推荐技术之FM算法

该算法08年火起来的，获得NetFlix8万美金的大奖



Background：线性代数

矩阵的秩：矩阵的秩是矩阵中线性无关的行或列的个数。

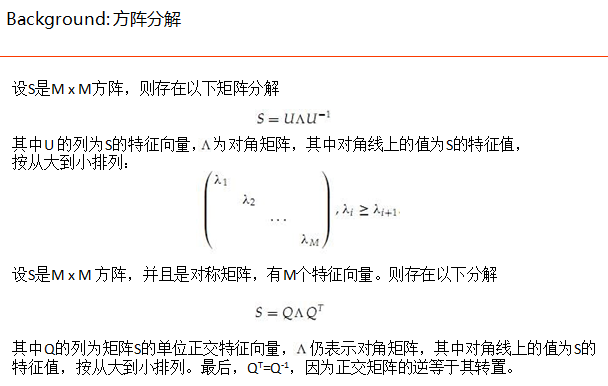
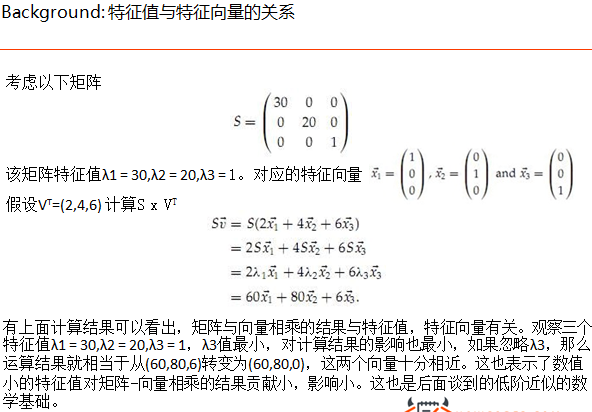
对角矩阵：对角矩阵是除对角线外所有元素都为零的方阵。

单位矩阵：如果对角矩阵中所有对角线上的元素都为零，该矩阵称为单位矩阵。

特征值：对一个M\*M矩阵C和向量X，如果存在λ，使得下式成立：

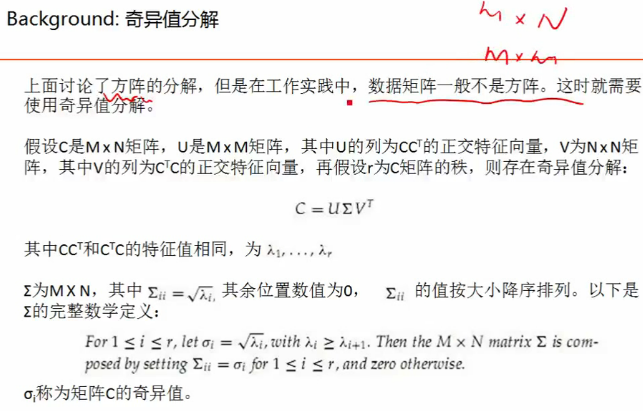
则称λ为矩阵C的特征值，X称为矩阵的特征向量。非零特征值的个数小于等于矩阵的秩。

一个矩阵乘以一个向量等于一个标量乘以一个向量，这个标量就是矩阵的特征值。

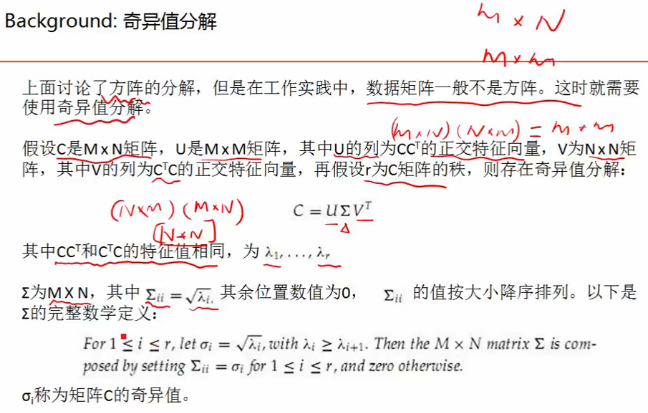


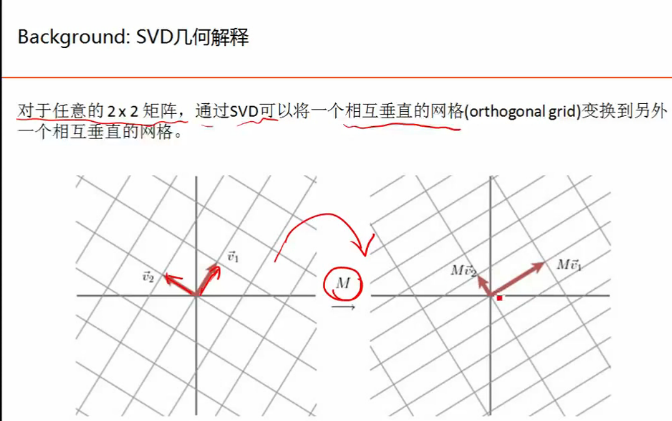
将一个方阵分解后就可以得到一个矩阵，一个对角阵，一个矩阵逆，三者相乘的结果。

当S是对称矩阵时，可发现u矩阵逆和u的转置是等价的。

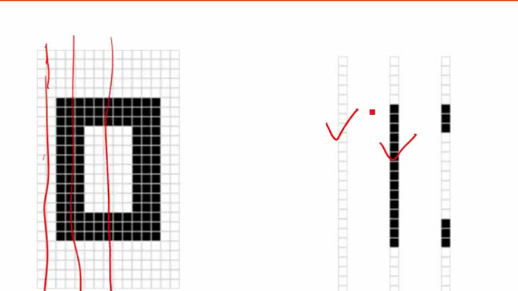


现实生活中的矩阵一般不是方阵(M\*M)，一般是矩阵(M\*N)的，不是方阵。

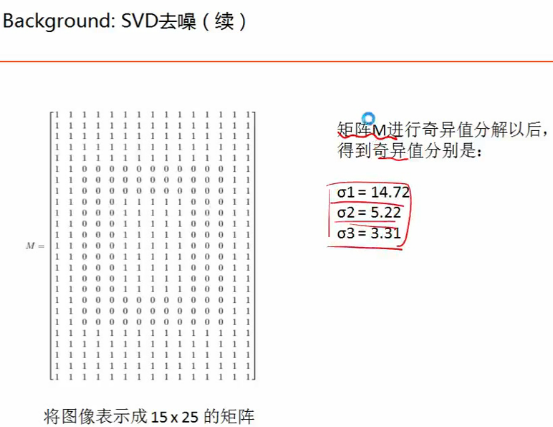


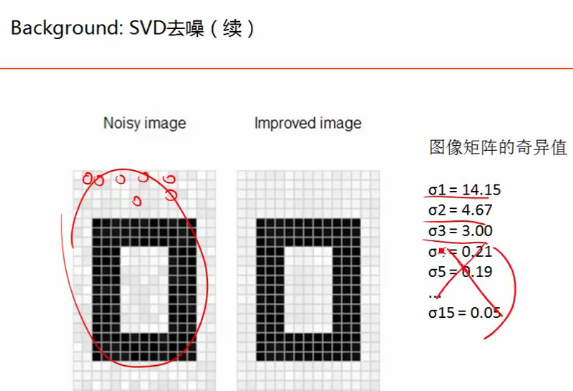


Svd得到M，然后将v1和v2分别乘以M得到右图当中的。当得到的其中一个向量M\*v1特别小，那么我们可以将其丢弃，使其变成一维空间。起到降维的作用。

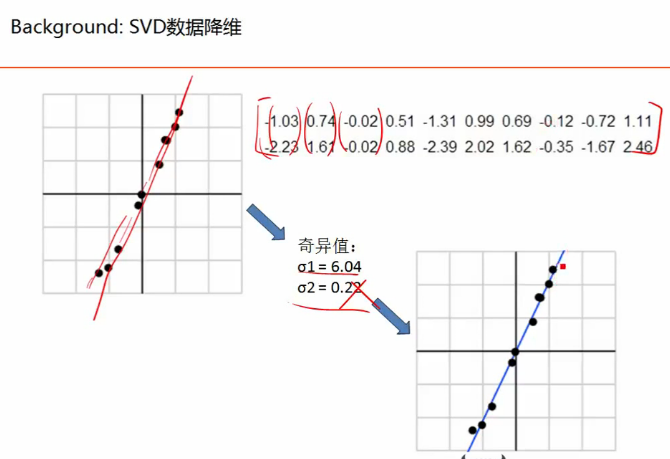
比如图像领域当中，用svd做特征表示，

左边的三条竖线可以由右边的三条竖线图像表示（有可能对应是横向的）。

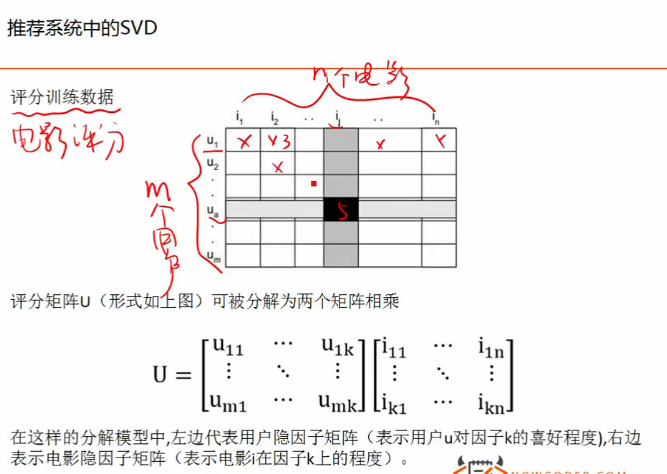




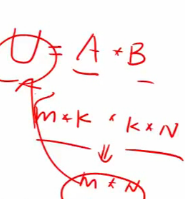
左边的图像加了一些噪声，通过svd呢，将其表示为多个子矩阵的，把子矩阵当中比较小的去掉，然后还原得到中间图形，这样就做到一个去噪的效果。

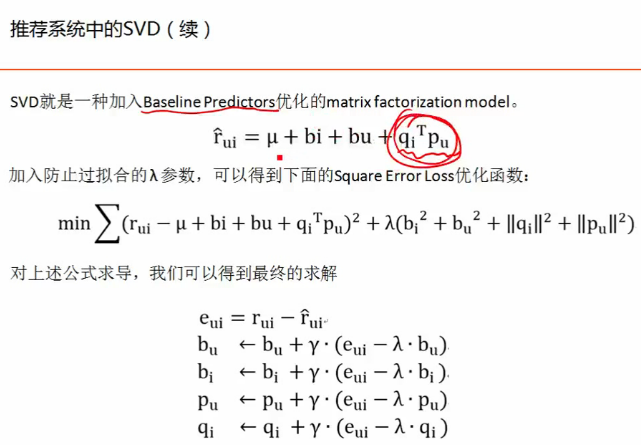


左图近似直线，右侧表示出每个点，然后svd得到奇异值，将奇异值小的丢弃，得到右图完全线性的图形。



推荐系统中，用户评价电影，m个用户，n个电影，中间是对电影的评分。

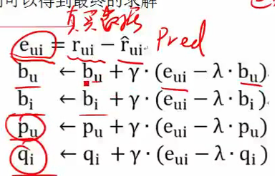
这个矩阵非常稀疏如果此人没有对某一个电影评分的话，那么我们可以根据矩阵求出来一个估值。

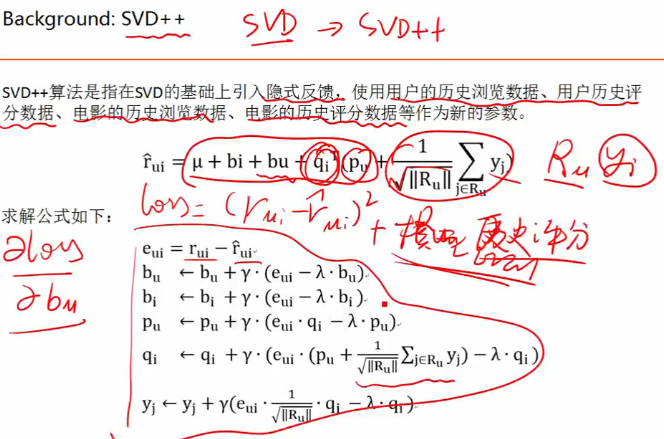


电影评分可以分解为两个矩阵的相乘，即。如果某个人对所有电影的评分都非常高，那么就根据Baseline Predictors去做调整，在公式中就是bu；比如有些电影，它天然的分就比较高，在公式中是bi；加上baseline之后，模型可以刻画的更细致一点。这就叫svd。

为了防止过拟合加入正则项，通过加L1或者L2范数，这里我们加入了L2范数(因为再svd当中好做求解)。

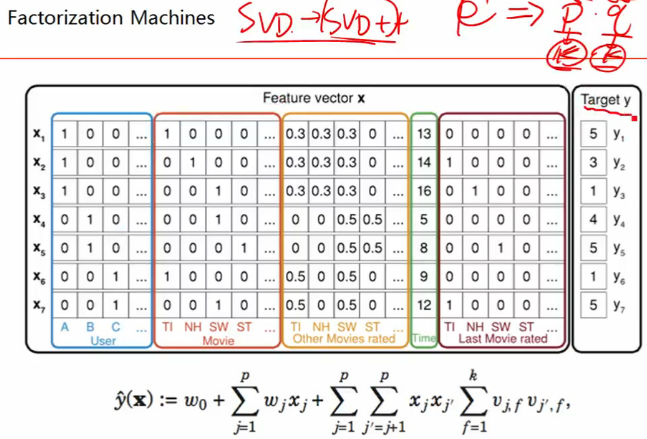
我们想最小化loss，那么就对这个loss对不同的参数求梯度，

分别对各项求导。e(ui)=真实-预测，求出e(ui)带入下边的式子中。

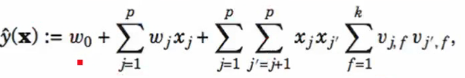


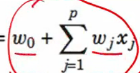
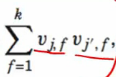
SVD经过变换得到svd++,即这是svd，然后再后边加一些变种的项，大集合R(u)历史评分，对该集合求平均(不是绝对的平均)，干扰模型的学习，让模型更好一点。

SVD++加了一些隐式的反馈进来，再netflix百万大奖中获奖挺高。

FM算法：用矩阵分解SVD、SVD++，把一个C矩阵最后变成一个P矩阵乘以一个Q矩阵，有一些隐层的k的表示，但是要加上一些隐式反馈的话，就变成了SVD++；但是要加其他的特征（比如要加实时特征）也不是很好加，于是就出来了FM，FM是把要加的信息全部表示成特征，

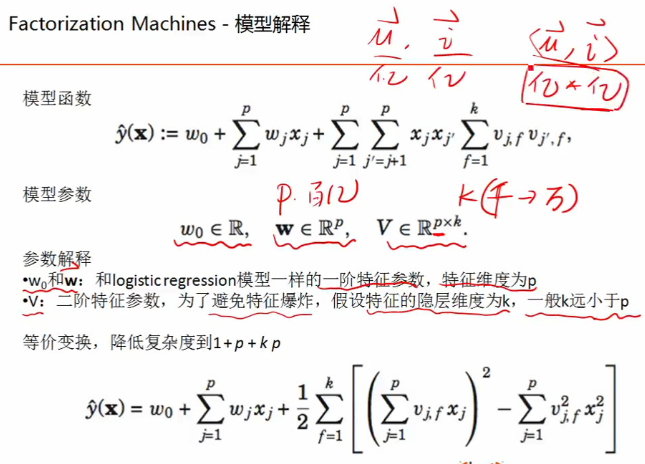
比如学习目标评分数据Target，FM相当于把SVD做了一个释放，可以加很多特征进来，不只是隐式反馈，可以把自己能够想到的所有特征全部加进来，只要我的程序能够处理，我可以想怎么加就怎么加。比如用户x7，User存放此用户年龄性别职业，浏览记录，使用设备等信息特征；Movie存放此用户的电影信息，电影的特征；评估过其他的特征也放进来等等总之所有能放进来的吧。

模型描述：

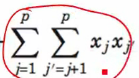
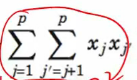
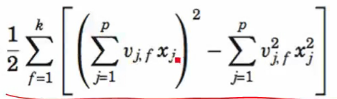
线性回归：。隐层项：vj是权重，wj是权重。

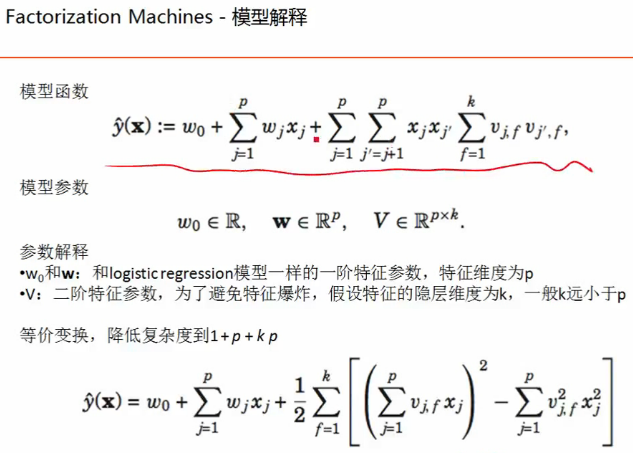
把x(j)隐层到一个k维空间，得到隐层向量然后再相乘。P是特征的维度，一般情况下特征的维度可能会很大，上百亿，上千亿。K可能比较小。

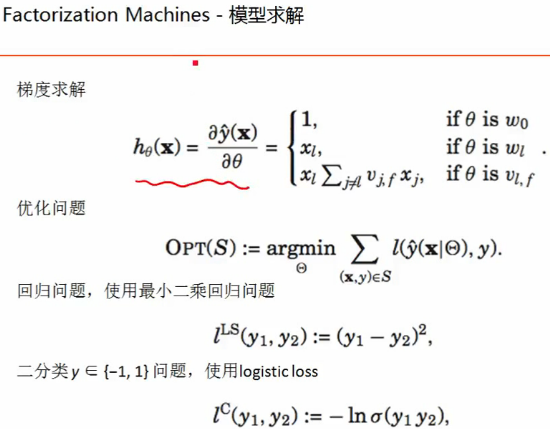
后面怎么优化，怎么求解呢？

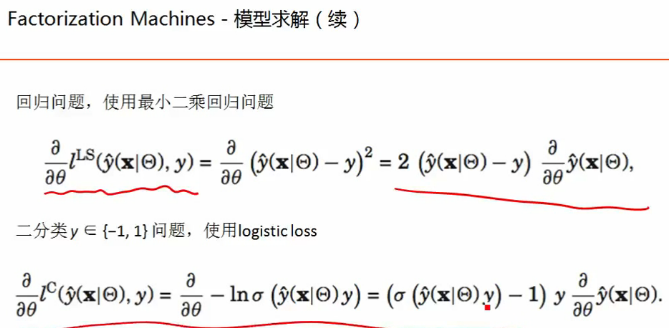


<u,i>亿级的特征向量\*亿级的特征向量，两者笛卡尔积，结果是亿亿维的特征向量，结果很难去算；FM算法的思想是先把亿级的特征向量都降到万的级别，然后再乘，这个模型就变得没有那么复杂。

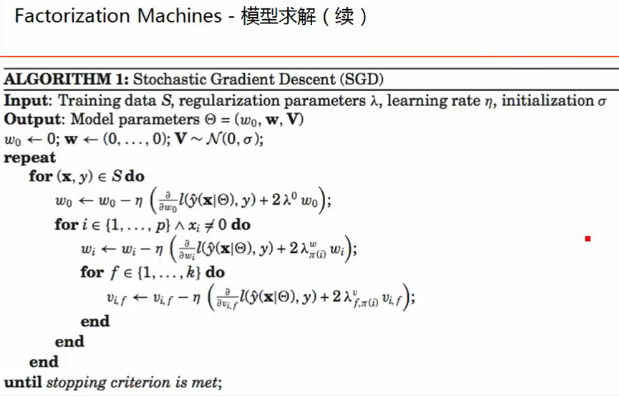
式子当中有，所以复杂度还是相当高，整个式子的复杂度就是1+P+(P的平方)，可以推导成，复杂度由p的平方降到了k\*p，





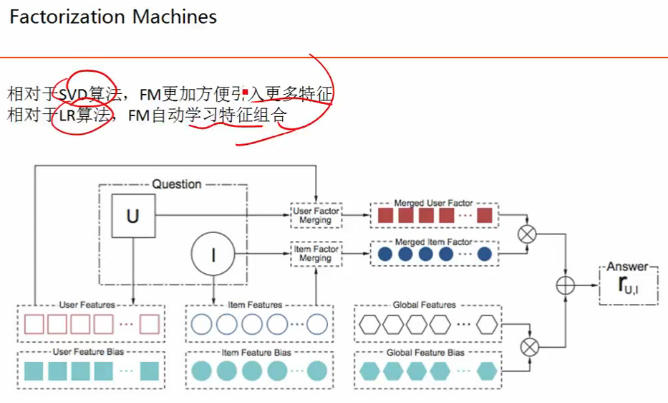


可以得到w和w(i)还有v相关的一个梯度公式，然后我们就可以使用SGD进行求解。

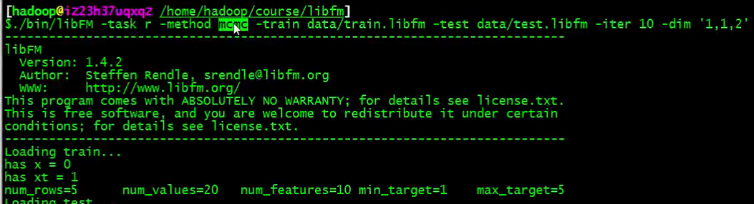


训练数据S，正则化系数lambda，学习率η，高斯方差σ（FM是一个非线性模型，所有的非线性模型都要有一个随机初始值，就是通过高斯分布来采出来的）

从训练集S当中拿到一条训练样本，根据前边的计算公式，可以得到w0的更新值，wi呢，x是向量，y是target是个标量，一般数据都是稀疏表达的，当x正好和特征空间当中的某一维命中了(w0,w1,w3,w5,y),这样w0可以根据公式计算出来，w1呢。最后的for循环是指隐层向量的维度，根据w1由v1\*k来计算，然后可以得到更新的公式，这样把数据不断遍历直到收敛。

K一般是经过人工设定的。

User、Iterm



-task包括：逻辑回归和回归。

-method方法：mcmc；一般提供SGD，ALS还有mcmc。

-train训练数据 -test测试数据

-iter迭代轮数是10轮。

-dim ‘1,1,2’ 是说1有没有w0，第二个数有没有w向量，第三个数有没有隐层向量维度，这里是2.

-out是最后test输出的结果

libFM

下载源码<http://www.libfm.org/>

训练集：我们有2个用户，3个商品的历史数据

测试集：同样的2个用户，但是有4个商品，其中3个训练集出现的商品，另外1个新的商品

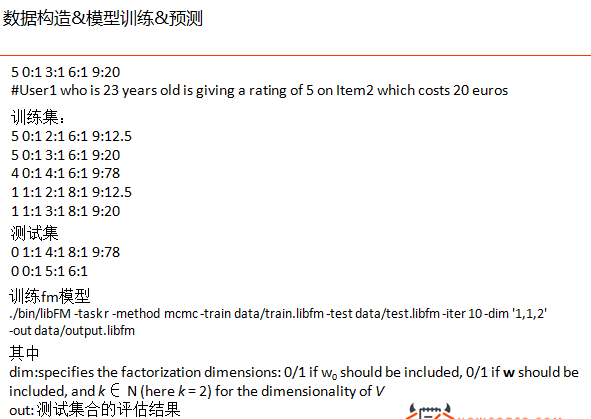
每个用户有一个年龄特征[“18-25″, “26-40″, “40-60″]

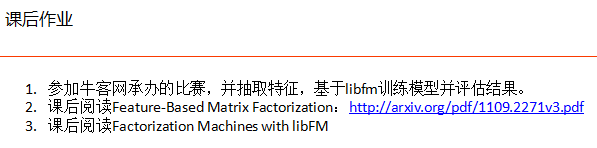
每个商品有一个价格特征

特征ID化：

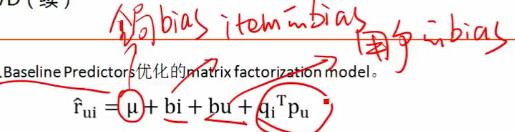
* 0 is User1
* 1 is User2
* 2 is Item1
* 3 is Item2
* 4 is Item3
* 5 is Item4
* 6 is the category “18-25″,
* 7 is the category “26-40″,
* 8 is the category “40-60″

9 will represent the price feature



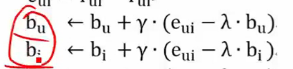


FM的复杂度比逻辑回归要复杂，比他会慢。



什么是笛卡尔组合特征？组合特征值大小是多少呢？

比如用逻辑回归时，用组合特征比较多，用户id，itermid，在做非线性刻画时就会将这两个id拼起来，形成一个新的id，这个就叫特征组合。

Bias是模型学习出来的。这两个都是bias的梯度推导公式。

课后题：通过大数据描绘用户画像，通过精准的推荐可以提高用户体验。我们提供S公司部分用户在APP内某月的历史卡片浏览记录，请预测未来三天用户在APP内的赞，踩等交互行为。

点击率是在线广告系统中衡量广告效率的最重要指标之一。我们提供S公司部分用户在APP内11月份的卡片浏览记录，请根据用户历史浏览交互行为预测11月28，29,30日三天的赞，踩交互行为。

## 数据获取

请点击下载本次比赛的所有数据，"高速下载"的密码为ste5。解压后会得到  
1. page\_view\_data.tar.gz  
该文件记录用户在11月份的浏览记录，解压后文件每行通过'\t'分隔，各列字段意义如下  
dev\_id —用户id  
stat\_date —日期yyyymmdd  
stat\_time —时间HHMMSS  
post\_id —卡片/帖子id  
type — 1是S公司卡片数据，2是L社区数据  
stat\_view — APP内浏览的次数  
stat\_click — 点击查看详情的次数  
每个用户对同一内容可能会有多次记录，表示用户的多次浏览/查看详情行为  
2.post\_data.txt  
该文件提供11月卡片/帖子具体的内容，文件每行通过'\t'分隔，各列字段意义如下  
id —帖子id  
title —帖子title，可能为空  
content --帖子内容  
3.train.txt  
该文件提供11月份前27天用户的交互记录，文件每行通过'\t'分隔，各列字段意义如下  
dev\_id —用户id  
post\_id —卡片/帖子id  
praise —1表示赞，2表示踩  
time —时间yyyymmddHHMMSS

[普通下载](http://ncbigdata.oss-cn-hangzhou.aliyuncs.com/data%2F%E6%97%B6%E9%97%B4%E9%BB%91%E5%AE%A2_%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98%E8%B5%9B%E6%95%B0%E6%8D%AE_contest_data.tar.gz)[高速下载](http://pan.baidu.com/s/1bobAO9L)data%2F%E6%97%B6%E9%97%B4%E9%BB%91%E5%AE%A2\_%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98%E8%B5%9B%E6%95%B0%E6%8D%AE\_contest\_data.tar.gz

## 结果文件上传

根据page\_view\_data.tar.gz的浏览记录，请预测用户28,29,30日的赞踩行为。提交文件每行应该包含dev\_id,post\_id,praise三个字段，各个字段通过'\t'分隔。具体请参考sample\_submission.txt  
注：提交文件最多包含20万行，最大10M  
记分规则：  
提交数据的记分通过F1-Measure来打分  
P 准确率  
R 召回率  
F1 = 2P\*R/(P+R)