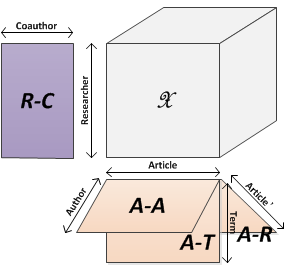
多矩阵协同分解

1. 原理



对所有的输入矩阵，程序会扫描一遍它们的列项目（column item）和行项目（row item），

根据项目的个数n，随即初始化n个矩阵，作为因子（factor）。

开始循环，

首先根据训练矩阵的分布模型，计算矩阵的损失（loss）

然后计算每一个因子的梯度（gradient）

最后一次更新所有的因子。

当达到制定循环次数之后退出循环。

1. 程序流程

训练：Train.m



输入：

1. 要训练的矩阵，是Matrix类的数组，每一个Matrix包括：

1、矩阵的数据

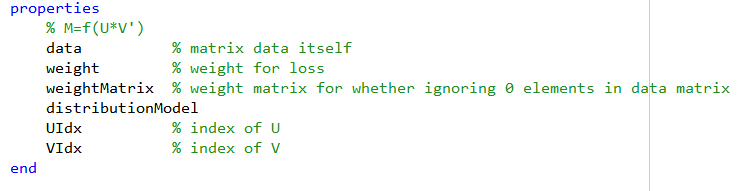
2、矩阵的loss的weight

3、矩阵内非missing elements的元素排布

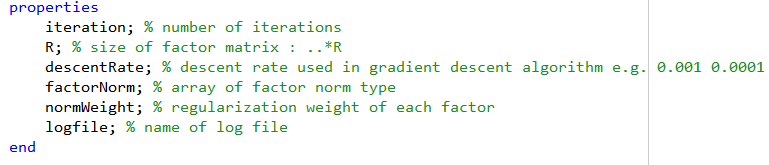
4、矩阵的分布模型

5、矩阵的行的index

6、矩阵的列的index



1. 参数，Param类型，包括，迭代次数，R的大小，梯度下降速度，因子的norm类型，因子的norm权重（用于惩罚函数），要写入的log文件文件名。



输出：

converge：是否收敛

factors：训练得到的factor数列（cell类型）

流程：



遍历train matrix，得到factor的个数



得到每一个factor的大小



随机初始化所有的factor matrix，调用函数



开始循环



设置梯度下降速率，如果loss增加就减小下降速率



计算误差，第一个for循环计算每一个矩阵的误差。第二个循环对每一个因子计算惩罚项



计算梯度，第一个for循环对每一个矩阵计算它对应的两个因子的梯度，第二个for循环对每一个因子根据惩罚项计算它的梯度。



更新所有的因子。

1. 预测精确度测试

目录下的movielen.dat文件是测试与训练数据。其中经过测试系统对bernoulli分布的test数据预测的RMSE（root mean square error）为0.01029。对poisson分布的test矩阵预测RMSE为0.926。