

个人资料



sun_168

访问：120027次
积分：1546
等级：BLOG > 4
排名：第15612名

原创：32篇 转载：7篇
译文：1篇 评论：17条

文章搜索

文章分类

Linux (12)
数据挖掘 (3)
算法 (3)
hadoop (9)
C++ (6)
内存 (3)
java (7)
感慨 (1)
面试题 (1)
实用电脑技术 (10)
论文写作 (1)
matlab (1)

文章存档

2014年03月 (1)
2013年12月 (1)
2013年02月 (1)
2013年01月 (2)
2012年12月 (6)

展开

阅读排行

在word中插入分页符，多
(19405)
您的系统管理员不允许访

学院APP首次下载，可得50C币！ 欢迎来帮助开源“进步” 当讲师？爱学习？投票攒课吧 CSDN 2015博客之星评选结果公布

浅谈矩阵分解在推荐系统中的应用

标签：matrix 协同过滤算法 推荐系统 矩阵分解

2014-03-14 16:28 10004人阅读 评论(5) 收藏 举报

分类： 数据挖掘 (2) 算法 (2)

版权声明：本文为博主原创文章，未经博主允许不得转载。

推荐系统是当下越来越热的一个研究问题，无论在学术界还是在工业界都有很多优秀的人才参与其中。近几年举办的推荐系统比赛更是一次又一次地把推荐系统的研究推向了高潮，比如几年前的Netflix百万大奖赛，KDD CUP 2011的音乐推荐比赛，去年的百度电影推荐竞赛，还有最近的阿里巴巴大数据竞赛。这些比赛对推荐系统的发展都起到了很大的推动作用，使我们有机会接触到真实的工业界数据。我们利用这些数据可以更好地学习掌握推荐系统，这些数据网上很多，大家可以到网上下载。

推荐系统在工业领域中取得了巨大的成功，尤其是在电子商务中。很多电子商务网站利用推荐系统来提高销售收入，推荐系统为Amazon网站每年带来30%的销售收入。推荐系统在不同网站上应用的方式不同，这个不是本文的重点，如果感兴趣可以阅读《推荐系统实践》（人民邮电出版社，项亮）第一章内容。下面进入主题。

为了方便介绍，假设推荐系统中有用户集合有6个用户，即 $U=\{u1, u2, u3, u4, u5, u6\}$ ，项目（物品）集合有7个项目，即 $V=\{v1, v2, v3, v4, v5, v6, v7\}$ ，用户对项目的评分结合为R，用户对项目的评分范围是[0, 5]。R具体表示如下：

表 1 用户对项目的评分表

$\begin{matrix} V \\ U \end{matrix}$	v1	v2	v3	v4	v5	v6	v7
u1	3	?	5	5	?	4	?
u2	5	?	?	4	?	?	?
u3	5	5	?	?	5	5	?
u4	?	4	4	?	2	5	?
u5	?	5	?	5	3	3	?
u6	?	?	?	?	?	?	?

推荐系统的目标就是预测出符号“?”对应位置的分值。推荐系统基于这样一个假设：用户对项目的打分越高，表明用户越喜欢。因此，预测出用户对未评分项目的评分后，根据分值大小排序，把分值高的项目推荐给用户。怎么预测这些评分呢，方法大体上可以分为基于内容的推荐、协同过滤推荐和混合推荐三类，协同过滤算法进一步划分又可分为基于内存的推荐（memory-based）和基于模型的推荐（model-based），本文介绍的矩阵分解算法属于基于模型的推荐。

矩阵分解算法的数学理论基础是矩阵的行列变换。在《线性代数》中，我们知道矩阵A进行行变换相当于A左乘一个矩阵，矩阵A进行列变换等价于矩阵A右乘一个矩阵，因此矩阵A可以表示为 $A=PEQ=PQ$ （E是标准阵）。

矩阵分解目标就是把用户-项目评分矩阵R分解成用户因子矩阵和项目因子矩阵乘的形式，即 $R=UV$ ，这里R是 $n \times m$ ， $n=6$ ， $m=7$ ，U是 $n \times k$ ，V是 $k \times m$ 。直观地表示如下：

(15978)
安装 Realtek HD Audio I

(15398)
浅谈矩阵分解在推荐系统 (9981)

C++ 优先队列用法 (8712)

Hadoop中mapred包和m (5279)

不容忽视的setJarByClas (4806)

重装win系统后修复ubun (3625)

Linux的LiveCd与CD、D (3533)

Hadoop集群中添加硬盘 (3044)

评论排行

浅谈矩阵分解在推荐系统 (5)

在word中插入分页符,多 (4)

Hadoop中mapred包和m (3)

求任意数据类型的字节长 (1)

第一个Hadoop程序 (1)

再论Java的随机数 (1)

Linux的LiveCd与CD、D (1)

google的一道面试题 (1)

在map函数中获取当前记 (0)

C程序在Linux中的编译 (0)

推荐文章

*App竞品技术分析 (6) 热修复

*架构设计: 系统间通信 (17)
——服务治理与Dubbo 中篇 (分析)

*你的计划为什么执行不下去? 怎么破?

*图解堆算法、链表、栈与队列 (多图预警)

*【android】仿360手机卫士的简易设计思路及源码

*Android平台Camera实时滤镜实现方法探讨(九)-磨皮算法探讨(一)

最新评论

浅谈矩阵分解在推荐系统中的应/姚攀2015: 赞

浅谈矩阵分解在推荐系统中的应/haizhiguang:: 用户因子向量和项目因子向量的更新公式, 迭代更新公式即可找到可接受的局部最优解。迭代终止的条件下面...

浅谈矩阵分解在推荐系统中的应/机智的大龙哥: 请问一下如果新来了一个user,该如何填上分数呢?

浅谈矩阵分解在推荐系统中的应/flybabala: 谢谢啦, 写得详细, 希望您能继续更新和扩展。

浅谈矩阵分解在推荐系统中的应/mhttee: 写的很全面, 尤其是很小的点都有提到, 不错的参考

求任意数据类型的字节长度 zlingh: #define msizeof(L_Value) ((char*)(&L_Value + 1))~...

在word中插入分页符, 多出一行 丧尽心: 说这么多没解决问题顶个毛用

Hadoop中mapred包和mapred/jack_qin1992: 很好的内容, 学习了!

第一个Hadoop程序 Cai_i: 能直接再eclipse中运行吗, 为什么我运行不了, 而按照你的方法打包jar后才行。。

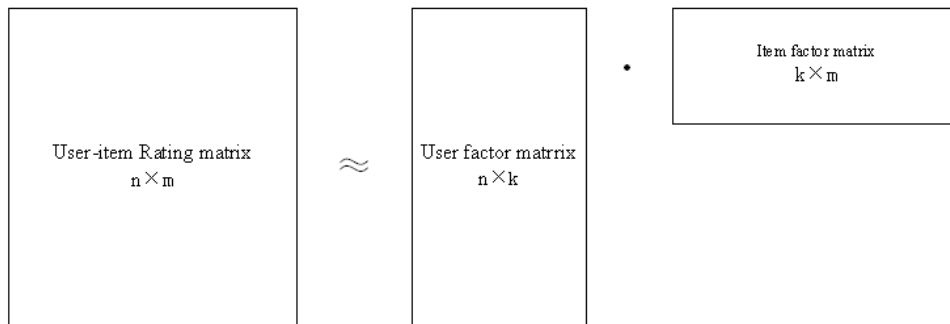


图 1 矩阵分解原理的直观表示http://blog.csdn.net/sun_168

高维的用户-项目评分矩阵可以分解为两个低维的用户因子矩阵和项目因子矩阵, 因此矩阵分解和PCA不同, 不是为了降维。用户*i*对项目*j*的评分 r_{ij} 可以表示为 $r_{ij} = \text{innerproduct}(u_i, v_j)$, 更一般的情况是 $r_{ij} = f(u_i, v_j)$, 这里为了介绍方便就是用 u_i 和 v_j 内积的形式表示, 这也就是矩阵分解与低维矩阵乘积拟合评分矩阵的方法。

首先假设, 用户对项目*j*的预测评分 \hat{r}_{ij} 与真实评分 r_{ij} 之间的差服从高斯分布, 基于这一假设, 可推导出目标函数如下:

$$\begin{aligned}
 & \hat{r}_{ij} \leftarrow U_i^T V_j \\
 & \Downarrow \text{误差服从高斯分布} \\
 & p(R_{ij} - U_i^T V_j | 0, \delta^2) \\
 & \Downarrow \text{等价} \\
 & p(R_{ij} | U_i^T V_j, \delta^2) \\
 & \Downarrow \\
 & p(R | U, V, \delta^2) = \prod_{i=1}^N \prod_{j=1}^M [N(R_{ij} | U_i^T V_j, \delta^2)] \\
 & \Downarrow \text{最大化} \\
 & \ln p(R | U, V, \delta^2) = -\frac{1}{2\delta^2} \sum_{i=1}^N \sum_{j=1}^M (R_{ij} - U_i^T V_j)^2 - \frac{1}{2} \left(\sum_{i=1}^N \sum_{j=1}^M (\ln \delta^2 + \ln 2\pi) \right) \\
 & \Downarrow \text{最小化} \\
 & \sum_{i=1}^N \sum_{j=1}^M (R_{ij} - U_i^T V_j)^2
 \end{aligned}$$

最后得到矩阵分解的目标函数如下:

$$L = \sum_{i=1}^N \sum_{j=1}^M (R_{ij} - U_i^T V_j)^2$$

从最终得到目标函数可以直观地理解, 预测的分值就是尽量逼近真实的已知评分值。有了目标函数之后, 下面就开始谈优化方法了, 通常的优化方法分为两种: 交叉最小二乘法 (alternative least squares) 和随机梯度下降法 (stochastic gradient descent)。

首先介绍交叉最小二乘法, 之所以交叉最小二乘法能够应用到这个目标函数主要是因为L对U和V都是凸函数。首先分别对用户因子向量和项目因子向量求偏导, 令偏导等于0求驻点, 具体解法如下:

在word中插入分页符，多出一行
yuanqita: 我尝试了1楼的方法，
发现使用段前分页也非常好使，
而且不需要转换文档，不错，有
多个解决方法。

$$\frac{\partial L}{\partial U_i} = -2 \sum_{j=1}^N \sum_{j=1}^M V_j (R_{ij} - U_i^T V_j) \quad \xRightarrow{\text{令对U的偏导等于0}} \quad U_i = \frac{\sum_{j=1}^M R_{ij} V_j}{\sum_{j=1}^M V_j V_j} \quad (1)$$

$$\frac{\partial L}{\partial V_j} = -2 \sum_{i=1}^N \sum_{i=1}^M U_i^T (R_{ij} - U_i^T V_j) \quad \xRightarrow{\text{令对V的偏导等于0}} \quad V_j = \frac{\sum_{i=1}^N R_{ij} U_i}{\sum_{i=1}^N U_i U_i} \quad (2)$$

http://blog.csdn.net/sun_168

上面就是用户因子向量和项目因子向量的更新公式，迭代更新公式即可找到可接受的局部最优解。迭代终止的条件下面会讲到。

接下来讲解随机梯度下降法，这个方法应用的最多。大致思想是让变量沿着目标函数负梯度的方向移动，直到移动到极小值点。直观的表现如下：

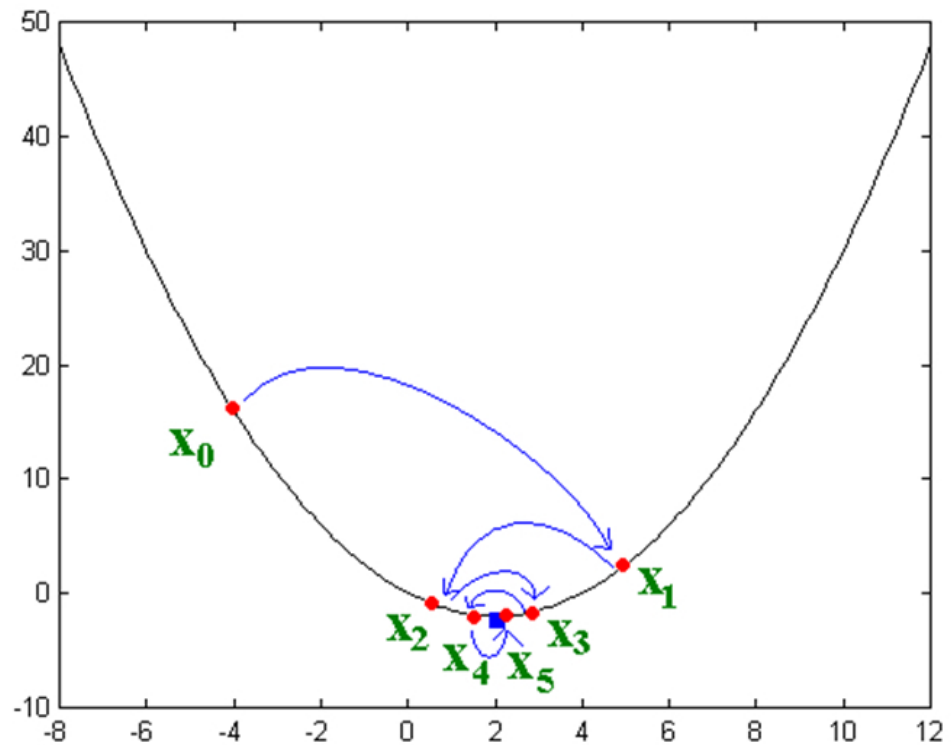


图 2 梯度下降法示意图 http://blog.csdn.net/sun_168

其实负梯度的负方向，当函数是凸函数时是函数值减小的方向走；当函数是凹函数时是往函数值增大的方向移动。而矩阵分解的目标函数L是凸函数，因此，通过梯度下降法我们能够得到目标函数L的极小值（理想情况是最小值）。

言归正传，通过上面的讲解，我们可以获取梯度下降算法的因子矩阵更新公式，具体如下：

$$U_i = U_i - \gamma \frac{\partial L}{\partial U_i} \quad (3)$$

$$V_j = V_j - \gamma \frac{\partial L}{\partial V_j} \quad (4)$$

(3) 和 (4) 中的 γ 指的是步长，也即是学习速率，它是一个超参数，需要调参确定。对于梯度见 (1) 和 (2)。

下面说下迭代终止的条件。迭代终止的条件有很多种，就目前我了解的主要有

- 1) 设置一个阈值，当L函数值小于阈值时就停止迭代，不常用
- 2) 设置一个阈值，当前后两次函数值变化绝对值小于阈值时，停止迭代
- 3) 设置固定迭代次数

另外还有一个问题，当用户-项目评分矩阵R非常稀疏时，就会出现过拟合（overfitting）的问题，过拟合问题的解决方法就是正则化（regularization）。正则化其实就是在目标函数中加上用户因子向量和项目因子向量的二范数，当然也可以加上一范数。至于加上一范数还是二范数要看具体情况，一范数会使很多因子为0，从而减小模型大小，而二范数则不会它只能使因子接近于0，而不能使其为0，关于这个的介绍可参考论文[Regression Shrinkage and Selection via the Lasso](#)。引入正则化项后目标函数变为：

$$L = \sum_{i=1}^N \sum_{j=1}^M (R_{ij} - U_i^T V_j)^2 + \lambda_1 \|U_i\|^2 + \lambda_2 \|V_j\|^2 \quad (5)$$

（5）中 λ_1 和 λ_2 是指正则项的权重，这两个值可以取一样，具体取值也需要根据数据集调参得到。优化方法和前面一样，只是梯度公式需要更新一下。

矩阵分解算法目前在推荐系统中应用非常广泛，对于使用RMSE作为评价指标的系统尤为明显，因为矩阵分解的目标就是使RMSE取值最小。但矩阵分解有其弱点，就是解释性差，不能很好为推荐结果做出解释。

后面会继续介绍矩阵分解算法的扩展性问题，就是如何加入隐反馈信息，加入时间信息等。

引用：

说明上图 图2摘自李金屏课件

上一篇 [如何把matlab画的图转化为wmf格式](#)

顶 5 踩 0

我的同类文章

数据挖掘（2）	算法（2）
<ul style="list-style-type: none">距离和相似度度量	<ul style="list-style-type: none">在线广告

主题推荐 应用

猜你在找

- | | |
|----------------------------|-----------------|
| 有趣的算法（数据结构） | 浅谈矩阵分解以及应用1 |
| 数据结构和算法 | 浅谈矩阵分解以及应用2 |
| 《C语言/C++学习指南》加密解密篇（安全相关算法） | 浅谈矩阵分解以及应用3 |
| 移动APP测试基础到进阶 | 推荐系统—矩阵分解 |
| 移动端海水技术实现 | 推荐系统中矩阵分解简介阅读感悟 |



app外包



音频矩阵



小米笔记本



小米笔记本电脑



高中矩阵公式

查看评论

5楼 姚攀2015 2015-11-03 09:22发表



赞

4楼 haizhiguang 2015-10-05 12:26发表



[引文]: 用户因子向量和项目因子向量的更新公式，迭代更新公式即可找到可接受的局部最优解。迭代终止的条件下面会讲到。

通过最小二乘法得到的 U_i 和 V_j 之后的意义是什么呢？是到了这一步就可以通过已有的信息计算矩阵中的“?”值，或者是预测用户对其他物品的评分了吗？

3楼 机智的大龙哥 2015-09-15 19:15发表



请问一下如果新来了一个user,该如何填上分数呢？

2楼 flybabala 2014-04-10 10:47发表



谢谢啦，写得详细，希望您能继续更新和扩展。

1楼 mhtlee 2014-03-15 16:34发表



写的很全面，尤其是很小的点都有提到，不错的参考

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 