

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

CSDN

博客 (//blog.csdn.net/)

下载 (//download.csdn.net?ref=toolbar)

更多 ▾

立即体验

登录 (https://passport.csdn.net/account/login?ref=toolbar)

注册 (http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

推荐系统笔记二、矩阵分解协同过滤

原创2016年01月05日 20:45:27

标签：[推荐系统](#) / [矩阵分解](#) / [协同过滤](#) / [时间信息](#)



AlexInML (http://blog.csdn.net/wangjian1204)

+ 关注

原创37

粉丝32

喜欢0

未开通码云 (https://gitee.com/wangjian1204)

- 他的最新文章
- Conda虚拟环境 (http://blog.csdn.net/wangjian1204/article/details/78508949)

深度学习目标检测之RPN-based方法 (http://blog.csdn.net/wangjian1204/article/details/78172588)

flask快速搭建tensorflow http服务 (http://blog.csdn.net/wangjian1204/article/details/76732337)

在Spark上进行两个大数据集的匹配 (http://blog.csdn.net/wangjian1204/article/details/74906887)

hadoop命令OutOfMemoryError GC (http://blog.csdn.net/wangjian1204/article/details/71732171)

- 相关推荐
- 浅谈矩阵分解在推荐系统中的应用 (http://blog.csdn.net/sun_168/article/details/20637833)

矩阵分解在协同过滤推荐算法中的应用 (http://blog.csdn.net/jianglibo1024/article/details/54799844)

推荐算法——基于矩阵分解的推荐算法 (http://blog.csdn.net/google19890102/article/details/51124556)

推荐系统-矩阵分解原理详解 (http://blog.csdn.net/u010111016/article/details/50821071)

一、概述：

矩阵分解模型是把用户偏好和item属性投影到同一个隐因子空间（latent factor space），以用户偏好和item属性的匹配程度来预测评分。通常推荐系统可以用于模型训练的信息主要有用户的显式反馈、隐式反馈和时间信息等。显式反馈（explicit feedback）：用户直接表明对item的兴趣，例如评分[1,2,3,4,5]。隐式反馈（implicit feedback）：通过观察用户行为得到的信息，例如浏览历史、搜索记录等。

二、符号定义和度量标准：

- U：用户集合；I：item集合；
- R：评分集合， r_{ui} 表示用户 $u \in U$ 对item $i \in I$ 的评分； \hat{r}_{ui} 表示预测评分；
- K：(u,i)集合，且评分 r_{ui} 已知；
- t_{ui} ：用户u对item i评分的时间；
- U_i ：已经给item i打分的用户集合； I_u ：已经被用户u打分的item集合；

评价推荐系统时最重要的两个问题是评分准确率和Top-N推荐问题。评分准确率问题常用的度量标准有Root Mean Squared Error（RMSE），Precision，Recall等。Top-N推荐常用的度量标准有NDCG，Average Reciprocal Hit-Rank（ARHR）等。

三、偏置项：

矩阵分解模型是把用户偏好和item属性投影到同一个隐因子空间（latent factor space），以用户偏好和item属性的匹配程度来预测评分。但是，仅仅这样是不够的，例如A和B两个用户的偏好类似，但是A比较容易满足，倾向于给item打高分；而B倾向于给item打低分（Item上也有类似的情况）。为了解决这个问题，需要为每个用户和item增加一个偏置项。评分 r_{ui} 的偏置部分定义为：

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

五、SVD++：

用户的**隐式反馈**可以提供额外的偏好信息，能在一定程度上提高预测准确性。例如SVD++把用户是否对item打分作为一种隐式反馈。

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^\top (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j)$$

对于用户u打分的item，增加一个隐偏好属性 $y_j \in R^f$ ，表示用户u对打分item的某种偏好。正则化项 $|I_u|^{-\frac{1}{2}}$ 用于消除用户评分个数的影响。

模型参数 b_i, b_u, q_i, p_u, y_j 通过最优化下面这个目标函数获得：

$$\min_{b_i, b_u, q_i, p_u} \sum_{(u,i) \in K} (r_{ui} - \mu - b_u - b_i - q_i^\top (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j))^2 + \lambda \left\{ \sum_u (b_u^2 + \|p_u\|^2) + \sum_i (b_i^2 + \|q_i\|^2 + \|y_i\|^2) \right\}$$

与SVD类似，可以通过梯度下降方法求解参数。

如果要加入多种隐式反馈信息，如收藏、租借等，则在用户偏好属性中加入多个隐式反馈项即可：

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^\top (p_u + |I_u^1|^{-\frac{1}{2}} \sum_{j \in I_u^1} y_j^1 + |I_u^2|^{-\frac{1}{2}} \sum_{j \in I_u^2} y_j^2 + \dots)$$

六、加入时间信息：

时间信息在推荐中有很重要的地位。一个用户对某个item的评价可能在一年后会发生很大的变化。时间信息可以告诉我们用户在某个时间点对item的喜好程度以及随着时间推移用户偏好的改变。

在SVD模型中，容易随时间发生变化的参数有： b_u 如用户打分的严格程度发生改变； b_i 如item的流行度发生改变； p_u 用户的喜好发生改变。由于item的属性往往是固定不变的，所以可以认为 q_i 是恒定的。
下面介绍 b_u, b_i, p_u 各项常用的时间函数表示。

bi项：

$$b_i^1(t) = b_i + b_{i, Bin(t)}$$

b_i 是一个常数项偏置， $b_{i, Bin(t)}$ 是一个分段函数， $Bin(t)$ 是t所在的分段编号。例如从2000年到2016年，每一年分为一段，则总共有16段，如果t是2015年的某一天，则 $Bin(t) = 16$ 。

$$b_i^2(t) = b_i + b_{i, Bin(t)} + b_{i, period(t)}$$

和 $b_i^1(t)$ 相比， $b_i^2(t)$ 增加了一个周期项 $b_{i, period(t)}$ 用来处理周期性的偏差，如羽绒服在不同季节评价的偏差。

bu项：

首先构建用户评分偏置的线性渐变模型（例如用户u的平均评分可能越来越高）：定义用户u评分的平均时间为 t_u ，那么在 t 时刻：

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

参数 β 通过交叉验证获得。

$$b_u^1(t) = b_u + \alpha_u \cdot dev_u(t)$$

式 $b_u^1(t)$ 每个用户包含两个参数 b_u, α_u 。

$$b_u^2(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\delta|t-t_l^u|} b_{tl}^u}{\sum_{l=1}^{k_u} e^{-\delta|t-t_l^u|}}$$

式 $b_u^2(t)$ 设置 k_u 个时间点 $t_1^u, \dots, t_{k_u}^u$ ，参数 δ 通过交叉验证学习得到， b_{tl}^u 从评分数据中学习。

$$b_u^3(t) = b_u + \alpha_u \cdot dev_u(t) + b_{u,t}$$

$b_u^3(t)$ 和 $b_u^1(t)$ 相比，增加了一项 $b_{u,t}$ ，用于处理每一天有可能产生的突变。例如用户u在某一天对item 的评分都很高，可能是因为他那天心情很好。

$$b_u^4(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\delta|t-t_l^u|} b_{tl}^u}{\sum_{l=1}^{k_u} e^{-\delta|t-t_l^u|}} + b_{u,t}$$

同样的， $b_u^4(t)$ 在 $b_u^2(t)$ 的基础上增加了 $b_{u,t}$ 。另外，以上这四种方法都可以添加周期项 $b_{u,period(t)}$ 来处理用户的周期性偏差。

pu项：

随着时间的改变，用户的喜好也在发生变化，所以把用户的偏好 p_u 作为一个时间函数会比较合适。
 p_u 的时间函数和用户的偏置项 b_u 类似，例如：

$$p_u(t) = p_u + \alpha_u \cdot dev_u(t) + p_{u,t}$$

有了前面三项的时间函数，timeSVD++的预测函数如下：

$$\hat{r}_{ui} = \mu + b_i(t_{ui}) + b_u(t_{ui}) + q_i^\top \{p_u(t_{ui}) + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j\}$$

最后有个问题，**预测函数中如何设定未来某一天的参数**，如 $b_{u,t}$ ， $p_{u,t}$ ？事实上，在训练模型中加入局部时间项主要是为了获得更好的稳定项（如 $b_u + \alpha_u \cdot dev_u(t)$ ），所以在预测过程中可以忽略局部时间项（同时这个也无法计算得到），用稳定项来预测评分。

七、参考资料

Recommender Systems Handbook

版权声明：本文为博主原创文章，如需转载请在文章开头标明原始地址并联系博主。



相关文章推荐

浅谈矩阵分解在推荐系统中的应用 (http://blog.csdn.net/sun_168/article/details/206378...)

矩阵分解算法在推荐系统利用的应用，详细介绍应用方法，及目标函数的推导过程。...

sun_168 (http://blog.csdn.net/sun_168) 2014年03月14日 16:28 37360

矩阵分解在协同过滤推荐算法中的应用 (<http://blog.csdn.net/jianglibo1024/article/detai...>)

在协同过滤推荐算法总结中，我们讲到了用矩阵分解做协同过滤是广泛使用的方法，这里就对矩阵分解在协同过滤推荐算法中的应用做一个总结。(过年前最后一篇！祝大家新年快乐！明年的目标是写120篇机器学习，深度学习...

jianglibo1024 (<http://blog.csdn.net/jianglibo1024>) 2017年01月31日 18:38 1347



【前端逆袭记】我是怎么从月薪4k到40k的！

谨以此篇文章献给我奋斗过的程序人生！我第一次编码是在我大一的时候....

(http://www.baidu.com/cb.php?c=IgF_pyfqHmknj0dP1f0IZ0qnfK9ujYzP1ndPWb10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1Y4uH-9uWPWPWDkrHRdPHT40AwY5HDdnHnvrHcvrjn0IgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEIAqspynElvNBnHqdIAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPH0vP1T)

推荐算法——基于矩阵分解的推荐算法 (<http://blog.csdn.net/google19890102/article/d...>)

一、推荐算法概述对于推荐系统(Recommend System, RS)，从广义上的理解为：为用户(User)推荐相关的商品(Items)。常用的推荐算法主要有：基于内容的推荐(Content-Ba...

google19890102 (<http://blog.csdn.net/google19890102>) 2016年04月12日 17:07 12146

推荐系统-矩阵分解原理详解 (<http://blog.csdn.net/u010111016/article/details/508210...>)

目前推荐系统中用的最多的就是矩阵分解方法，在Netflix Prize推荐系统大赛中取得突出效果。以用户-项目评分矩阵为例，矩阵分解就是预测出评分矩阵中的缺失值，然后根据预测值以某种方式向用户推荐。常...

u010111016 (<http://blog.csdn.net/u010111016>) 2016年03月07日 17:44 8412

矩阵分解 (MATRIX FACTORIZATION) 在推荐系统中的应用 (<http://blog.csdn.net/lissan...>)

前言 最近一段时间隐语义模型 (Latent Factor Model , LFM) 在推荐系统中的应用越来越广泛，本文所介绍的矩阵分解方法也是基于这个隐语义模型。这里需要说明的一点是，这里...

lissanwen (<http://blog.csdn.net/lissanwen>) 2016年04月22日 00:16 2350

智齿是什么

牙齿有洞怎么办



亦庄写字楼出租

牙科价目表

it培训机构排名



使用LFM (Latent factor model) 隐语义模型进行Top-N推荐 (<http://blog.csdn.net/HarryHuang1990>)

最近在拜读项亮博士的《推荐系统实践》，系统的学习一下推荐系统的相关知识。今天学习了其中的隐语义模型在Top-N推荐中的应用，在此做一个总结。隐语义模型LFM和LSI, LDA, Topic Model...

 HarryHuang1990 (<http://blog.csdn.net/HarryHuang1990>) 2013年08月12日 14:30  22485

【机器学习】LFM (Latent Factor Model) (<http://blog.csdn.net/scecit/article/details...>)

LFM (Latent Factor Model) 参考了[Key_Ky博客](%28<http://www.cnblogs.com/Key-Ky/p/3579363.html>%29)的潜在矩阵分解的代码...

 scecit (<http://blog.csdn.net/scecit>) 2015年04月20日 19:41  1228



[推荐算法]ItemCF,基于物品的协同过滤算法 (<http://blog.csdn.net/yeruby/article/details...>)

ItemCF : ItemCollaborationFilter, 基于物品的协同过滤 算法核心思想 : 给用户推荐那些和他们之前喜欢的物品相似的物品。比如, 用户A之前买过《数据挖掘导论》, 该算法会根据此...

 yeruby (<http://blog.csdn.net/yeruby>) 2015年03月09日 15:11  22254



个性化推荐中的矩阵分解技术 (<http://blog.csdn.net/winone361/article/details/5070575...>)

<http://baogege.info/2014/10/19/matrix-factorization-in-recommender-systems/> 引言 随着Netflix P...

 winone361 (<http://blog.csdn.net/winone361>) 2016年02月20日 19:03  2640



机器学习中的矩阵分解方法 (<http://blog.csdn.net/u011081315/article/details/76252524>)

基于郭栋老师的教学PPT, 配上相关paper和资料, 做到对矩阵分解技术有个大致了解。 一个假设 : 数据由有限的潜在因子决定, 数据样本的观测值是潜在因子的一个映射。 矩阵分解发展历史 : 经典的...

 u011081315 (<http://blog.csdn.net/u011081315>) 2017年07月28日 13:50  191



推荐算法——基于上下文的推荐 (<http://blog.csdn.net/u011060119/article/details/7629...>)

将时间信息和地点信息建模到推荐算法中, 从而让推荐系统能够准确的预测用户在某个特定时刻以及特定地点的兴趣。本文中主要考虑时间因素 : 在给定时间信息后, 推荐系统从一个静态系统变成了一个时变的系统, 用户的...

 u011060119 (<http://blog.csdn.net/u011060119>) 2017年07月29日 10:09  441



上下文感知推荐系统 (http://blog.csdn.net/qq_17116557/article/details/51356412)

在推荐系统领域, 人们往往只关注“用户-项目”之间的关联关系, 而较少考虑它们所处的上下文环境 (如时间、位置、周围人员、情绪、活动状态、网络条件等等)。但是在许多应用场景下, 仅仅依靠“用户-项目”二元关系...

 qq_17116557 (http://blog.csdn.net/qq_17116557) 2016年05月09日 21:29  2868

推荐系统_全 (<http://blog.csdn.net/u011263983/article/details/51549693>)

推荐系统的分类 常见评测标准 <http://blog.csdn.net/u011263983/article/details/51544495> 相似度 1) 同现相似度 2) 欧氏距离相似度 3) 余...

 u011263983 (<http://blog.csdn.net/u011263983>) 2016年05月31日 22:26  5309



矩阵分解模型 (1) : ALS学习算法 (<http://blog.csdn.net/oucpowerman/article/details/...>)

一、矩阵分解模型。 用户对物品的打分行为可以表示成一个评分矩阵 $A(m \times n)$, 表示 m 个用户对 n 各物品的打分情况。如下图所示 : 其中, $A(i,j)$ 表示用户 $user_i$ 对物品 $item_j$ 的打分。 ...

 oucpowerman (<http://blog.csdn.net/oucpowerman>) 2015年11月15日 11:42  11411



ALS矩阵分解推荐模型 (<http://blog.csdn.net/linger2012liu/article/details/44085913>)

ALS矩阵分解推荐模型 其实通过模型来预测一个user对一个item的评分，思想类似线性回归做预测，大致如下 定义一个预测模型（数学公式），然后确定一个损失函数，将已有数据作为训练集，不断...

 linger2012liu (<http://blog.csdn.net/linger2012liu>) 2015年03月05日 21:08  7729



基于Spark mllib的矩阵分解(ALS)推荐算法 (<http://blog.csdn.net/nihaoxiaocui/article/d...>)

推荐引擎背后的想法是：预测人们可能喜好的物品并通过探寻物品之间的联系来辅助这个过程。推荐引擎很适合如下两类常见场景（两者可兼有）。 - 可选项众多：可选的物品越多，用户就越难找到想要的物品。如...

 nihaoxiaocui (<http://blog.csdn.net/nihaoxiaocui>) 2016年08月31日 20:22  1862



从item-base到svd再到rbm，多种Collaborative Filtering(协同过滤算法)从原理到实现 (htt...

0.说明 本文的所有代码均可在 DML 找到，欢迎点星星。 一.引入 推荐系统（主要是CF）是我在参加百度的电影推荐算法比赛的时候才临时学的，虽然没拿什么奖，但是...

 Dark_Scope (http://blog.csdn.net/Dark_Scope) 2013年12月14日 19:49  46579



矩阵分解的隐因子模型 (http://blog.csdn.net/zdy0_2004/article/details/49005591)

矩阵分解的隐因子模型 <http://www.cnblogs.com/yangxiao99/p/4864350.html> 推荐系统是现今广泛运用的一种数据分析方法。常见的如，“你...

 zdy0_2004 (http://blog.csdn.net/zdy0_2004) 2015年10月09日 20:40  1756



基于矩阵分解的隐因子模型 (<http://blog.csdn.net/ice110956/article/details/21955461>)

推荐系统是现今广泛运用的一种数据分析方法。常见的如，“你关注的人也关注他”，“喜欢这个物品的用户还喜欢。。” “你也许会喜欢”等等。常见的推荐系统分为基于内容的推荐与基于历史记录的推荐。 ...

 ice110956 (<http://blog.csdn.net/ice110956>) 2014年03月24日 14:55  3609

推荐系统评价：NDCG方法概述 (<http://blog.csdn.net/u010670689/article/details/7319...>)

摘要：哪一种模型更适合挖掘信息，主要的决策因子是推荐质量，而推荐系统包括很多很棘手的问题，下面就由这篇文章带大家推荐系统评价。【编者按】在信息过剩的互联网时代，推荐系统的地位随着大数据的...

 u010670689 (<http://blog.csdn.net/u010670689>) 2017年06月13日 22:24  289