文章编号：2096-1472(2018)-01-18-03
DOI:10.19644/j.cnki.issn2096-1472.2018.01.005

基于MyMediaLite平台的推荐方法探究

林 楠，杨文渊，马伊莉，朱婷婷，陈圣磊
(南京审计大学经济与贸易学院，江苏 南京 211815)

摘 要：推荐系统是互联网和电子商务的产物。它是建立在对海量数据训练的基础上的一种智能平台，能够向顾客提供个性化的信息服务和决策。随着电子商务大数据的高速发展，推荐系统正逐渐成为学术界的研究热点之一。针对推荐系统理论性强、内容抽象的特点，本文介绍了以MyMediaLite为平台的个性化推荐实践方案，并详细阐述了其具体的实施过程。通过介绍MyMediaLite的系统结构框架，以及分析基于MyMediaLite的实验过程，为研究者使用MyMediaLite推荐系统库进行算法研究奠定了基础。

关键词：个性化推荐；机器学习；MyMediaLite；推荐系统；协同过滤
中图分类号：TP317 **文献标识码**：A

Research on the Recommendation Method Based on the MyMediaLite Platform

LIN Nan,YANG Wenyuan,MA Yili,ZHU Tingting,CHENG Shenglei
(*School of Economics and Trade*,*Nanjing Audit University*,*Nanjing* 211815,*China*)

Abstract: The recommendation system is the product of Internet and e-commerce.It is an intelligent platform built on the basis of massive data training.It provides personalized information service and decision-making to customers. With the rapid development of big data in electronic commerce,the recommendation system is becoming one of the hot topics in the academic field.In view of the highly theoretical and abstract nature of the recommendation system,this paper introduces the personalized recommending practice scheme based on MyMediaLite,and expounds its specific implementation process in detail.By introducing the system framework of MyMediaLite and analyzing the experimental process based on MyMediaLite,it establishes a foundation for researchers to conduct studies on the algorithms with MyMediaLite recommendation system library.

Keywords:personalized recommendation;machine learning;Mymedialite;the recommendation system;collaborative filtering

1 引言(Introduction)

互联网的日益普及，使得电子商务成为人们生活中不可或缺的一部分。随着大量商品和用户群体的涌入，电子商务系统中的数据量呈爆炸式增长，这便使得推荐系统获得了极大的发展空间。短时间内涌现出很多诸如SVDFeature[1]、MyMediaLite[2]、Apache Mahout[3]等知名的开源推荐系统。SVDFeature包含一个很灵活的Matrix Factorization推荐框架，能方便地实现奇异值分解(SVD)和改进的SVD++[4]等算法。Apache Mahout是一种能实现大多数分布式机器学习和数据挖掘的平台。其中有一部分包含了推荐算法中的协同过滤算法，但大部分其他类型推荐算法并未涉及。MyMediaLite提供了多种项目预测和评级预测任务下先进的算法，以及为大多数推荐模型提供了增量式更新数据。与其他平台相比，MyMediaLite囊括的算法与模型更加全面，对于推荐系统的系统学习和研究更有帮助。

2 MyMediaLite推荐系统库(MyMediaLite recommendation system library)

2.1 MyMediaLite简介

MyMediaLite是由德国希尔德海姆大学基于微软.NET平台开发的轻量级、多用途、可拓展的推荐系统算法库。它包含了包括SVD++算法、K-近邻(KNN)[5]算法和直接优化物品排序的矩阵分解算法(BPRMF)[6]等在内的几十个不同的推荐算法。除了提供了常见场景的推荐算法，MyMediaLite也有Social Matrix Factorization这样独特的功能。在实现方面，MyMediaLite推荐系统库涉及了两个最常见的任务：评级预测和项目预测。此外，精心设计的软件框架可以使新算法的实现和评估更加容易。通过使用开源的Mono的项目，MyMediaLite可以在所有相关操作系统上使用。这个库的使用不再仅限于C#，它可以很容易被其他语言如Ruby和Python调用。

基金项目：南京审计大学大学生创新创业训练计划资助(编号：2016AX04001Z).

2.2 推荐任务

2.2.1 评级预测

评级预测加载的评级是用户偏好的显式反馈(explicit feedback)。评级可以分为1—n级，例如1—5级，5级表示用户十分喜欢，而1级则表示不喜欢。评级预测算法根据给定的已知评级集估算未知的评级，从而推荐系统可以根据预测的评级进行推荐。

MyMediaLite中包含的评级预测算法有：不同变种的K-近邻(KNN)分类算法、简单基线算法(Slop-One)[7]和矩阵分解算法(SVD++)等。

评级预测下的常用的准确性度量指标：RMSE(均方根误差)和MAE(平均绝对误差)。

(1)RMSE：均方根误差是均方误差的算术平方根，如式(1)所示。其中observedt表示实际评级，predictedt表示预测评级。均方根误差常用于衡量推荐结果与实际结果的离散程度。均方根误差越大，表明推荐结果与实际结果离散程度越大，推荐算法越不精准。均方根误差越小，表明推荐结果与实际结果的离散程度越小，推荐算法精度越高。

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{t=1}^{N}(observed_t-predicted_t)^2} \tag{1}$$

(2)MAE：平均绝对误差是绝对误差的平均值，如式(2)所示。平均绝对误差能更好地反映预测值误差的总体情况。平均绝对误差越大，表明推荐越不精准。平均绝对误差越小，表明推荐越精准。

$$\mathrm{MAE}=\frac{1}{N}\sum_{t=1}^{N}|observed_t-predicted_t| \tag{2}$$

2.2.2 项目预测

与评级预测不同的是，项目预测加载的数据是用户偏好的隐式反馈(implicit feedback)，如用户浏览页面的时间、转发行为、购买等。购买商品或者长时间浏览某个商品的页面的行为可以很好地从侧面体现出用户的偏好。现实生活中的大多数推荐系统(如电子商务等)也往往并不依赖评级，它们往往根据用户的历史行为推测出用户的偏好，从而推荐出与用户偏好相一致的产品。

MyMediaLite中包含的项目预测算法有：K-近邻分类算法、简单基线算法(Random,MostPopular)、矩阵分解算法(BPR-MF,WR-MF)等。

项目预测下常用的准确性度量指标有：prec@N和AUC。

(1)prec@N(precision at N)：推荐准确率表示算法推荐成功的比率，如式(3)所示。其中test表示测试集，top-N表示系统推荐给用户的N个项目。

$$\mathrm{prec@N}=\frac{|test\cap top-N|}{N} \tag{3}$$

(2)AUC(Area Under the ROC Curve)：AUC被定义为ROC曲线下的面积。ROC(Receiver Operating Characteristic)有两个指标：sensitivity(敏感度)和Specificity(特异度)[8]。前者为任选一个用户喜欢的项目，该项目被系统推荐的概率；后者为任选一个用户不喜欢的项目，该项目未被系统推荐的概率。设定一个域值，项目被正确推荐的概率大于域值的，认为是用户喜欢的项目；概率小于域值的，认为是用户不喜欢的项目。图1所示为ROC曲线，纵坐标Sensitivity，横坐标为1－specificity。

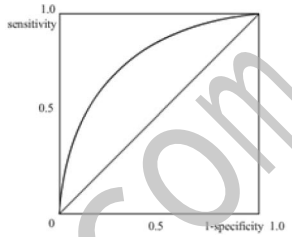

图1 ROC曲线

Fig.1 ROC curve

Sensitivity值在0到1之间变化，给定一个域值，就对应图中左上角的曲线上的一个点。图中穿过原点和(1,1)点的直线表示一个随机推荐的系统，即任选一个用户喜欢的项目，该项目被系统推荐的概率为0.5。由图1可知，曲线越向左上角靠近，则AUC越大，说明推荐系统的推荐精度越高，反之则越低。

3 基于MyMediaLite的实验过程(Experimental process based on MyMediaLite)

由于MyMediaLite使用C＃进行编写和Windows的广泛使用，本次实验将使用Visual Studio 2010的开发环境进行实验。

3.1 数据选取

本次实验我们使用的是推荐系统研究中权威的测试集MoiveLens-100k。数据集一共记录了943名用户对于1682部电影的100000条评级记录(1—5)。其中每名用户至少评论过20部电影。

其中u.base文件中记录的是数据集中完整的数据。它是一个制表符分隔数据的列表，每一行的数据包含用户ID、项目、评级、时间戳。u1.base和u1.test到u5.base和u5.test则是将u.base中的完整数据按照80%/20%的比率分割成的5组训练数据与比对数据。ua.base、ua.test和ub.base、ub.test则是按照将每个用户10%的评级加入比对数据的规则进行分割得到的训练数据和比对数据。训练数据(.base)用于训练模型，从而得到用户的偏好，而比对数据(.test)则用于测试推荐系统推荐的精准度。

需要注意的是，将MoiveLens数据集用于MyMediaLite时，RatingPrediction下的算法将截取前三列作为数据(用户、项目、评分)，而ItemRecommender下的算法将只取前两列作为数据(用户、项目)。

3.2 运行配置

MyMediaLite推荐系统库是以C#源码的形式提供给用户使用的，它实际上是一个依赖于.NET框架的程序集。虽然源代码的形式可以让我们深入了解算法底层的细节，从而根据业务的场景进行算法配置和调优，但这样无疑增加了很多使用难度。在使用MyMediaLite进行推荐实验之前，我们需要进行相关的配置。

3.2.1 引用的动态链接库文件

打开程序集中源码(src)文件夹下的解决方案文件MyMediaLite.sln，则可以在VS2010中查看MyMediaLite的解决方案。通过项目MyMediaLite，可以对MyMediaLite推荐算法库中算法的源码进行查看。

MyMediaLite项目需要四个类库文件的支持：C5、MathNet.Numerics、MathNet.Numerics.IO和Mono.Posix。

需要注意的是，MyMediaLite推荐算法库的发行版不会自带Mono.Posix的库文件，需要用户自行下载并引用，否则项目将无法通过之后的编译。

3.2.2 动态链接库文件的生成

将MyMediaLite项目生成，之后便可以在src/MyMediaLite/bin/Debug目录下找到生成的MyMediaLite.dll、MyMediaLite.pdb和项目引用的库文件。

动态链接库文件(.dll)为我们使用MyMediaLite推荐系统库提供了应用程序接口。调试配置文件(.pdb)可以为我们的应用程序和源码之间建起一座桥梁，它将让我们在调试时准确定位到源码。

3.3 新建实验项目

使用C#新建一个控制台程序，在项目中引用MyMediaLite.dll，然后在主函数中输入相应代码(以评级预测任务中的SVD++为例)，源代码如下所示。该代码展示了使用MyMediaLite推荐系统库的常见步骤：(1)加载数据；(2)建立推荐系统模型；(3)训练数据；(4)得出推荐结果。

```
using System;
using MyMediaLite.Data;
using MyMediaLite.Eval;
using MyMediaLite.IO;
using MyMediaLite.RatingPrediction;
public class RatingPrediction
{
    public static void Main(string[] args)
    {
    var training_data=RatingData.Read(args[0]);
    var test_data=RatingData.Read(args[1]);
    var recommender=new SVDPlusPlus();
    recommender.Ratings=training_data;
    recommender.Train();
    var results=recommender.Evaluate(test_data);
    Console.WriteLine("RMSE={0} MAE={1}",
results["RMSE"], results["MAE"]);
    Console.WriteLine(results);
     Console.WriteLine(recommender.Predict(1, 1));
        var r=recommender.Recommend(1, 20);
        for (int i=0; i<20; i++)
            Console.WriteLine(r[i]);
        Console.Read();
    }
}
```

3.4 参数设置

接下来设置程序的命令行参数，将一组训练数据与比对数据作为参数传入到应用程序中。在项目属性中选择调试，将启动选项的工作目录选定为MoiveLens数据集的文件地址，命令行参数选择u1.base文件和u1.test文件(以空格隔开)，这样就能顺利地把训练数据和比对数据加载到应用程序中。

3.5 结果分析

运行应用程序的结果如图2所示。由结果可知，通过对训练数据集u1.base的训练，以及与比对数据集u1.test的比对，推荐算法SVD++推荐的均方误差(RMSE)为0.9701232，平均绝对误差(MAE)为0.7696601。根据用户的偏好模型，推荐算法SVD++给出了用户1对项目1的预测评级4.022615，以及用户1预测评级最高的20个商品的编号和相应的评级。

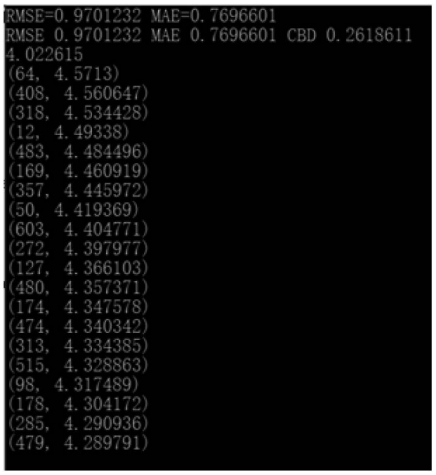

图2 推荐结果

Fig.2 Recommendation results

4 结论(Conclusion)

个性化推荐系统具有理论性强、方法繁多、实践困难等特点。使用诸如MyMediaLite推荐算法库这类开源的平台，能够帮助使用者更加全面、深入、系统地学习推荐系统。但随着个性化推荐系统研究的深入和技术的快速提升，仅仅局限于学习MyMediaLite推荐系统库中的推荐方法还是不够的，我们将继续深入研究，以改进MyMediaLite中的算法。

(下转第11页)