推荐系统学习-矩阵分解算法

基本思想

推荐系统里面需要用到MF算法，也就是把一个矩阵(user_num x item_num)分解为两个矩阵(user_num x trait_num)，(trait_num x item_num) 的乘积。这个过程很好理解，就是从评分矩阵中提取出来物品的特征，分解出来的两个矩阵分别表明用户对该特征的偏好程度和物品对该特征的包含程度。比如

$$U=\begin{bmatrix}1 & 2 & 3\\ 3 & 2 & 1\end{bmatrix}, V=\begin{bmatrix}1\\ 0\\ 1\end{bmatrix}$$

其中，三个特征分别表示喜剧、恐怖、剧情，所以 *U* 矩阵的第一行就表示用户1对喜剧爱好程度是1，恐怖爱好程度是2，剧情爱好程度是3，以此类推。而 *V* 的第一列就表示电影1包含了1的喜剧成分，0的恐怖成分，1的剧情成分。所以显然就有用户1对电影的评分是 $1\times 1+2\times 0+3\times 1=4$ 分。

这便是基本的矩阵分解（Matrix Factorization）方法。利用这个算法，我们只需要让最小化目标为

$$\min_{U,I}\sum_{i=0}^{n} R(i)-U(i)\cdot V(i)$$

R(i) 表示第 *i* 个评分，*U(i)* 表示第 *i* 个评分对应的用户。

进一步可以用矩阵表示，让R_{ij}表示第 *i* 个用户对 第 *j* 个商品的评分，让

$$E_{ij}=\begin{cases}0 & \text{无评分}\\ 1 & \text{有评分}\end{cases}$$

于是优化目标就可以改写为

$$\min_{U,I} E\cdot(R-UV)$$

但是这个算法还有诸多局限性，比如

- 每个用户的评分是不同的，有的用户心地善良，评分都是4,5分。有的用户眼光挑剔，评分大多是2,3分。
- 每个电影的评分也是不同的。有的电影由于一些原因，自带加分Buff，有的电影则因为某些原因自带减分Buff。
- 可能会出现过拟合现象。

为了解决以上问题，可以采用BiasedMF算法。这个算法里面加入了用户偏移和商品偏移。此时，一个商品的评分用以下公式表示：$Rp_{ij}=U(i)\cdot V(j)+bu(i)+bi(j)$

*Rp*表示预测的评分，*U V*定义同上，*bu*，*bi*分别是用户、商品的偏置系数。

同时，为了防止过拟合，对*U, V, bu, bi* 的二范数乘以一个重整化系数(通常称为regularization， reg)，加入损失函数中。

于是优化目标变为

$$\min_{U,I} E \cdot (R - (UV + bu \cdot [1 \quad 1 \dots \quad 1] + \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \cdot bi)) + \lambda(||U||^2 + ||V||^2 + ||bu||^2 + ||bi||^2)$$

但是此时可以明显看到，重整化系数让所有的矩阵都朝着绝对值变小的方向发展，也就是预测的评分会出现偏低。这样显然是不对的，所以需要对评分进行预处理，把所有评分都减去全部评分的均值。这样重整化系数就产生了避免评分离均值太远的趋势，较为合理。

具体算法

一种简单的算法时随机梯度下降。

即：对于每一个评分，进行误差计算然后求导，然后根据梯度下降的方法更新权重。

如：

$$Loss_{ij} = (r_{ij} - rp_{ij})^2 + 规范化项 = (r_{ij} - U_i \cdot V_j - bu_i - bi_j)^2 + 规范化项 = error^2 + 规范化项 \quad (1)$$

于是$\frac{\partial Loss}{\partial U_{ik}} = -2 \cdot (error \cdot V_{kj} - \lambda \cdot U_{ik})$

同理，容易看出：

$$\frac{\partial Loss}{\partial V_{kj}} = -2 \cdot (error \cdot U_{ik} - \lambda \cdot V_{kj})$$

$$\frac{\partial Loss}{\partial bu_i} = -2 \cdot (error - \lambda \cdot bu_i)$$

$$\frac{\partial Loss}{\partial bi_j} = -2 \cdot (error - \lambda \cdot bi_j)$$

而根据梯度下降的公式：

$\theta = \theta - learningRate \cdot \frac{\partial Loss}{\partial \theta}$

就可以在每一次迭代中更新参数，从而找到最优的U, V, bi, bu

同样的，也可以对整个矩阵进行梯度下降。这样子方便用Tensorflow训练。

$$Loss = \sum_{ij}(r_{ij} - rp_{ij}) + 规范化项 \quad (2)$$

规范化项的变化

(1)和(2)两个误差公式的区别，在于每次计算的评分的数目。在多个评分一起训练的时候，针对某一个参数，会有多个评分的误差对其进行修正，所以规范化项需要相应增加一定的倍数。

严格来说，（1）式最后的结果并不符合优化目标

$$\min_{U,I} E \cdot (R - (UV + bu \cdot [1 \quad 1 \dots \quad 1] + \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \cdot bi)) + \lambda(||U||^2 + ||V||^2 + ||bu||^2 + ||bi||^2)$$

因为对于每个参数，假设有 k 个评分关联了这个参数，那么实际上这个参数的规范化项也被带入计算了 k 次，所以最后的优化目标近似

$$\min_{U,I} E \cdot (R - (UV + bu \cdot [1 \quad 1 \dots \quad 1] + \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \cdot bi)) + k \cdot \lambda(||U||^2 + ||V||^2 + ||bu||^2 + ||bi||^2)$$

而整体训练，是完全符合原始的优化目标的。

数学意义

假设每个用户的特征是符合单位正态分布的，即：

$U_i \sim N(0, \sigma_u^2)$

同理，对于商品的特征，用户偏好，商品偏好我们也做如下假设

$V_i \sim N(0, \sigma_v^2) \quad bi \sim N(0, \sigma_{bi}^2) \quad bu \sim N(0, \sigma_{bu}^2)$

然后假设评分在预测值的基础上，加上了方差为 σ_r 的噪声。

注意符合以上条件时，很显然评分的均值就是0，所以首先需要把所有评分都减去平均值处理。

为了找到最好的参数组合，相当于是使得如下的概率取到最大值：

$$\max_{U,V,bi,bu} P(U, V, bi, bu | R, \sigma_r^2, \sigma_v^2, \sigma_{bi}^2, \sigma_{bu}^2)$$

根据贝叶斯公式，有

$$P(U, V, bi, bu | R, \sigma_r^2, \sigma_v^2, \sigma_{bi}^2, \sigma_{bu}^2) = \frac{P(R, U, V, bi, bu | \sigma_r^2, \sigma_v^2, \sigma_{bi}^2, \sigma_{bu}^2)}{P(R | \sigma_r^2, \sigma_v^2, \sigma_{bi}^2, \sigma_{bu}^2)}$$

很显然，我们只改变 U, V, bi, bu 四个参数，所以这个式子的分子相当于一个常数，可以不去考虑。

而分子又可以化为

$$P(R, U, V, bi, bu | \sigma_r^2, \sigma_v^2, \sigma_{bi}^2, \sigma_{bu}^2) = P(R | U, V, b_u, b_i) \cdot P(U | \sigma_u^2) P(V | \sigma_v^2) \cdot P(bi | sigma_{bi}^2) \cdot P(bu | \sigma_{bu}^2)$$

很容易就可以看出

$$P(R_{ij} | U, V, b_u, b_i) = \frac{1}{\sqrt{2\pi\sigma_r^2}} e^{-\frac{(R_{ij} - Pred_{ij})^2}{2\sigma_r^2}}$$

其中，$Pred_{ij} = U(i) \cdot V(j) + bu(i) + bi(j)$

因为每个评分的噪声是独立的，因此总的概率就是所有单个评分的概率连乘：

$$P(R|U,V,b_u,b_i)=\prod_{ij}[\frac{1}{\sqrt{2\pi\sigma_r^2}}e^{-\frac{(R_{ij}-Pred_{ij})^2}{2\sigma_r^2}}]^{I_{ij}}$$

其中，I_{ij} 为 1 表示有评分，为0则无评分（不贡献乘积）。

同理有：

$$P(U|\sigma_u^2)=\prod_{ij}\frac{1}{\sqrt{2\pi\sigma_u^2}}e^{-\frac{U_{ij}^2}{2\sigma_u^2}}$$

$$P(V|\sigma_v^2)=\prod_{ij}\frac{1}{\sqrt{2\pi\sigma_v^2}}e^{-\frac{V_{ij}^2}{2\sigma_v^2}}$$

$$P(bu|\sigma_{bu}^2)=\prod_{j}\frac{1}{\sqrt{2\pi\sigma_{bu}^2}}e^{-\frac{bu_j^2}{2\sigma_{bu}^2}}$$

$$P(bi|\sigma_{bi}^2)=\prod_{j}\frac{1}{\sqrt{2\pi\sigma_{bi}^2}}e^{-\frac{bi_j^2}{2\sigma_{bi}^2}}$$

这些概率相乘，然后取对数，就得到了熟悉的表达式：

$$-\frac{E\cdot(R-(UV+bu\cdot[1\quad 1\dots\quad 1]+\begin{bmatrix}1\\1\\\dots\\1\end{bmatrix}\cdot bi))}{\sigma_r^2}-(\frac{||U||^2}{\sigma_u^2}+\frac{||V||^2}{\sigma_v^2}+\frac{||bu||^2}{\sigma_{bu}^2}+\frac{||bi||^2}{\sigma_{bi}^2})$$

于是很容易得出，U, V, bu, bi 对应的规范化系数应该分别为：

$\sigma_u^2/\sigma_r^2,\sigma_v^2/\sigma_r^2,\sigma_{bu}^2/\sigma_r^2,\sigma_{bi}^2/\sigma_r^2$

在前面的推导中，我们全部用λ 代替了。因为实际上我们并不能猜测出U、V、bu、bi的方差，因此这个公式只是从理论上证明了规范化系数的合理性，并没有告诉我们如何确定规范化系数。

参考文献

[1] [Mnih and Salakhutdinov, 2007] Andriy Mnih and Ruslan Salakhutdinov. Probabilisticmatrixfactorization. InNIPS ’07, 2007.