即：在整个数据集中，对于每个用户，我们将其最近一次的交互记录作为测试集，并利用余下的数据集作为训练集。考虑到在评估过程中为每个用户排列所有项目会花费太多时间，所以遵循一般的策略，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。排序列表的性能由**命中率（HR**@K**）**和**归一化折损累积增益（NDCG**@K**）**来衡量，这也是在排序推荐任务中被广泛应用的评价指标[26,31,55]。

在Top\_N排序的推荐任务中，通常是替用户找到他们可能感兴趣的项目，然后将用户可能喜欢的项目生成一个项目列表进行推荐，主要使用点击率（HR）和[归一化折损累积增益](https://en.wikipedia.org/wiki/Discounted_cumulative_gain)（NDCG）作为推荐模型的评价指标。

点击率（HR）是一个常见的用来衡量召回率的指标。HR@N表示推荐模型为用户推荐的前N个项目中，有多少是用户真实喜欢的项目。该指标的数值越大，表示推荐的项目列表越好，其计算公式为：

 (4-17)

公式中，GT表示整个测试集合，分子则表示替每个用户生成的个性化Top-N推荐列表中属于测试集合的总数。

[归一化折损累积增益](https://en.wikipedia.org/wiki/Discounted_cumulative_gain)NDCG同样是一种衡量推荐列表排序优劣的评价指标，但是该指标进一步考虑到列表中所有项目的相关性。下面将详细介绍该评价指标。

当推荐模型替用户生成一个项目列表，想要去判断这个列表的好坏则需要考虑到列表中每个项目的排名顺序。列表中的每一个项目都被给定一个相关的非负数分值，也就是所谓的增益(G)。对于那些没有用户交互的项目，其增益值为0。再将推荐列表中所有项的相关性分值累加，所得到的分值作为这个推荐列表的一个得分，也就是所谓的累积增益(CG)。由于CG依然没有考虑列表中的项目处于不同排序位置对整个推荐列表的影响，用户们更希望去看到相关性更高的项目排在列表的更前边。在累积增益的基础上进一步考虑位置的影响因素（DCG）。先将每项都除以一个数（通常是该项排序位置的对数），也就是折损值，然后对这些分值进行累加得到DCG。

 (4-18)

 (4-19)

上述公式中，reli表示列表中排序第 i位的项目的相关性，k 则表示整个推荐列表的大小。公式2-31中，分子的值越大，表示第i个项目的相关性越大，推荐的效果越好。分母中的i表示项目在推荐列表中的排序位置，i越大表示该项目在推荐列表中排序越靠后，推荐效果越差。

一般在Top\_N的推荐任务中，通常将测试集中所有的项目在进行合适的排序，然后取它的前K项作为推荐列表并计算DCG@K。IDCG表示最理想情况下的DCG，即DCG最大的时候。

 (4-20)

最后将之前的DCG值除以IDCG的值得到NDCG@K，该结果的取值范围在0到1之间。NDCG@K的值越大，表示推荐列表的排序的质量越好。对于用户u的推荐列表，其NDCG@K值计算为：

 (4-20)

在实验中，设定这两个指标的排名列表大小为10，即：K=10，可以通过HR直观地去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。对比实验中为每个测试用户都计算这两个指标并取其平均值。

### 4.3.3实验内容

为了验证本文所提AM-NMF模型的推荐性能，我们同样基于Tensorflow框架与2001年提出的经典的ItemKNN[35]算法、2016提出的 eALS[37]算法、2017年X. He等人基于Wide&Deep[25]提出的MLP模型、2017年X. He 等人进一步提出的NeuMF[26]模型等算法进行了对比实验。

**ItemKNN:**这是一个经典的推荐算法，我们还原了标准的ItemKNN算法思想，并且结合2008年Y. Hu[47]等人提出的基于隐反馈数据的协同过滤算法（WMF），让其适用于隐式反馈数据。

**eALS:**这是H. Zhang等人在2016提出的一个推荐算法，其结果表现优于传统的矩阵分解(MF)算法和标准化权重的WMF[47]算法。eALS算法将所有的未观察交互都视作负实例并且根据项目流行度去进行非统一性的加权计算，通过优化均方误差去提升推荐性能。

**MLP:**这是一个在用户和项目的嵌入层之上应用多层神经网络去构建推荐模型的算法，具体算法思想见上文2.3.1所述。2017年X. He应用4层网络的MLP算法在两个数据集上进行对比实验，取得了较好的推荐效果。

**NeuMF:**这是X. He等人在2017提出的一个基于神经网络的混合推荐模型，算法具体思想如4.1所述。该算法使用隐式反馈并且随机去初始化用户和项目的表示，通过交叉熵损失函数[26,]优化推荐模型。由于本文算法没有考虑预训练，我们对比的算法也没有进行预训练操作。

实验内容主要包括以下两部分：

(1)：在公开数据集上将所提AM-NMF模型与上述的4个推荐模型进行对比实验，通过HR和NDCG评价指标进行比较，论证所提推荐模型的有效性。

(2)：对所提AM-NMF模型的相关参数进行比较分析，论证不同超参数和相关操作对模型推荐性能的影响。

### 4.3.4对比实验

在模型学习时，我们增加了Dropout和批标准化(BN)操作，帮助去避免过拟合并对学习过程中的向量表示进行归一化，采用Adam方法对损失函数进行优化。模型的相关参数设置如下，模型能取得较好的推荐性能：学习率为0.001、batch-size为256、Dropout概率为0.5、隐藏层结构为[128,64,32,16,8]，注意力网络采用均值为0.02的高斯分布去初始化注意力向量表示。在MovieLens数据集中，设置隐因子个数为128、训练负实例个数为9、嵌入层中采用均值为0，标准偏差为0.01的高斯分布去随机初始化用户和项目的向量表示，模型能获得较好的推荐结果。在Pinterest数据集中，设置隐因子个数为64、训练负实例个数为8、嵌入层中采用均值为0，标准偏差为0.02的高斯分布去初始化用户和项目的向量表示，模型能获得较好的推荐结果。通过HR@10和NDCG@10评价指标进行比较，本文所提AM-NMF模型较其他算法模型有所提升。实验结果如表4-2和图4-2，4-3所示：

表4-2 推荐模型HR@10和NDCG@10对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Datasets** | **Metrics** | **Methods** | | | | |
| **ItemKNN** | **eALS** | **MLP** | **NeuMF** | **AM-NMF** |
| ML 1m | **HR@10** | 0.622 | 0.698 | 0.701 | 0.711 | **0.727** |
| **NDCG@10** | 0.358 | 0.426 | 0.423 | 0.432 | **0.451** |
| Pinterest | **HR@10** | 0.786 | 0.867 | 0.866 | 0.873 | **0.882** |
| **NDCG@10** | 0.485 | 0.538 | 0.545 | 0.551 | **0.568** |

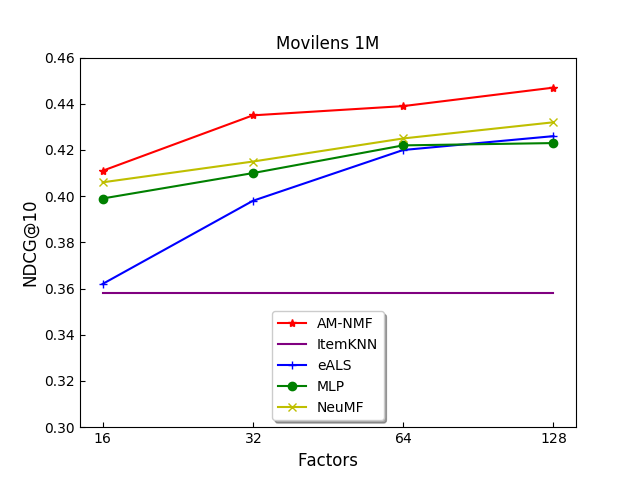
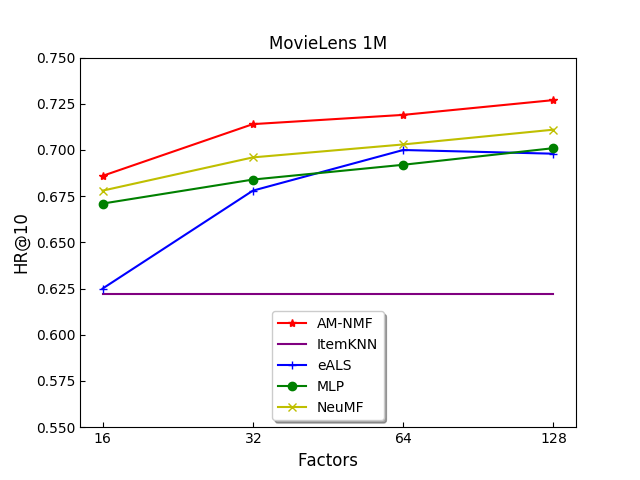


图4-2 MovieLens数据集，各模型在不同隐因子下HR@10和NDCG@10对比

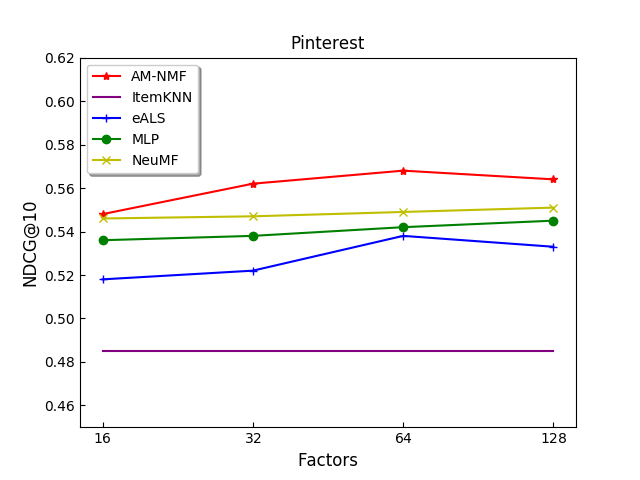
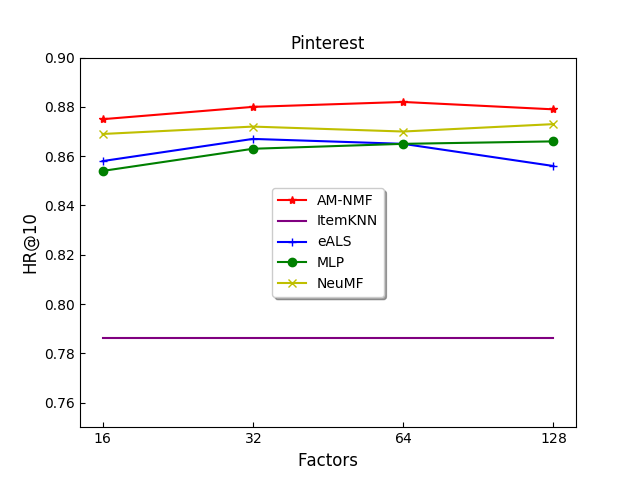
****

图4-3 Pinterest数据集，各模型在不同隐因子下HR@10和NDCG@10对比

### 4.3.5参数分析

在这一部分，同上文中的参数分析类似，我们继续去对比所提出的AM-NMF模型中相关超参数对模型推荐系统的影响。考虑模型中的隐因子个数、隐藏层结构（网络层数）和训练模型的负实例个数这几个方面，通过在MovieLen 1M数据集上进行对比实验，进一步探索相关参数对模型性能的影响。模型中的其他设置如下：

为了验证不同隐因子个数对推荐模型的影响，设置学习率为0.001、batch-size为256、隐藏层结构为[128,64,32,16,8]、负实例个数为9、嵌入层中采用均值为0，标准偏差为0.01的高斯分布去随机初始化用户和项目的向量表示，采用Adam算法去优化模型。分别对比不同的隐因子个数[16,32,64,128,256]进行实验，结果如图4-4所示。结果表明隐因子个数为128或256时模型能取得较好的推荐性能，但更多的隐因子个数不但没有明显地提升推荐性能，反而会消耗更多的计算时间。

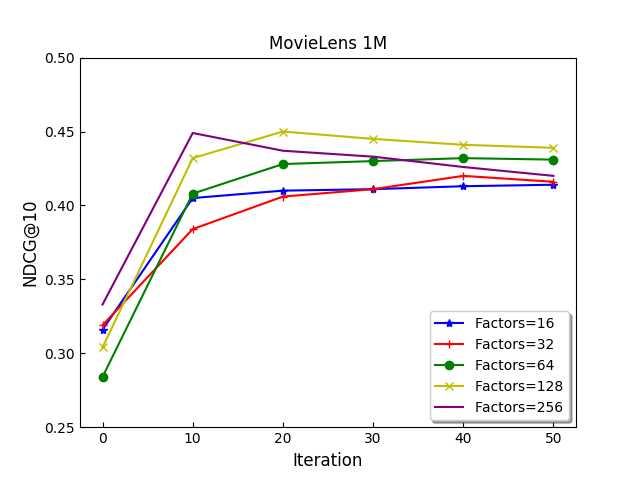
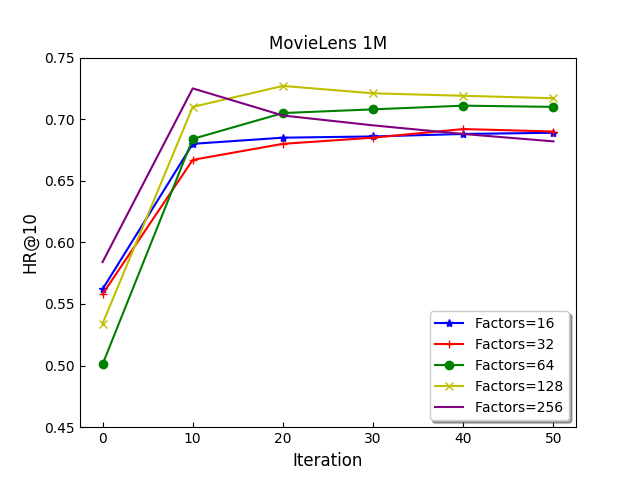


图4-4 隐因子个数对比实验

同样的，设置学习率为0.001、batch-size为256、隐因子个数为128、负实例个数为9，嵌入层中采用均值为0、标准偏差为0.01的高斯分布去随机初始化用户和项目的向量表示，采用Adam算法去优化模型。分别对比隐藏层中不同网络层数对推荐模型性能的影响，实验结果如图4-5所示。当隐藏层数为4层或5层时模型能取得较好的推荐效果，这也证明更深层的网络能学习更多的隐特征表示；但到达一定的层数，更深的网络并不能带来明显的推荐性能明显，还会需要更强大的计算能力，消耗更多的计算资源。

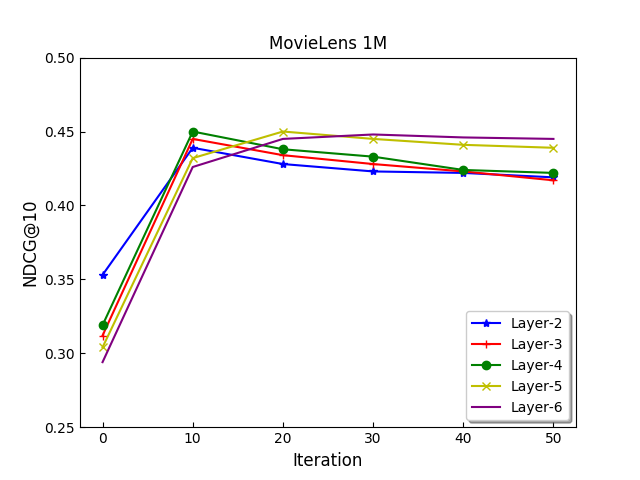
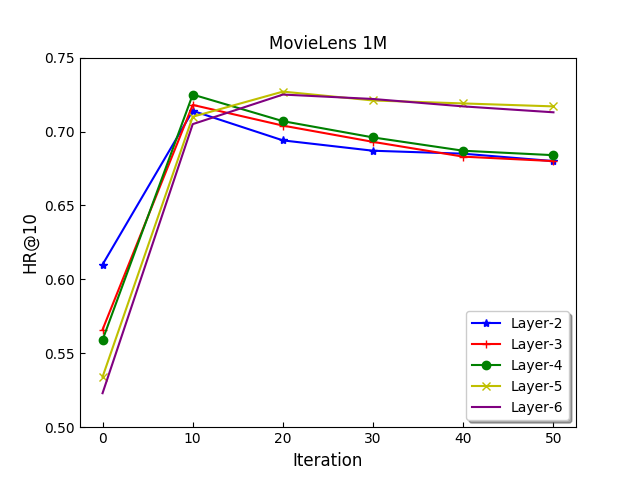


图4-5 网络层数对比实验

在上文中提到，模型需要从未观察到的数据中去抽取负实例来进行模型的训练。我们考虑设置不同的负采样率（即：负样本与正样本的比例）去进行对比实验，观察其对模型的推荐性能的影响。设置学习率为0.001、batch-size为256、隐因子个数为128、嵌入层中采用均值为0，标准偏差为0.01的高斯分布去随机初始化用户和项目的向量表示，采用Adam算法去优化模型，分别在4层[128,64,32,16]和5层[128,64,32,16,8]网络结构上进行了对比，实验结果如图4-6所示。从结果中可以发现，更多的负实例能有效提高模型的性能，当取负实例为9时能获得最好的推荐效果。同样的，更多的负实例需要消耗更多的计算资源，带来计算的负担。

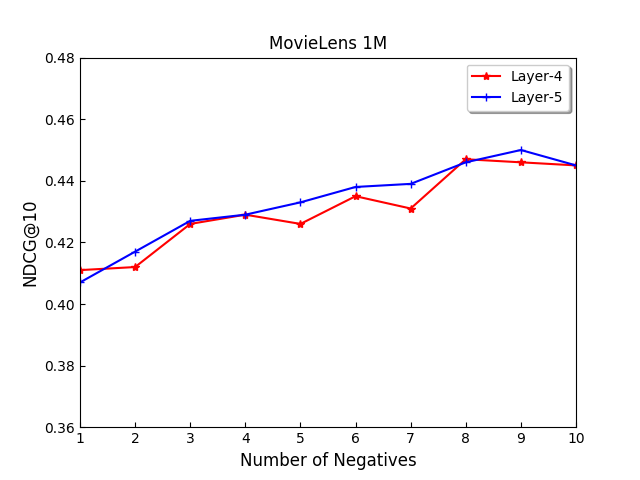
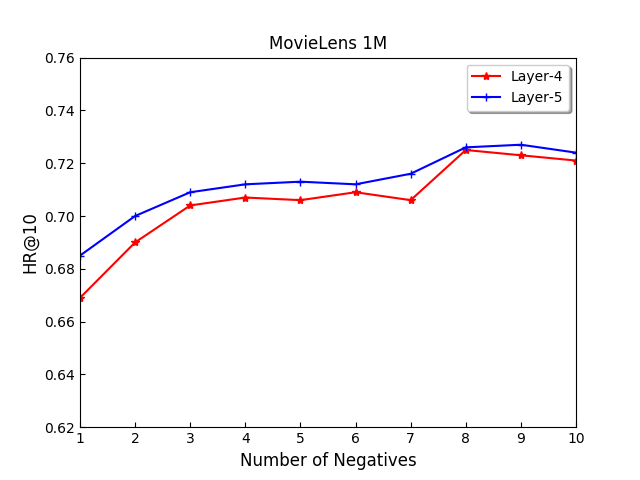


图4-6 负实例个数对比实验

## 4.4本章小结

考虑到多交互深度推荐模型存在的不足之处，受到混合推荐模型和注意力机制的启发，进一步提出了融合注意力机制的多交互神经矩阵分解模型（AM-NMF），并对该模型的整体结构和学习过程进行了介绍说明。在MovieLen 1M和Pinterest数据集与其他推荐模型进行相关实验对比，通过HR和NDCG评价指标结果，表明了所提出的AM-NMF模型能更精确地进行项目推荐。最后针对模型相关参数进了分析，通过MovieLen 1M数据集进行实验对比，验证并说明了相关参数对模型整体推荐性能的影响。

# 第5章 总结与展望

## 5.1总结

近年来，由于深度学习技术的不断发展，为推荐系统的研究带来了新的机遇和挑战。传统的推荐模型只能通过线性方式去获得用户-项目浅层的表示，而且面临着“冷启动”和“稀疏性”问题。基于深度学习的推荐模型可以通过深层次非线性网络结构，去获取用户和项目更深层次特征表示，但目前大多数模型都只是在矩阵技术的基础上引入深度神经网络去构建模型。在本文中，我们在已有的研究工作基础上进行了相关改进，主要工作包括如下三个部分：

（1）考虑到目前大多数模型仅仅使用评分信息作为模型的输入数据，在遭遇“数据稀疏性”问题时会大大降低模型的推荐性能，而且多层神经网络的结构都是标准的MLP模型，只利用最后一层输出的用户和项目隐表示进行交互。针对上述问题，本文提出融合辅助信息的多交互深度矩阵分解模型MA-DMF。首先，在输入数据方面，在用户-项目的评分数据基础上，进一步融合了项目属性和标签等辅助信息，使用One-hot编码后的稀疏向量分别作为模型中用户和项目的输入向量，辅助信息不但利用了辅助信息中包含的用户和项目属性偏好，更是在一定程度上缓解了推荐模型中数据稀疏性的问题；另外，所提模型采取了两个并行的深度神经网络去分别学习用户和项目的隐特征表示，在隐藏层的多层神经网络学习时，对每次网络所学习到的用户和项目隐特征表示进行了一次多交互的操作，即通过点积交互的方式去获得不同网络层所学习到的隐表示结果；最后聚合所有层的内积交互来获得模型的预测结果。在多个数据集上与其他模型进行对比实验，结果表明MA-DMF模型能更准确地进行评分预测，有效地提升推荐质量。

（2）目前大多数基于深度学习的推荐模型对于深层神经网络中所学习到的用户-项目的特征表示都是统一对待，没有考虑到用户-项目特征交互的多样性及关联程度，不同的特征交互具有不同的重要程度，针对这个问题，我们引入注意力机制去学习一个关于其特征交互的权值矩阵，用来判别不同的重要程度。另外，受混合推荐模型启发，将浅层的推荐模型与深度学习推荐模型相结合有去进一步提高整体模型的推荐性能。本文在上一个工作所提出的多交互网络结构基础上，进一步提出了融合注意力机制的多交互神经矩阵分解模型AM-NMF。将用户和项目的评分ID信息One-hot编码后的稀疏向量作为模型的输入；与深层推荐模型共享同一个嵌入层，直接利用嵌入后用户和项目的隐向量进行点积交互得到一个浅层的推荐结果；同样嵌入后的隐向量作为模型隐藏层的输入向量，基于上述提出的多交互深度网络构建模型的隐藏层，在多交互的深度神经网络层之后构建一个注意力网络层，利用多层网络之后的用户和项目隐表示去学习一个注意力权重向量；结合注意力权重向量和多交互的隐表示结果获得一个深层的推荐结果；最后对浅层模型和基于注意力的深层模型推荐结果进行加权作为模型最终的推荐结果，为用户进行Top\_N的项目推荐。在多个数据集上与其他模型进行对比实验，结果表明AM-NMF推荐模型更精确地为用户进行合适的项目推荐，表现出更好的推荐性能。

（3）本文所提出的两个推荐模型对自身的相关参数进行了实验分析，探讨不同的参数和相关操作对所提模型推荐性能的影响。MA-DMF模型分别针对隐层网络结构、隐因子数量、和正则化参数和批标准化操作进行了对比实验，验证其对模型推荐性能的影响。而AM-NMF模型同样通过实验对比，验证了隐因子数量、隐层网络结构和负实例数目对模型推荐结果的影响。

## 5.2展望

本文工作虽然取得了一定的进步，但还存在一定的不足之处和可以进一步探索的地方，未来将可以从以下几点进行研究改进：

（1）在数据稀疏性方面，虽然融合了一些辅助信息，但在未来可以考虑进行跨域信息的融合。通过融合各种不同类型的、跨平台的异构数据进行推荐来进一步解决“冷启动”和“稀疏性”的问题，提升推荐模型性能。

（2）传统的机器学习算法都有自身的优势，在一些机器学习任务中表现出了较好的效果，当前也有很多学者将因子分解机制、矩阵分解等算法与深度学习技术相结合去构造新的推荐模型并取得了较好的效果。在未来可以进一步考虑更多传统的算法与深度学习框架相结合去构建新的推荐模型。

（3）对于个性化推荐，在未来可以进一步考虑用户行为的序列模式、用户间的社会化关系、用户和项目的度量学习、用户和项目属性特征随时间的动态变化等，在DNN,CNN,RNN的基础上，针对不同的推荐场景去构建新的深度学习框架。

# 参考文献

[1] 张晓娜. 中国互联网络信息中心发布第41次《中国互联网络发展状况统计报告》[N]民主与法制时报, 2018-02-06(004).

[2] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[3] 赵晨婷, 马春娥. 探索推荐引擎内部的秘密[J]. 2011-03-16, [2017-09-22]. https://www. ibm. com/developerworks/cn/web/1103\_ zhaoct\_recommstudy1/index. html, 2011.

[4] Tkalcic M, Chen L. Personality and recommender systems[M]. Springer US: Recommender Systems Handbook, 2015: 715-739.

[5] You Wen. Ye Shui-sheng. A survey of collaborative filtering algorithm applied in E-commerce recommend system[J]. Computer Technology and Development, 2006, 16 (9): 70-72.

[6] Bobadilla J, Ortega F, Hernando A, et al. Recommender systems survey[J]. Knowledge-Based Systems, 2013, 46(1): 109–132.

[7] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19(1): 1-15.

[8] 黄立威, 刘艳博. 基于深度学习推荐系统[J]. 计算机学报. 2018, 40(156): 1-28.

[9] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems[C]. In The adaptive web, Springer, 2007: 325-341.

[10] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008: 426-434.

[11] Hu L, Cao J, Xu G, et al. Personalized recommendation via cross-domain triadic factorization[C]. The, International Conference. 2013:595-606.

[12] Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications[J]. Data mining and knowledge discovery, 2001, 5(1-2): 115-153.

[13] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM. December, 1992, 35(12): 61-70.

[14] Leng Yajun, Lu Qing, Liang Chang-yong. Survey of recommendation based on collaborative filtering[J]. Pattern Recognition and Artificial Intelligence, 2014, 27(8): 720-734.

[15] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[16] Wang H, Wang N, Yeung D Y. Collaborative deep learning for recommender systems[C]. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015: 1235-1244.

[17] Zhang F, Yuan N J, Lian D, et al. Collaborative Knowledge Base Embedding for Recommender Systems[C]. The ACM SIGKDD International Conference. ACM, 2016: 353-362.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems[J]. Computer, IEEE, 2009, 42(8): 30-37.

[19] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S.T.Roweis(eds.), Advances in Neural Information Processing Systems[C]. Curran Associates, Inc, 2008: 1257-1264.

[20] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]. Proceedings of the 24th international conference on Machine learning. ACM, 2007: 791-798.

[21] Liwei Huang, Yanbo Liu. Recommendation system based on Deep Learning[J]. Journal of Computer. 2017, 40(156): 1-28.

[22] Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using click through data[C]. Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013: 2333-2338.

[23] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems[C]. In Proceedings of the 24th International Conference on World Wide Web. ACM, 2015:278-288.

[24] Xue H J, Dai X, Zhang J, et al. Deep Matrix Factorization Models for Recommender Systems[C]. IJCAI. 2017: 3203-3209.

[25] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems[C]. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, 2016: 7-10.

[26] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]. Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017: 173-182.

[27] Lin Z, Feng M, Santos C N, et al. A structured self-attentive sentence embedding[J]. arXiv preprint arXiv:1703.03130, 2017.

[28] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, “Attentive collaborative filtering: Multimedia recommendation with item and component-level attention,” in SIGIR, 2017: 335–344.

[29] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, “Neural attentive session-based recommendation,” in CIKM, 2017: 1419–1428.

[30] Xiao J, Ye H, He X, et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks[J]. arXiv preprint arXiv:1708.04617, 2017.

[31] He X, He Z, Song J, et al. Nais: Neural attentive item similarity model for recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(12): 2354-2366.

[32] Cheng Z, Ding Y, He X, et al. A^ 3NCF: An Adaptive Aspect Attention Model for Rating Prediction[C]. IJCAI. 2018: 3748-3754.

[33] Resnick P, Varian HR. Recommender systems. Commun ACM, 1997, 40: 56–58.

[34] Konstan JA, Miller BN, et al. GroupLens: Applying collaborative filtering to usenet news. Comm ACM, 1997, 40(3): 77—87.

[35] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]. International Conference on World Wide Web. ACM, 2001: 285-295.

[36] Lei Y, Chen G. Recommendation-Based Trust Model in P2P Network Environment[M]. Intelligent Information Processing VI. Springer Berlin Heidelberg, 2012: 305-310.

[37] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016: 549–558.

[38] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” in UAI, 2009: 452–461.

[39] Funk S. Netflix Update: Try This at Home[EB/OL]. [2017-04-11]. http://sifter.org/simon/journal/20061211.html.

[40] 王建洋. 基于深度学习的电影推荐系统研究与实现[D]. 西南交通大学, 2018.

[41] M. A. Vorontsov, Stochastic parallel-gradient-descent technique for high-resolution wave-front phase-distortion correction [J]. J. Opt. Soc. Am. A, 1998. Vol. 15, No. 10: p. 2745-2758.

[42] Lian J, Zhang F, Xie X, et al. CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems[C]. Proceedings of the 26th international conference on World Wide Web companion. International World Wide Web Conferences Steering Committee, 2017: 817-818.

[43] Song X, He X, Gao J, et al. Unsupervised learning of word semantic embedding using the deep structured semantic model[J]. 2014.

[44] Rong X. word2vec parameter learning explained[J]. arXiv preprint arXiv:1411.2738, 2014.

[45] Wang X, He X, Nie L, et al. Item Silk Road: Recommending Items from Information Domains to Social Users[J]. 2017:185-194.

[46] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems. 2017: 5998-6008.

[47] Cao D, He X, Miao L, et al. Attentive group recommendation[C]. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 2018: 645-654.

[48] He X, Chua T S. Neural Factorization Machines for Sparse Predictive Analytics[J]. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, In SIGIR. 2017: 355-364.

[49] Rendle S, Gantner Z, Freudenthaler C, et al. Fast context-aware recommendations with factorization machines[C]. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011: 635-644.

[50] Rendle S. Factorization machines[C]. 2010 IEEE International Conference on Data Mining. IEEE, 2010: 995-1000.

[51] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for implicit feedback datasets[C]. In ICDM, 2008: 263-272.

[52] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion[C]. In NIPS, 2013: 926-934.

[53] Harper F M, Konstan J A. The movielens datasets: History and context[J]. Acm transactions on interactive intelligent systems (tiis), 2016, 5(4): 19.

[54] X. Geng, H. Zhang, J. Bian, and T.-S. Chua. Learning image and user features for recommendation in social networks. In ICCV, 2015: 4274-4282.

[55] X. He, T. Chen, M.-Y. Kan, and X. Chen. TriRank: Review-aware explainable recommendation by modeling aspects. In CIKM, 2015: 1661-1670.

[56] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

[57] Zheng L, Noroozi V, Yu P S. Joint deep modeling of users and items using reviews for recommendation[C]. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017: 425-434.

[58] Chen J, Zhang H, He X, et al. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention[C]. Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2017: 335-344.

[58] [Minh-Thang Luong](https://arxiv.org/search/cs?searchtype=author&query=Luong%2C+M), [Hieu Pham](https://arxiv.org/search/cs?searchtype=author&query=Pham%2C+H), [Christopher D. Manning](https://arxiv.org/search/cs?searchtype=author&query=Manning%2C+C+D). Effective Approaches to Attention-based Neural Machine Translation. arXiv preprint arXiv: 1508.04025v3, 2015.

[59] Xiong C, Callan J, Liu T Y. Learning to attend and to rank with word-entity duets[C]. Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017: 763-772.

[60] Chen L, Zhang H, Xiao J, et al. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 5659-5667.

[61] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

# 致谢

研究生三年的求学生涯即将结束，有些短暂，却让人怀念。在这三年来，不但让我的专业知识有了一个很大的提高，也逐渐从一个懵懂少年变得更加成熟和自信。我的成长和收获都离不开老师、同学、家人给与的指导和帮助，在此我向所有帮助、关心、鼓励我的人表示由衷的感谢和深深的祝福。同样感谢那些只通过邮件进行交流，为我提供帮助和支持的陌生科研工作者们，祝福你们身体健康，前程似锦。

在这里，首先要感谢陪伴和指导了我三年的导师，唐雁教授，让我非常有幸成为“唐门”的一员。这三年的研究生生涯里，在唐老师的耐心指导和鼓励下，让我的学术水平和生活能力都得到了很大的提高。在研究学习中，无论遇到任何问题，唐老师都会认真细心地去为我解答；在遇到生活中困惑的问题，也会耐心地为我排忧解难并指明方向，唐老师这种严谨的教学态度和独特的人格魅力也让我深深折服，时刻激励和鞭策着我。正是唐老师的言传身教，让我知道做学术要认真严谨，做学术要稳扎稳打，用心沉淀；更让我知道做人要勤奋刻苦，脚踏实地，有梦想有追求；也正是唐老师的耐心指导，让我的毕业论文能够顺利完成。在这里再次向唐老师表示最真诚的感谢，您辛苦了！

同时，我也要感谢学院的授课老师，张自力老师、贾韬老师、刘志明老师、余国先老师。你们在课堂上不但教会了我云计算、机器学习等方面的专业知识，更是通过自身的经历告诉了我许多人生哲理，教会了我们“做人做事做学问、尽心尽力尽责任”的道理。正是你们的谆谆教导，帮助我发现了自身的不足，为我指明了未来的目标和方向，对我将来的工作和生活都产生了深远的影响。在此毕业之际，心怀不舍和感激，向你们表示真诚的感谢，遇到你们是我研究生三年来最大的收获和荣幸。

另外，我要感谢实验室的师兄师姐牟春倩、胡金戈、侯宇博、王思杰、游婷婷、张艺琨，在研究工作中给予的支持；感谢同门的小伙伴王书涛、毛德磊在生活和学术中给我的帮助和启发；感谢师弟师妹王渊、闫航宇、蒋维、高名衍，谢谢你们在生活上的关心和帮助。还要感谢我的好朋友张龙、王可尧、肖林龙、邹斌，感谢你们在学习和生活上提供的帮助。由衷地感谢你们，很幸运能与你们一起学习和生活，正是你们的陪伴，让我度过了一个快乐并有意义的研究生生涯。

最后，我要感谢我的家人们，特别是父母和我的女朋友彭瑜，感谢你们始终在我身后默默地支持我、鼓励我、包容我，你们的支持和鼓励是我学习和工作最大的动力。

值此毕业论文完成之际，再次向上面所有给予我帮助的人致以最诚挚的谢意！也衷心感谢参加论文评审和答辩工作的各位专家和教授，祝您们身体健康，工作顺利。

李同欢

2019年3月于西南大学

# 攻读硕士学位期间公开发表的论文

[1] 李同欢,

[2] Tonghuan

[3] Tonghuan Li,