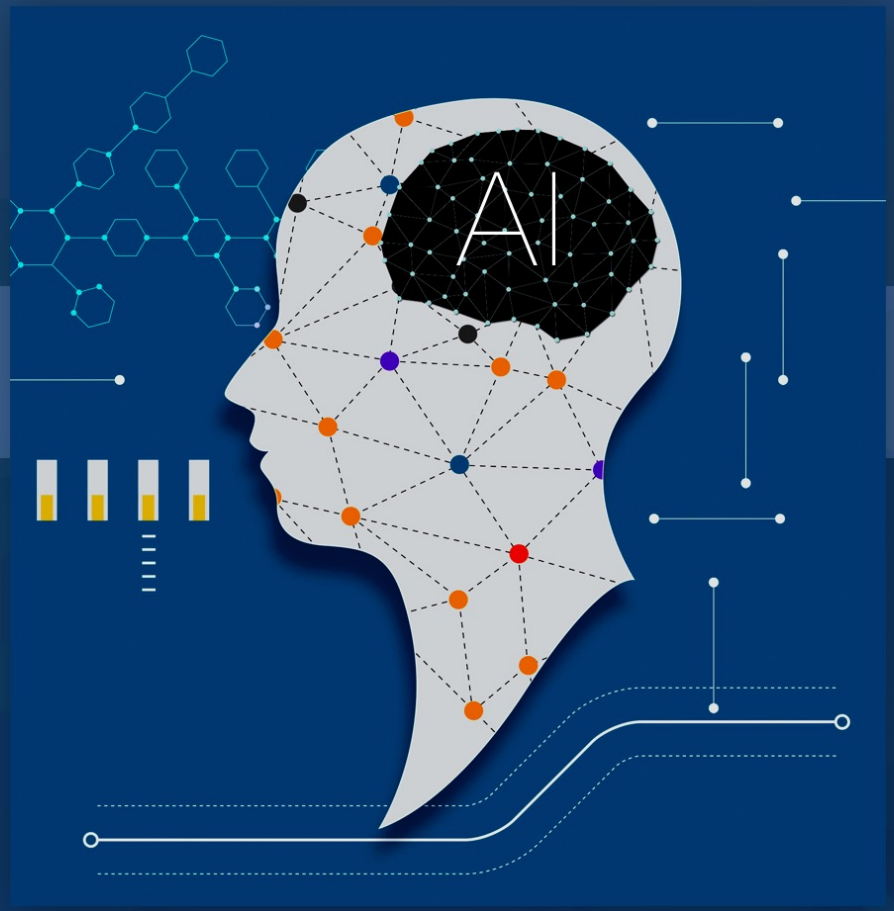


贪心科技
让每个人享受个性化教育服务

推荐系统 矩阵分解

推荐系统是什么

- 信息处理的系统, 用于预测用户对物品的偏好.
- 推荐系统可以应用在诸多领域:
  - 推荐个性化的新闻, 视频, 音乐, 图书, 游戏, 食品, 衣物以及其它各种商品
  - 在社交网络里面为你推荐可能会志趣相投的朋友
  - 在金融市场为您推荐合适的股票, 基金证券等理财产品
  - 婚恋市场里为单身朋友推荐合适的另一半

推荐系统已经成为许多网站和手机App不可或缺的一部分, 它们依靠推荐系统提高商品销量, 吸引用户注意力, 提高用户活跃度, 吸引新用户.

Not Secure | www.sohu.com/a/206943279_499203

搜狐 | 新闻 体育 汽车 房产 旅游 教育 时尚 科技 财经 娱

金梧桐投资

790 文章 | 174万 总阅读

查看TA的文章>

分享到

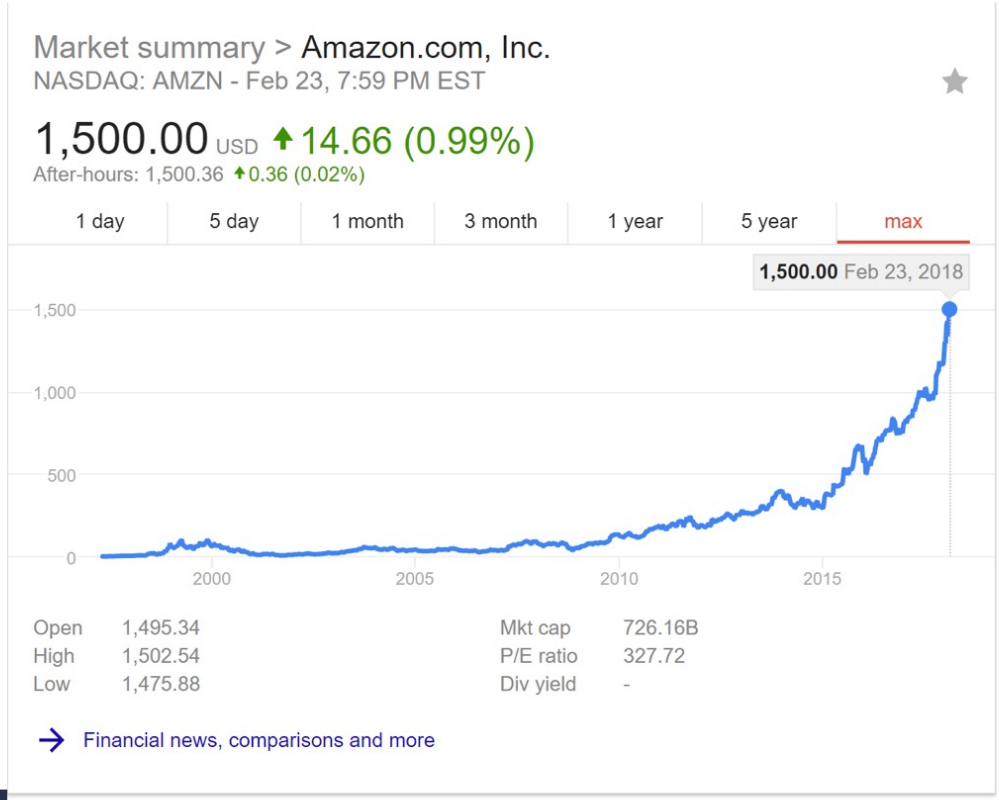
世界新首富Amazon CEO Jeff Bezos，身家突破千亿美元

2017-11-27 18:26

股价 / 创客 / 运营

上周五是美国Black Friday黑五购物节，Amazon股价一路走高，市值已超过5700亿美元，其创始人和CEO Jeff Bezos 身家也首次超过1000亿美元，成为全球首富。

亚马逊的推荐系统


Market summary > Amazon.com, Inc.
NASDAQ: AMZN - Feb 23, 7:59 PM EST
1,500.00 USD ↑14.66 (0.99%)
After-hours: 1,500.36 ↑0.36 (0.02%)
1 day
5 day
1 month
3 month
1 year
5 year
max
1,500.00 Feb 23, 2018
1,500
1,000
500
0
2000
2005
2010
2015
Open 1,495.34
High 1,502.54
Low 1,475.88
Mkt cap 726.16B
P/E ratio 327.72
Div yield -
Financial news, comparisons and more
Google Finance - Yahoo Finance - MSN Money
Disclaimer

亚马逊的推荐系统

fortune.com/2012/07/30/amazons-recommendation-secret/

Bookmarks | Zend Debugger & Xd | Start debugger | Stop debugger | Debug this page | Mail - yuan76@live.c | Cisco Secure Desktop

FORTUNE

Amazon's recommendation secret

Judging by Amazon's success, the recommendation system works. The company reported a 29% sales increase to $12.83 billion during its second fiscal quarter, up from $9.9 billion during the same time last year. A lot of that growth arguably has to do with the way Amazon has integrated recommendations into nearly every part of the purchasing process from product discovery to checkout. Go to Amazon.com and you'll find multiple panes of product suggestions; navigate to a particular product page and you'll see areas plugging items "Frequently Bought

亚马逊的推荐系统

fortune.com/2012/07/30/a...

Apps Bookmarks Zend Debugger & Xd Start debugger Stop debugger Debug this page » Other bookma

SUBSCRIBE

The tactic prevents email inboxes from being flooded, at least by Amazon. At the same time it maximizes the purchase opportunity. In fact, the conversion rate and efficiency of such emails are “very high,” significantly more effective than on-site recommendations. According to Sucharita Mulpuru, a Forrester analyst, Amazon’s conversion to sales of on-site recommendations could be as high as 60% in some cases based off the performance of other e-commerce sites.

- 根据美国财富杂志的报道， Amazon的销售额高速增长，得益于它将系统整合进用户购买， 从产品发现到付款的整个流程。根据华尔街分析师的估计，Amazon的在线推荐系统的购买转化率高达60%（http://fortune.com/2012/07/30/amazons-recommendation-secret/）。

推荐系统类型

1. 人类手工生成的推荐系统

早期的门户网站, 里面的内容都是由网站编辑手工选择的, 这也是一种原始的推荐系统

2. 简单的聚合推荐系统

- 例如KTV里面的歌曲点播排行榜, 畅销书排行榜, 电影票房排行榜
- 按照物品的时间性质推荐, 例如最近上架的新品推荐

3. 真正的个性化, 千人千面的推荐系统(本课程的关注点)

Amazon的商品推荐, Netflix的电影推荐

推荐系统模型

- U: 所有的用户集合
- P: 所有的物品集合

- 推荐系统的模型$U \times P \rightarrow R$
- R: 用户对物品的喜爱程度, 许多信息系统喜欢使用1至5分的评价值来表达, 也可以使用喜欢(点赞👍)或者不喜欢(鄙视)来表达.

推荐值矩阵

	大话西游	夏洛特烦恼	黄飞鸿	笑傲江湖
小明		2	5	4
小李	1	1		3
韩梅梅	5	5		2
小静	4		1	1

用户属性

	性别	年龄
小明	男	27
小李	男	21
韩梅梅	女	18
小静	女	23

物品属性

	武侠	动作	喜剧	文艺	悬疑	爱情
大话西游	1	0	1	0	0	1
夏洛特烦恼	0	0	1	0	0	1
黄飞鸿	0	1	0	0	0	0
笑傲江湖	1	1	0	0	0	0

推荐系统的关键问题

1. 收集数据, 建立推荐值矩阵
   - 数据具有时效性. 一个用户的喜好会睡着时间的变化而变化. 一个人童年喜欢看的书籍和电影在成年后可能不再喜欢了. 一个刚刚有小孩出生的用户喜欢的商品会出现更多的婴儿用品, 例如尿不湿.

2. 从推荐值矩阵中已知数据预测未知的数据
   - 推荐系统的核心功能

3. 建立一个评价体系, 用于检验推荐系统的推荐效果
   - 如何知道你的推荐系统是不是一个好的推荐系统

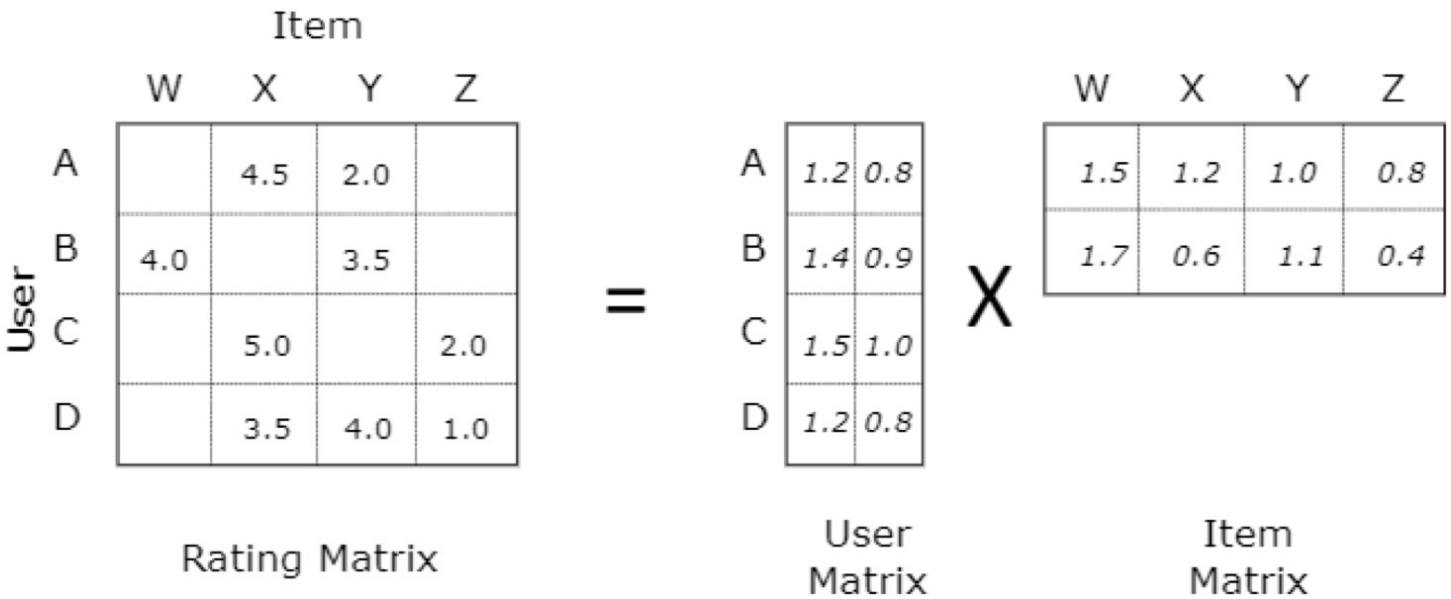
1. 收集数据

- 显式收集
  - 请用户为物品打分, 点赞和评价
  - 挑战在于数据经常不完整, 许多用户可能从来不, 或者很少主动通过打分, 点赞或者留言的方式表达自己的喜好
- 隐性收集
  - 有的用户对某个物品的喜爱可能是隐形的, 可以通过用户行为来推断用户的喜爱度.
    - 视频网站：用户的行为特征, 快进, 重播
    - 购物网站：购买行为表达了用户的喜爱, 退货行为表达了不喜爱

2. 从推荐值矩阵中已知数据预测未知的数据

- 关键的挑战:
  - 当用户和物品数量都比较大的情况下, 推荐值矩阵通常是一个稀疏矩阵, 矩阵中大多数值是未知的. 大多数用户很可能没有为大多数物品表达喜好
  - 冷启动问题:
    - 新的物品还没有任何用户为它打过分
    - 新的用户没有任何的打分行为

常用推荐方法:矩阵分解


Item
W X Y Z
User
A 4.5 2.0
B 4.0 3.5
C 5.0 2.0
D 3.5 4.0 1.0
Rating Matrix
=
A 1.2 0.8
B 1.4 0.9
C 1.5 1.0
D 1.2 0.8
User Matrix
X
W X Y Z
1.5 1.2 1.0 0.8
1.7 0.6 1.1 0.4
Item Matrix

基于矩阵分解的推荐系统

- 这个算法的基本思想:
  - 将推荐值矩阵R分解为矩阵U 和 P , 使得 U 和 P 的乘积得到的矩阵R*中的元素值与R的中已知元素的值非常接近.
  - 那么R*中对应于R中的未知的元素值就是预测值.

基于矩阵分解的推荐系统

物品

用户

$$\underset{R}{\begin{pmatrix} 1 & 3 & 5 & 4 \\ & 2 & & 4 \\ 3 & 4 & 3 & \end{pmatrix}} \approx \underset{U}{\begin{pmatrix} -0.77 & -1.84 \\ -0.2 & -1.85 \\ -1.98 & -0.54 \end{pmatrix}} \underset{P^T}{\begin{pmatrix} -1.46 & -1.67 & -0.88 & -0.32 \\ 0.04 & -0.89 & -2.3 & -2.04 \end{pmatrix}} = \underset{R^*}{\begin{pmatrix} 1.06 & 2.93 & 4.9 & 4 \\ 0.21 & 1.97 & 4.41 & 3.84 \\ 2.88 & 3.79 & 2.98 & 1.73 \end{pmatrix}}$$

预测值

基于矩阵分解的推荐系统：原理

- 推荐值矩阵 R 里有一些值是已知的, 如果我们将 R 分解为两个矩阵 U 和 P, U的每一行代表一个用户画像向量, P 的每一行代表一个物品的画像向量.
- 如果 U 和 P 能够分别准备表达用户和物品画像向量, 那么 U 和 P^{T} 的乘积 R* 中的值就应该是推荐值.

- 假设用户对物品的喜好的值是取决于几个(假设是 k 个)因素, 我们不知道这些因素是什么, 所以我们命名它们为隐性因子.
- 假设用户 U_i 对物品 $\mathrm{P_j}$ 的喜好 R_{ij} 是用户画像向量在 k 个因子上的值与物品画像向量在 k 个因子上的值的点积 $R_{ij} = \mathrm{U}_i \cdot \mathrm{P_j}$
- 如果能够通过分解推荐值矩阵, 使得以上的假设在所有已知的推荐值上面是成立的, 我们期望这个方式也可以用于预测未知的推荐值

基于矩阵分解的推荐系统

物品

用户

- $$\begin{pmatrix} 1 & 3 & 5 & 4 \\ & 2 & & 4 \\ 3 & 4 & 3 & \end{pmatrix} \approx \begin{pmatrix} -0.77 & -1.84 \\ -0.2 & -1.85 \\ -1.98 & -0.54 \end{pmatrix} \begin{pmatrix} -1.46 & -1.67 & -0.88 & -0.32 \\ 0.04 & -0.89 & -2.3 & -2.04 \end{pmatrix} = \begin{pmatrix} 1.06 & 2.93 & 4.9 & 4 \\ 0.21 & 1.97 & 4.41 & 3.84 \\ 2.88 & 3.79 & 2.98 & 1.73 \end{pmatrix}$$
- R $\quad U \quad P^T \quad \mathrm{R}^*$
- $4 \approx \boldsymbol{U}_2\boldsymbol{P}_4^{\boldsymbol{T}} = (-0.2 \quad -1.85)\begin{pmatrix} -0.32 \\ -2.04 \end{pmatrix} = (-0.2*-0.32+-1.85*-0.204)=3.84$
- $R_{2,4}$:第2个用户对第4个物品的推荐值=4
- $\boldsymbol{U}_2$:第2个用户的用户画像向量=(−0.2, −1.85)
- $\boldsymbol{P}_4$:第4个物品的物品画像向量=(-0.32, -2.04)
- $\mathrm{R}^*_{2,4}$:第2个用户对第4个物品的推荐值的预测=3.84

基于矩阵分解的推荐系统

物品

用户

$$\underset{R}{\begin{pmatrix} 1 & 3 & 5 & 4 \\ & 2 & ? & 4 \\ 3 & 4 & 3 & \end{pmatrix}} \approx \underset{U}{\begin{pmatrix} -0.77 & -1.84 \\ -0.2 & -1.85 \\ -1.98 & -0.54 \end{pmatrix}} \underset{P^T}{\begin{pmatrix} -1.46 & -1.67 & -0.88 & -0.32 \\ 0.04 & -0.89 & -2.3 & -2.04 \end{pmatrix}} = \underset{R^*}{\begin{pmatrix} 1.06 & 2.93 & 4.9 & 4 \\ 0.21 & 1.97 & 4.41 & 3.84 \\ 2.88 & 3.79 & 2.98 & 1.73 \end{pmatrix}}$$

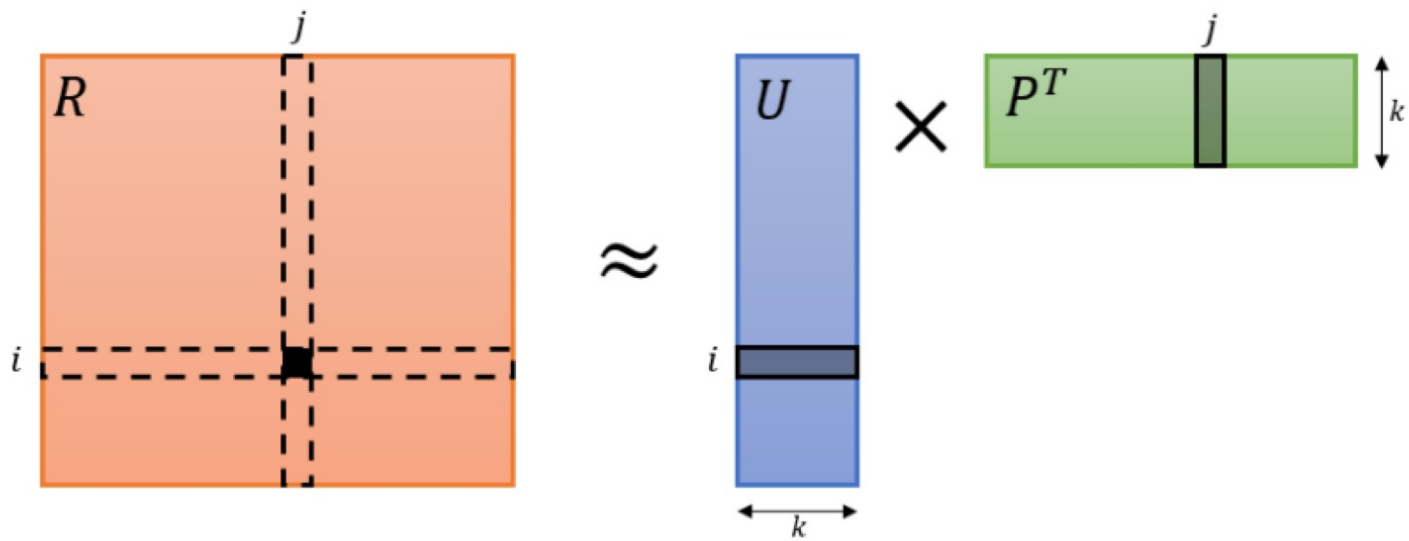
- $? \approx \boldsymbol{U}_2 \boldsymbol{P}_3^T = (-0.2 \quad -1.85)\begin{pmatrix} -0.88 \\ -2.3 \end{pmatrix} = (-0.2*-0.88+-1.85*-2.3)=4.41$
- $R_{2,3}$:第2个用户对第3个物品的推荐值=未知
- $\boldsymbol{U}_2$: 第2个用户的用户画像向量=(−0.2, −1.85)
- $\boldsymbol{P}_3$: 第3个物品的物品画像向量=(-0.88, -2.3)
- $R_{2,3}^*$:第2个用户对第3个物品的推荐值的预测=4.41

基于矩阵分解的推荐系统:矩阵分解

- $\arg\min_{\boldsymbol{U},\boldsymbol{P}} \sum_{(i,j)\in Z}\left(R_{ij} - \boldsymbol{U}_i \boldsymbol{P}_j^T\right)^2$
- $Z = \{(i,j): r_{ij}\text{已知}\}$
- $\boldsymbol{U}_i$,$\boldsymbol{P}_j$ 为行向量, 分别来自于矩阵 U 和矩阵 P 的第 i 行和第 j 行; 分别代表了第 i 个用户的画像向量, 和第 j 个物品的画像向量.

算法假设

- 评分(Rating)矩阵 $\mathrm{R} \in R^{m\times n}$ 包含了 m 个用户(user) 对 n 个物品的评分. 假设评分矩阵可以分解为用户矩阵 $\mathrm{U} \in R^{m\times k}$ 和 $\mathrm{P} \in R^{n\times k}$, 使得 $\mathrm{R} \approx U\times \mathrm{P}^{\mathrm{T}}$



算法假设(续)

- 评分(Rating)矩阵 $\mathrm{R} \in R^{m \times n}$ 包含了 m 个用户(user) 对 n 个物品的评分. 假设评分矩阵可以分解为用户矩阵 $\mathrm{U} \in R^{m \times k}$ 和 $\mathrm{P} \in R^{n \times k}$, 使得 $\mathrm{R} \approx U \times P^T$
- 其中 k 是矩阵分解的秩, 决定了 U, P 的维度
- $R_{ij} \approx U_i \cdot P_j$
- R_{ij} 为评分(Rating)矩阵第 i 行, 第 j 列的元素
- U_i 为用户矩阵第 i 行的向量
- P_j 为物品矩阵第 j 行的向量
- 假设一个评分是由 用户和物品 的 k 个影响因子决定的.

算法理解

- $R_{ij} \approx U_i \cdot P_j$
- 使用电影举例, 一部电影是有多个演员, 导演, 风格, 故事情节等构成, 任何一个观众都对不同的演员, 导演, 风格, 故事情节有不同的喜好.
- 如果一部电影的演员,导演,风格,故事情节与某一个观众喜好的演员,导演,风格,故事情节比较吻合(类似), 那么理论上观众的打分就会高.
- 通过矩阵分解, 我们并没有显式地找出电影的演员, 导演, 风格, 故事情节, 但是我们假设U_i就是代表了第 i 个用户 k 个这种隐含的喜好特征, P_j就是代表了第 j 个电影的 k 个这种隐含特征.

损失函数

$$\arg\min_{\boldsymbol{U},\boldsymbol{P}} \sum_{(i,j)\in Z} \frac{1}{2}\left(R_{ij} - U_i \cdot P_j\right)^2$$

$$Z = \{(i,j): r_{ij}\text{已知}\}$$

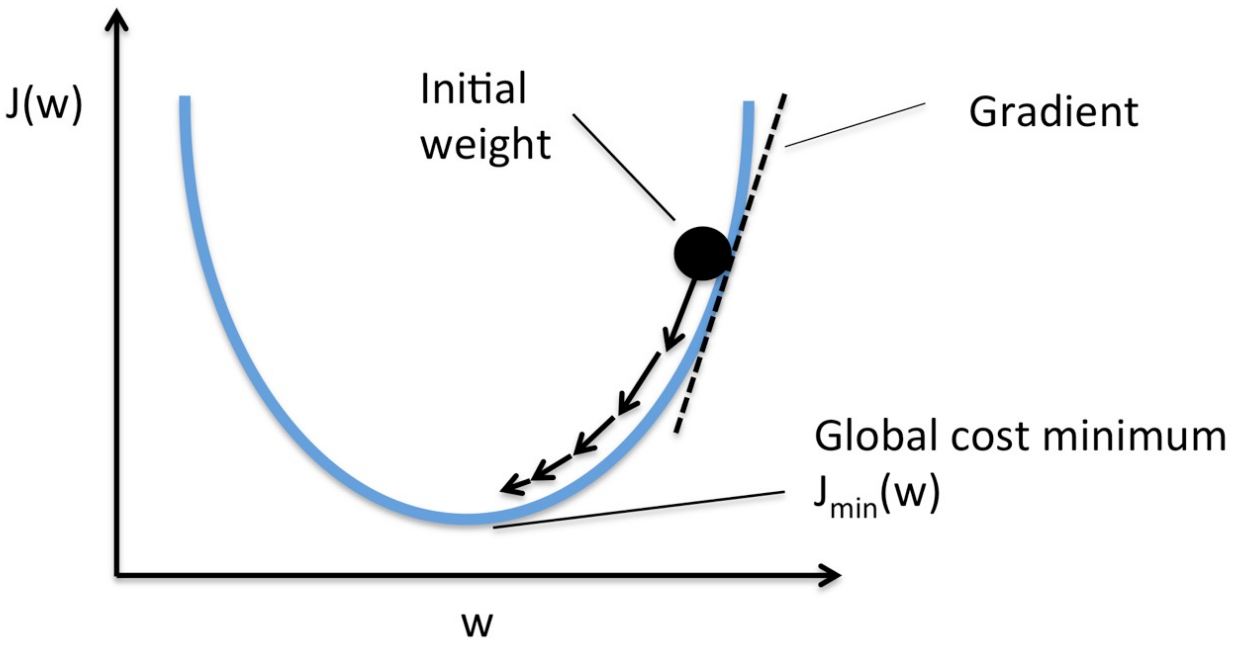
$$L_{ij} = \frac{1}{2}\left(R_{ij} - U_i \cdot P_j\right)^2$$

损失函数的梯度

$$\frac{\partial L_{ij}}{\partial U_i} = \frac{\partial}{\partial U_i}\left[\frac{1}{2}\left(R_{ij} - U_i \cdot P_j\right)^2\right] = -P_j\left(R_{ij} - U_i \cdot P_j\right)$$

$$\frac{\partial L_{ij}}{\partial P_j} = \frac{\partial}{\partial P_j}\left[\frac{1}{2}\left(R_{ij} - U_i \cdot P_j\right)^2\right] = -U_i\left(R_{ij} - U_i \cdot P_j\right)$$

Gradient Descent 梯度下降法

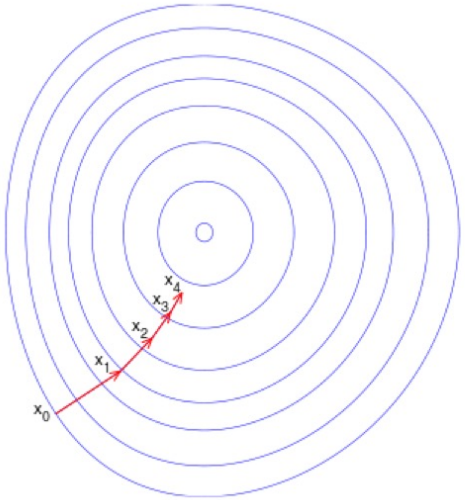


基于矩阵分解的推荐系统：加入正则化

$$\arg\min_{\boldsymbol{U},\boldsymbol{P}} \sum_{(i,j)\in Z} \frac{1}{2}\left(R_{ij} - U_i \cdot P_j\right)^2 + \lambda\left[\sum_{i=1}^{m}\|U_i\|^2 + \sum_{i=1}^{n}\|P_j\|^2\right]$$

- λ >0是正则化参数
- 添加正则化$\lambda\left[\sum_{i=1}^{m}\|U_i\|^2 + \sum_{i=1}^{n}\|P_j\|^2\right]$的目的是防止过拟合(在训练集上误差小, 在测试集上误差大)

基于矩阵分解的推荐系统
梯度下降法求解最小化问题



- 设定 k 的值, 设定学习步长γ(learning rate),初始化 U 和 P
- 重复以下步骤直到均方差满意为止:
  - 遍历Z中的(i,j), $Z=\{(i,j): r_{ij}\text{已知}\}$
  - $U_i \leftarrow U_i - \gamma \frac{\partial L_{ij}}{\partial U_i}$
  - $P_j \leftarrow P_j - \gamma \frac{\partial L_{ij}}{\partial P_j}$

$$\frac{\partial L_{ij}}{\partial U_i} = -P_j\left(R_{ij} - U_i \cdot P_j\right) + \lambda U_i$$

$$\frac{\partial L_{ij}}{\partial P_j} = -U_i\left(R_{ij} - U_i \cdot P_j\right) + \lambda P_j$$

基于矩阵分解的推荐系统：关于 k

- 假设推荐系统里面有m个用户, n个物品, 那么推荐值矩阵的大小为m行n列
- 矩阵分解以后, 用户矩阵 U 的大小为m行k列, 物品矩阵 P 的大小为n行k列
- k 决定了 U 和 P 的列数.
- 从原理上讲, k是我们直觉上认为有多少个隐含的因素决定了一个用户为什么要喜好某个物品.
- 在现实中, 我们无法直接推导出合适的k值, 我们一般采用交叉验证(cross validation), 通过实验, 测试不同的k值产生的平均绝对离差来找到合适的k值.

基于矩阵分解的推荐系统：优点

- 相对于协同过滤KNN, 矩阵分解后预测一个值的计算量比较小. 预测的计算就是求两个向量的点积, 预测时的计算复杂度和用户以及物品数量无关.

- 相对于基于内容的推荐系统, 矩阵分解很好的利用了其它用户对物品打分的数据.

基于矩阵分解的推荐系统：缺点

- 类似于协同过滤KNN, 矩阵分解也具有冷启动问题, 对于一个新用户, 或者一个新物品, 因为没有相关的喜好数据, 无法做出推荐.

- 假设了用户画像向量和物品画像向量是通过点积得到推荐值, 相当于默认了是线性的关系, 事实上可能是更加复杂的非线性关系.

评估推荐系统

1. 离线评估
2. 问卷调查
   - 同一个问题，两种形式
   - 所言非所意
3. 用户学习（User Study）
   - 不只是算法，针对整个系统，包括用户界面
   - 几十人用户的小范围测试能够发现90%左右的大问题
4. 在线测试 （A/B 测试）

评估推荐系统：离线测试

- 给定一个推荐系统数据集合, 可以把这个数据集合按照 3/7 开分为测试数据集合 T 和训练数据集合 U . 可以使用 U 来训练一个推荐系统, 然后在测试集 T 上面做出预测, 比较预测值和真实的值.
- 平均绝对离差

$$\text{(Mean-Absolute-Error)} = \frac{\sum_{(u,p)\in T} |r_{up} - r^*_{up}|}{N}$$

- N : 测试集中推荐值的总数量
- r_{up}:真实的用户$\boldsymbol{u}$对物品$\boldsymbol{p}$的推荐值
- r^*_{up}:预测的用户$\boldsymbol{u}$对物品$\boldsymbol{p}$的推荐值

物品

用户

1	3	5	4		5
	2		4	3	
3	4	3		3	
	3	3	?		?
3	2	2	?		
1	1			?	?

上图显示了一个推荐值矩阵数据集合, 其中橙色部分为训练数据集合,灰色部分为测试数据集合

评估推荐系统：使用平均绝对离差的问题

- 忽略了如下情况:
- 1. 预测的多样性
  - 以新闻网站为例, 如果一个用户最喜欢的新闻是体育新闻, 难道我们就只应该给此用户推荐体育新闻吗? 也许不会有用户喜欢自己的新闻App只出现体育新闻. 偶尔加入一些金融, 八卦也许更合理.
- 2. 预测的上下文
  - 对推荐值的预测需要根据用户当前的行为调整. 例如一用户相对于<<甄嬛传>>更喜欢<<纸牌屋>>, 但是如果用户正在看<<甄嬛传>>的21集, 在这个上下文里面需要给<<甄嬛传>>22集的推荐值高于 <<纸牌屋>>.
- 3. 预测的结果排序
  - 预测的物品在呈现给用户的时候总是需要排序的.平均绝对离差是没有考虑排序的情况. 实际上, 我们可能最关注的是那些推荐值比较高的情况. 因为物品太多了, 我们可能只会给用户显示前10个推荐值比较大的物品.
  - 如果一个系统在前10个推荐值大的物品的情况下很准确, 而对其余1万个推荐值小的物品情况下不太准确, 使用平均绝对离差这种评价方法可能会认为这个系统不好, 但是真实世界里面这种系统很可能是我们喜欢的好系统
  - 另一种方式: 只考虑排名靠前的k个预测值的误差

评估推荐系统（2）

- 并不需要预测一个绝对的推荐值, 更在意推荐的物品的排序

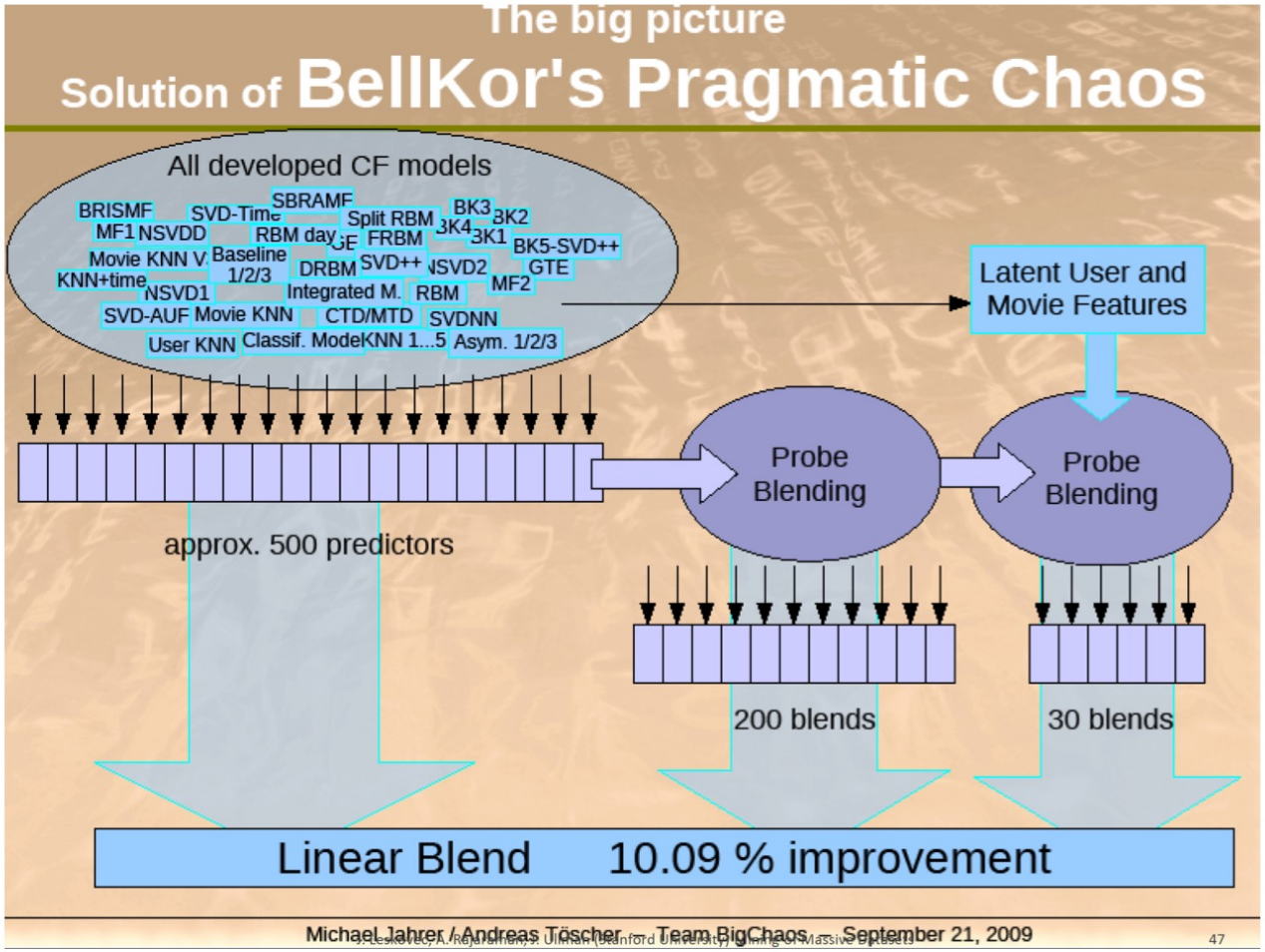
- 对于测试集中的物品, 两两配对, 如果预测值的大小比较与实际打分大小顺序一样, 则认为是“正确”的预测, 否则是“错误的”. 最后“正确”的预测占的比例越高则推荐越准确.

总结

- ”No free lunch theorem”: 世界上没有一种算法在解决任何问题的情况下都优于另外一种算法.

- 组合多种不同算法. 获得Netflix推荐系统比赛100万美元奖金的冠军团队就是采用了这种方法. 参见: http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/

模型集成


The big picture
Solution of BellKor's Pragmatic Chaos
All developed CF models
BRISMF
SVD-Time
SBRAMF
Split RBM
BK3
BK2
MF1
NSVDD
RBM day
FRBM
BK5-SVD++
Movie KNN V.
Baseline 1/2/3
DRBM
SVD++
NSVD2
GTE
KNN+time
NSVD1
Integrated M.
RBM
MF2
SVD-AUF
Movie KNN
CTD/MTD
SVDNN
User KNN
Classif. Mode
KNN 1...5
Asym. 1/2/3
Latent User and Movie Features
Probe Blending
Probe Blending
approx. 500 predictors
200 blends
30 blends
Linear Blend 10.09 % improvement
Michael Jahrer / Andreas Töscher — Team BigChaos — September 21, 2009
47

代码演示

贪心科技 GREEDY TECHNOLOGY

贪心科技　让每个人享受个性化教育服务

THANKS

贪心学院讲师：袁源