SVD++算法的规则化参数和学习步长探究

摘要：推荐算法中一个非常流行的算法SVD++的基础上，对该算法重要的初始化参数进行了研究。使用当前流行的随机梯度下降算法，使用数据集MovieLens-10k,在其它变量固定的情况下，当规则化参数逐渐增加、学习步长逐渐降低时，推荐算法的准确率逐渐提高，最后到达一个最高值。对学习速率从0.001~0.1，每次递增0.001，对规则化参数从0.001~0.1，每次递增0.001,从而得到10000个实验结果，在训练集-数据集切割比分别为0.8和0.6时，得到最优的规则化参数，并且在movielens-20m上对结果进行验证，说明了我们求得的规则化参数和学习步长在其他数据集上也有足够高的准确性。

近年来协同过滤推荐算法很受欢迎并且已经取得了很多的成果。在netflix比赛中，协同过滤扮演了很重要的角色，这也促进了它的流行。矩阵因式分解算法因为最近的创新,成为实现协同过滤算法的第一选择， 这里描述了如何使用隐世反馈来提高模型的准确性。协同过滤算法（Collaborative filtering,CF）基于评分或者隐世反馈行为(比如是否购买)来对特定的用户推荐商品，CF算法不需要任何用户或者项目的额外信息。这种CF算法能够对很多场景提供高准确率的推荐。开始于2006年10月的Netflx Prize对协同过滤算法产生了很大的促进作用，这是第一次研究社区提供一个大规模的的数据集，它包括1000万条电影评分，这吸引了许多人建立推荐系统来改进推荐准确性。因为所有的方法都用相同的标准来判断，更强大的模型能够取得的效果更加直接显著。推荐系统依赖各种类型的输入，最方便的是高质量的显式数据，用户对产品直接表明他们的偏好程度。例如netflix收集对电影收集星级评分，TiVo用户通过使用大拇指向上表示同意，大拇指向下表示反对来表明的电视节目的偏好。因为明确的反馈的数据集是很少的，一些推荐系统通过更多的隐世反馈来推断用户的偏好，这些隐式反馈通过观察用户行为来反映用户的偏好。在没有提供显式偏好的情况下，隐式偏好的资源是非常重要的。为了建立推荐系统，CF系统需要将两类基本的不同的输入相关联：用户和项目。CF推荐算法主要包括两种技术：基于邻居的方法和潜在因素方法。基于邻居的方法集中研究用户间或者项目间的关系，一个item-item模型基于一个用户对类似项目的评分。潜在语义模型，比如矩阵因式分解算法（比如SVD），通过将user-item-rating转换到一个相同的潜在因素空间。这个潜在空间通过用户和产品的因子矩阵来解释评分，从而提供推荐。矩阵因式分解技术实现方便并且有较高的准确性，并且因式分解技术能够操作隐式数据和时间信息。

推荐算法概述：

信息过载是互联网面临的主要挑战 . 如何从过载的信息中主动为用户定位和推送其感兴趣的内容是推荐系统的主要任务

随着电子商务的发展 ,商家提供的商品种类和数量急剧增长 . 具有明确需求的用户可通过搜索查找想购买的商品 . 然而 ,用户需求通常具有不确定性和模糊性 . 据亚马逊统计 ,在其网站购物的客户中 ,有明确购买意向的仅占 16 %. 如果商家能够从海量商品中把满足用户模糊需求的商品主动推荐给用户, 则有望将潜在需求转化为实际需求, 不仅能提高电子商务网站的销售量, 还有助于提高用户对网站的忠诚度. 在此背景下,能够根据用户特征有针对性推荐商品的个性化推荐系统应运而生, 并被广泛应用. 包括 Am azon 、eBay 、YouTube 和 Goog le 在内的诸多网站都部署了不同形式的推荐系统, 并产生了巨大的商业利润 . 据统计,2006 年推荐系统为 Amazo n 提高了 30 %的销售额

推荐系统具有重要的应用价值 ,不仅成为计算机领域具有挑战性的研究课题之一, 还吸引了来自数学、物理 、认知 、人工智能、管理、市场营销等众多领域的研究者. 2006 年 10 月 ,美国 Ne tfilx 公司举行了一场奖金高达 100 万美元的竞赛 ,征集更为有效的推荐算法. 这场历时三年的竞赛吸引了来自 186 个国家的 4 万多个团队参加

推荐系统还面临诸多未被很好解决的理论和应用层面上的难题 , 仍是信息检索 、数据挖掘 、机器学习和人工智能等领域的研究热点

# 推荐算法:

推荐算法的基本模型：

推荐算法的基本模型

# 推荐系统评价准则

[3]根据推荐的目的，推荐算法或为用户估计特定项目的评分，或为用户产生一个推荐项目列表，前者采用平均绝对误差(MAE)、均方根误差(RMSE)评价估分的准确性，后者多采用信息检索领域常用的查全率(recall)、查准率(precise)评价推荐列表的准确性。

一般将数据分成训练集和测试集两个部分，训练集用于模型中的参数估计，测试集用于评价算法的性能。

1. 平均绝对误差（mean absolute error,MAE）

MAE用于度量推荐算法的估计评分与真实值之间的差异，MAE值越小，估计的准确性越高，定义如下：

MAE用于度量推荐算法的估计评分与真实值之间的差异。MAE值越小，估计的准确性越高，定义如下：

其中pij为用户i对项目j的估计评分，rij+为用户i对项目j的真实评分，集合U和I分别为测试集中的用户集合和项目集合。n为pij的个数。文献[5]考虑到不同的推荐系统可能采用的评分等级不同，对EMAE进行了规范化处理

1. 均方根误差（root mean squared error,RMSE）

RMSE是netflix竞赛采用的评价准则。RMSE值越小，算法的准确度越高，定义如下：

1. 查全率（recall）

用于测量推荐列表中是否包含了用户偏好的全部项目。令Li表示推荐算法为用户i产生的推荐列表，Ri表示测试集中用户i偏好的全部项目，有：

其中，n表示训练集中用户数目表示推荐列表中被实际用户偏好的项目数目

1. 查准率（precision）

用于度量推荐列表中是否都是用户偏好的项目，定义为：

查全率和查准率没有考虑推荐项目的顺序问题—根据用户对项目的偏好程度排序。一种改进方法是：一次考察推荐列表中每一项目，若此项目在测试集中被用户选择过（说明用户偏好该项目），则计算到此项目位置推荐列表的查准率，进而对所有这样的查准率进行平均，作为该推荐结果的查准率

这里给了m个user和n个item的评分。我们准备了特殊的索引字符来区分user和item：对用户u，v，对项目I,j,l，一个评分rui表明用户u对项目i的偏好，越大的值代表越强烈的偏好。我们区分预测值和已经知道的值，通过使用 表示rui的预测值。（u，i）对存储在集合K={（u,i）| rui是已知的},每一个用户u绑定一个item的集合写作R(i),包括所有评价过项目i的用户，相似度使用一个集合N(u)，包括所有的被用户u评分过的item。

通过拟合发现的数据来对学习这个模型。我们模型的目标是生成预测未知值的模型。因此，应该要防止对发现数据的过拟合问题规则化，规则化参数是常量λ1，λ2…..通过交叉验证来确定这些值。这些值越大，规则化参数占的比重越大。

基线预测：CF模型尝试捕捉用户和项目的交互，能够产生不同的评分值。然而，许多发现的值独立于用户和项目。一个原则性的例子是传统的CF数据探索用户和项目，即一些用户比其他用户给更高的值，一些项目比其他电影有更高的值。我们将捕捉这些影响，不止包括user-item交互，而且还包括基线预测（即偏差）。因为这些预测能够捕捉许多被发现的信号，这对能否进行准确建模是很重要。这些模型保证将真正的user-item的交互隔离开来，从而建立更准确的用户偏好模型。使用u作为全局平均评分。一个基线预测对未知的rui由bui

和user-item交互组成，bui是user和item的影响：

bui =μ+bu+ bi

参数bu 和bi分别表示发现的user u和 item i的偏差，例如，猜想我们想要预测小明对战狼二的预测，现在，所有电影评分均值是3.7，并且，战狼2拍摄水准比平均水平要搞一些，可能比平均值要高0.5分左右，另外，小明是一个挑剔的评分者，他可能比平局值要低0.3的评分，因此，通过计算3.7-0.3+0.5，基线预测中小明的战狼二的基线预测值为3.9分。

为了估计bu和bi，可以通通过处理下面的最小平方问题

（公式）

这里，第一个项用来发现bu和bi的值来拟合给定的评分，规则化项目来避免过拟合通过惩罚参数的操作。这个问题能够使用随机梯度下降算法来优化。

对netflix数据平均评分μ是3.6，对于学习的用户偏差，他们的均值是0.044对标准的0.41的标准方差。平均的绝对值|bu|是0.32，学习的项目偏差平均是-0.26，标准差是0.48。平均的绝对值（|bi|）是0.43

为了比较不同算法的准确性，我们计算大约1亿的数据通过匿名的形式在1999年11月到2005年12月。评分是从1到5的整数

Latent factor models approach collaborative filtering with the holistic goal to uncover latent features that explain observed ratings; examples include pLSA [15],

neural networks [24], Latent Dirichlet Allocation [7], and models that are induced

by factorization of the user-item ratings matrix (also known as SVD-based models). Recently, matrix factorization models have gained popularity, thanks to their

attractive accuracy and scalability

In information retrieval, SVD is well established for identifying latent semantic

factors [9]. However, applying SVD to explicit ratings in the CF domain raises difficulties due to the high portion of missing values. Conventional SVD is undefined

when knowledge about the matrix is incomplete. Moreover, carelessly addressing

only the relatively few known entries is highly prone to overfitting. Earlier works

relied on imputation [16, 26], which fills in missing ratings and makes the rating matrix dense. However, imputation can be very expensive as it significantly increases

the amount of data. In addition, the data may be considerably distorted due to inaccurate imputation. Hence, more recent works [4, 6, 10, 17, 23, 24, 28] suggested

modeling directly only the observed ratings, while avoiding overfitting through an

adequate regularized model.

In this section we describe several matrix factorization techniques, with increasing complexity and accuracy. We start with the basic model – “SVD”. Then, we

show how to integrate other sources of user feedback in order to increase prediction

accuracy, through the “SVD++ model”

Matrix factorization models map both users and items to a joint latent factor space

of dimensionality f , such that user-item interactions are modeled as inner products

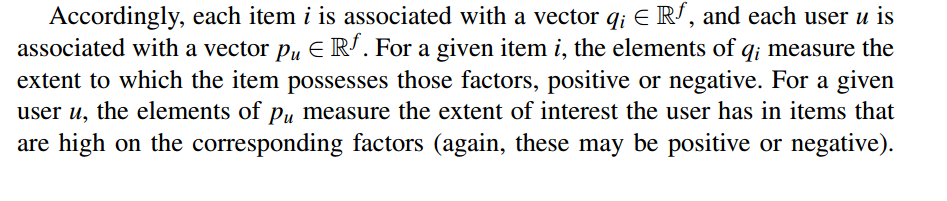
in that space. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. For example,

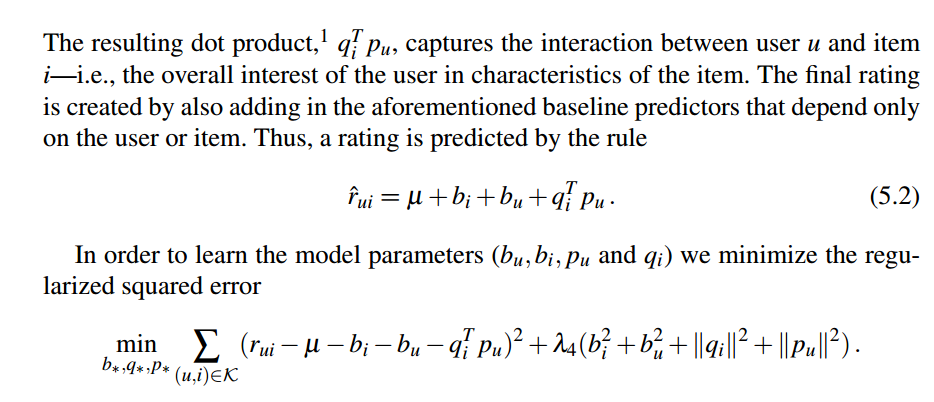
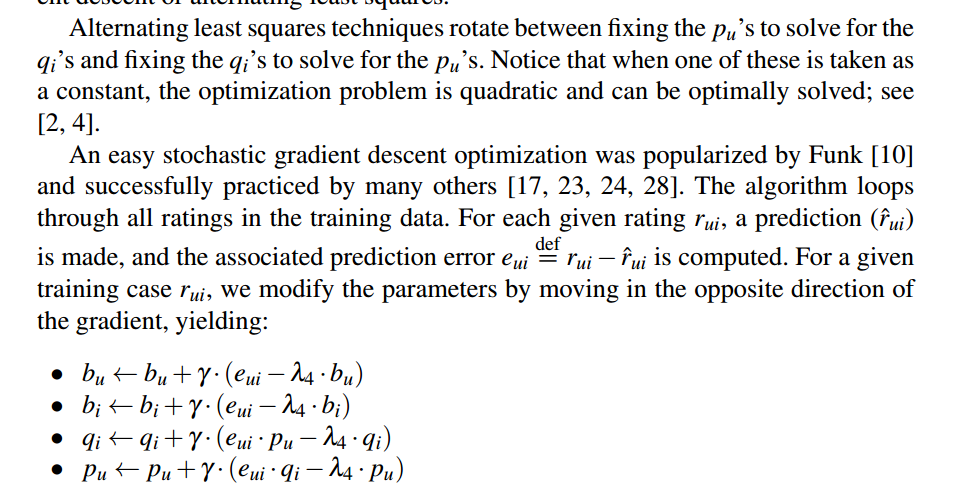
when the products are movies, factors might measure obvious dimensions such as

comedy vs. drama, amount of action, or orientation to children; less well defined

dimensions such as depth of character development or “quirkiness”; or completely

uninterpretable dimensions



Prediction accuracy is improved by considering also implicit feedback, which provides an additional indication of user preferences. This is especially helpful for those

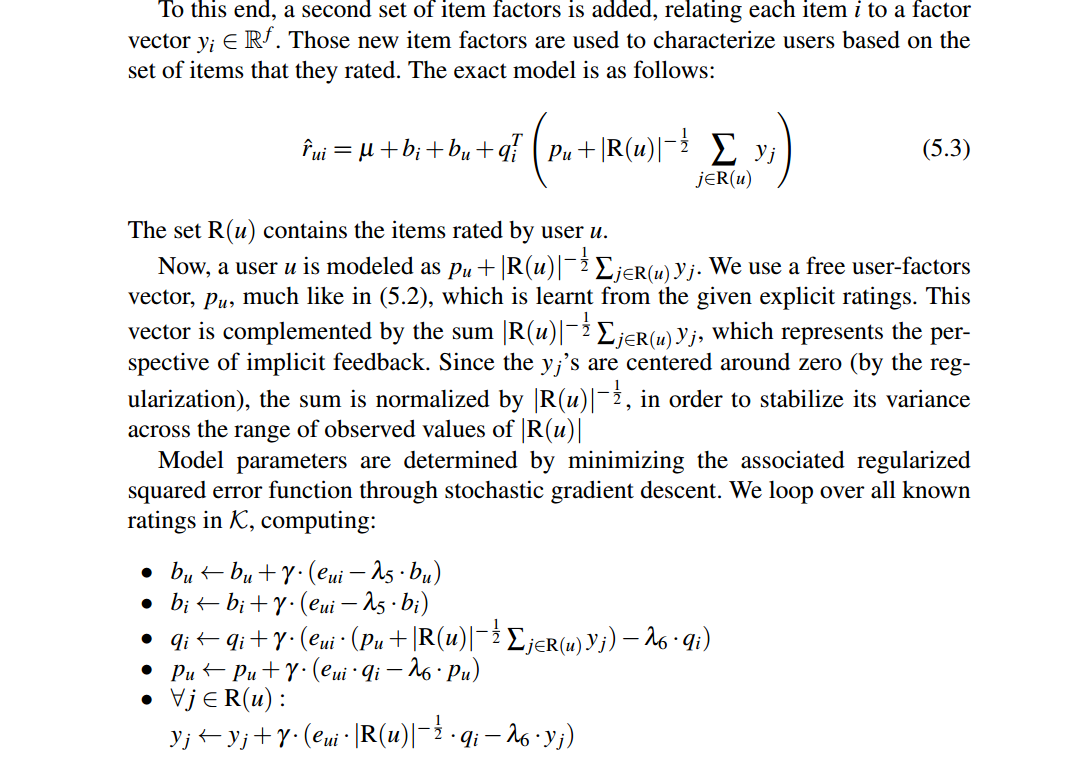
users that provided much more implicit feedback than explicit one. As explained

earlier, even in cases where independent implicit feedback is absent, one can capture a significant signal by accounting for which items users rate, regardless of their

rating value. This led to several methods [17, 23, 25] that modeled a user factor by

the identity of the items he/she has rated. Here we focus on the SVD++ method [17],

which was shown to offer accuracy superior to SVD



When evaluating the method on the Netflix data, we used the following values for

the meta parameters: γ = 0.007, λ5 = 0.005, λ6 = 0.015. It is beneficial to decrease

step sizes (the γ’s) by a factor of 0.9 after each iteration. The iterative process runs

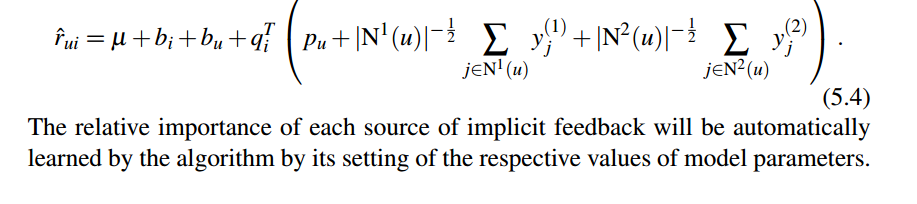
for around 30 iterations until convergence.

Several types of implicit feedback can be simultaneously introduced into the

model by using extra sets of item factors. For example, if a user u has a certain

kind of implicit preference to the items in N1(u) (e.g., she rented them), and a different type of implicit feedback to the items in N2(u) (e.g., she browsed them), we

could use the model



# 参考文献

[1]Kantor P B. Recommender systems handbook[M]. Berlin, Germany:: Springer, 2015.

[2] Takács G, Pilászy I, Németh B, et al. Matrix factorization and neighbor based algorithms for the netflix prize problem[C]//Proceedings of the 2008 ACM conference on Recommender systems. ACM, 2008: 267-274.

[3] 杨博, 赵鹏飞. 推荐算法综述[J]. 山西大学学报: 自然科学版, 2011 (3): 337-350.

1、添加一个空的图：

fig=plt.figure()

2、突出点，其他的为虚线：

plt.plot(randn(30).cumsum(),'ko--')

3、设置刻度：

ticks = ax.set\_xticks([0,250,500,750,1000])

4、设置x label：

ax.set\_xlabel('Stages')

5、添加图例：

ax.legend(loc='best')

6、设置标题：

ax.set\_title(‘jskkd’)

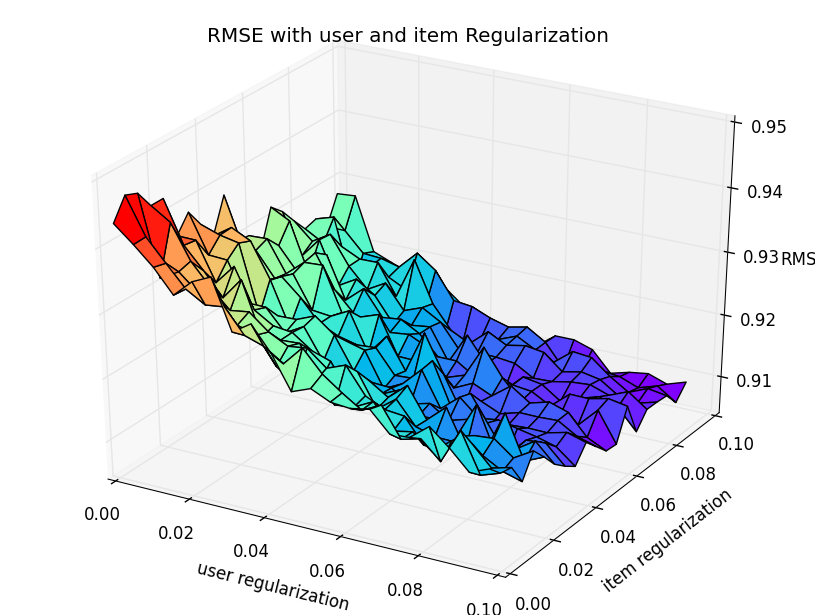
7、如果分训练集和测试集，当结果只剩一个时，每次的RMSE值都会不一样，有时候甚至会是NAN

立体我不想画一个三角面了，想画一个平滑曲面，还是算了，找不到，我看看哪里写错了，为什么我看x轴是规则化参数而不是学习速率

三维图把图例都做出来，二维图把图例做出来，然后根据一个定量，画多个二维图，然后分析图是这样的原因，整理好后给老师看，已经做完第一个实验的分析，没有做第二个实验的分析

下面继续做第二个实验的工作，Test12给加了标签，然后继续在test12中二维分析随user和item那个RMSE变化大，画出图，分析这两个规则化参数那个影响大，明天写相关的论文、背景等

下面分析user regularization和item regularization 在固定的learn rate 下的准确率变化情况

选取learnrate = 0.03

当选取learnrate的值为0.03时，RMSE随user regularization 和item regularization变化如图所示

从图中可以看出在user regularization 和item regularization 增大过程中，RMSE值逐渐变小，准确率逐渐提高