

第 5 章



基于异构反馈的评分预测

在解决评分预测问题时,由于缺乏足够的等级评分数据(例如,1~5分的数值评分),协同过滤算法经常面对数据稀疏性的问题。为了缓解这一问题,Koren^[1]、Liu^[2]和Zhang^[3]等除了使用显式反馈等级评分数据之外,还使用了隐式反馈数据(例如,点击、购买、是否评分等对物品没有明确偏好的用户反馈和行为)来帮助提高评分预测的效果。但是,这些研究没有考虑利用其他类型的显式反馈数据来帮助缓解等级评分数据的稀疏性问题,例如,二值评分数据,即0或1的评分,其中,0表示用户对物品不喜欢(而不是指用户对物品的偏好信息缺失),1表示用户对物品喜欢。

本章将重点介绍 Pan 等^[4]和 Liang 等^[5]提出的在迁移学习框架下利用二值评分数据来缓解数据稀疏性问题的两个协同过滤算法。第一个是基于迁移学习的共同矩阵分解算法,又称为迁移共同分解(transfer by collective factorization, TCF)^[4],第二个是基于偏好迁移的协同过滤算法,又称为偏好感知迁移(preference-aware transfer, PAT)^[5]。

5.1 异构协同过滤(HCF)问题

Pan 等^[4]和 Liang 等^[5]研究的是面向用户异构显式反馈的评分预测问题,目的是在迁移学习框架中,利用二值评分数据来缓解等级评分数据的稀疏性问题,进而提高预测每个用户 u 对测试集中未评分物品的等级评分的效果。如图 5-1 所示,问题的输入是一组等级评分数据 $\mathcal{R} = \{(u, i, r_{ui}) \mid r_{ui} \in \mathbf{G}\}$,其中 $\mathbf{G} = \{1, 2, \dots, 5\}$,以及一组二值评分数据 $\tilde{\mathcal{R}} = \{(u, i, \tilde{r}_{ui}) \mid \tilde{r}_{ui} \in \mathbf{B}\}$,其中 $\mathbf{B} = \{1, 0\}$ 。需要说明的是,等级评分数据和二值评分数据是两种不同类型的显式反馈数据,即它们是异构的。本章从迁移学习的视角对用户的两种显式反馈数据进行建模,并把拥有等级评分数据的一方称为目标方,把拥有二值评分数据的一方称为辅助方。表 5-1 对本章使用的符号进行了说明。

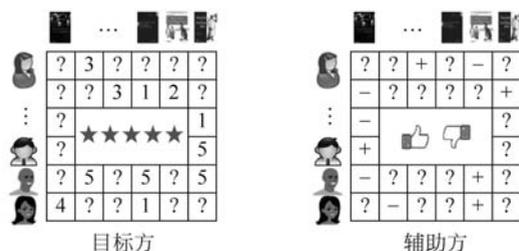


图 5-1 HCF 问题示意图

表 5-1 HCF 问题中的符号及其说明

符 号	说 明
n	用户数
m	物品数
$u \in \{1, 2, \dots, n\}$	用户 ID
$i, j \in \{1, 2, \dots, m\}$	物品 ID
$\mathbf{G} = \{1, 2, \dots, 5\}$	用户对物品的等级评分的范围
$\mathbf{B} = \{1, 0\}$	用户对物品的二值评分的范围, 1 表示喜欢, 0 表示不喜欢
$r_{ui} \in \mathbf{G}$	用户 u 对物品 i 的等级评分
$\tilde{r}_{ui} \in \mathbf{B}$	用户 u 对物品 i 的二值评分
$\mathcal{R} = \{(u, i, r_{ui})\}$	训练集中的等级评分记录集合
$\tilde{\mathcal{R}} = \{(u, i, \tilde{r}_{ui})\}$	训练集中的二值评分记录集合
\mathcal{I}	训练集中所有物品的集合
\mathcal{I}_u	训练集中用户 u 评过分的物品的集合
$\mathcal{I}_u^g, g \in \mathbf{G}$	训练集中用户 u 评分为 g 的物品的集合
$y_{ui} \in \{0, 1\}$	指示变量, 1 表示用户 u 对物品 i 评过分数, 0 表示没有评过分数
\mathcal{P}_u	训练集中用户 u 喜欢的物品的集合
\mathcal{N}_u	训练集中用户 u 不喜欢的物品的集合
$\mathcal{T}_E = \{(u, i, r_{ui})\}$	测试集中的等级评分记录集合
$\mu \in \mathbf{R}$	全局评分平均值
$b_u \in \mathbf{R}$	用户 u 的偏差(或偏置)
$b_i \in \mathbf{R}$	物品 i 的偏差(或偏置)
$d \in \mathbf{R}$	潜在特征向量的维度
$\mathbf{U}_u, \mathbf{W}_u \in \mathbf{R}^{1 \times d}$	用户 u 的潜在特征向量
$\mathbf{V}_i, \mathbf{C}_j^p, \mathbf{C}_j^n, \mathbf{C}_i^o, \mathbf{C}_i^g \in \mathbf{R}^{1 \times d}$	物品 i, j 和 i' 的潜在特征向量
$\mathbf{V}, \mathbf{C}^p, \mathbf{C}^n, \mathbf{C}^o, \mathbf{C}^g \in \mathbf{R}^{m \times d}$	物品的潜在特征矩阵
$\mathbf{B}, \tilde{\mathbf{B}} \in \mathbf{R}^{d \times d}$	目标方和辅助方的内部矩阵
\hat{r}_{ui}	用户 u 对物品 i 的预测评分
$\hat{\tilde{r}}_{ui}$	用户 u 对物品 i 的二值预测评分
γ	学习率
ω	等级评分和二值评分的交互权重参数
α, β	正则化项上的权衡参数

续表

符 号	说 明
λ	目标方和辅助方目标函数上的权衡参数
$\delta_g, \delta_o, \delta_p, \delta_n \in \{0, 1\}$	指示变量, 是否引入相应的特征向量
ω_p, ω_n	正反馈和负反馈的权重
T	算法的迭代次数

5.2 迁移共同分解

在基于迁移学习的协同过滤算法中, 一种典型的解决数据稀疏性问题的思路是将辅助方的二值评分数据(喜欢或不喜欢)中隐含的知识迁移到目标方, 以提高目标方的等级评分预测效果。Pan 等^[4]提出的 TCF 在迁移学习框架中联合建模了目标方的等级评分数据和辅助方的包含喜欢和不喜欢的二值评分数据, 有效地缓解了数据的稀疏性问题。本节将详细介绍 TCF 的技术细节。

5.2.1 技术细节

TCF^[4]的目标方和辅助方的预测公式分别如下所示:

$$\hat{r}_{ui} = \mathbf{U}_u \cdot \mathbf{B}\mathbf{V}_i^T \quad (5-1)$$

$$\hat{\tilde{r}}_{ui} = \mathbf{U}_u \cdot \tilde{\mathbf{B}}\mathbf{V}_i^T \quad (5-2)$$

其中, 目标方和辅助方共享用户的潜在特征向量为 \mathbf{U}_u , 和物品的潜在特征向量为 \mathbf{V}_i ; \mathbf{B} 和 $\tilde{\mathbf{B}}$ 分别用于捕捉目标方数据和辅助方数据中特有的信息(即目标方和辅助方的矩阵 \mathbf{U} 中的行和矩阵 \mathbf{V}^T 中的列之间的相关性), 因为目标方的等级评分数据和辅助方的二值评分数据的语义和分布等信息很可能是不一样的。

TCF^[4]的目标函数如下所示:

$$\begin{aligned} & \min_{\Theta} \sum_{u=1}^n \sum_{i=1}^m y_{ui} \left[\frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha}{2} (\|\mathbf{U}_u\|^2 + \|\mathbf{V}_i\|^2) \right] + \\ & \lambda \sum_{u=1}^n \sum_{i=1}^m \tilde{y}_{ui} \left[\frac{1}{2} (\tilde{r}_{ui} - \hat{\tilde{r}}_{ui})^2 + \frac{\alpha}{2} (\|\mathbf{U}_u\|^2 + \|\mathbf{V}_i\|^2) \right] + \frac{\beta}{2} \|\mathbf{B}\|_{\text{F}}^2 + \\ & \lambda \frac{\beta}{2} \|\tilde{\mathbf{B}}\|_{\text{F}}^2, \text{ s. t. } \mathbf{U}, \mathbf{V} \in D \end{aligned} \quad (5-3)$$

其中, $\Theta \in \{\mathbf{U}, \mathbf{V}, \mathbf{B}, \tilde{\mathbf{B}}\}$ 表示模型参数; y_{ui} 和 \tilde{y}_{ui} 分别是目标方和辅助方的指示变量, 它们为 1 时表示用户 u 对物品 i 有评分, 为 0 时表示用户 u 对物品 i 没有评分; D 表示潜在变量的值域。为了发现潜在的主题, 并降低噪声的影响, D 可以表示成 $D_{\text{R}} = \{\mathbf{U} \in \mathbb{R}^{n \times d}, \mathbf{V} \in \mathbb{R}^{m \times d}\}$ 或 $D_{\perp} = D_{\text{R}} \cap \{\mathbf{U}^T \mathbf{U} = \mathbf{I}, \mathbf{V}^T \mathbf{V} = \mathbf{I}\}$ 。因此, TCF 有两个变体: 当 D 表示成 D_{R} 时, TCF 为共同矩阵三分解(collective matrix tri-factorization, CMTF)^[4], 当 D 表示成 D_{\perp} 时, TCF 为共同奇异值分解(collective SVD, CSVD)^[4]。

令 $f_u = \sum_{i=1}^m y_{ui} \left[\frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha}{2} (\|U_{u\cdot}\|^2 + \|V_{i\cdot}\|^2) \right] + \frac{\beta}{2} \|B\|^2 + \lambda \sum_{i=1}^m \tilde{y}_{ui} \left[\frac{1}{2} (\tilde{r}_{ui} - \hat{\tilde{r}}_{ui})^2 + \frac{\alpha}{2} (\|U_{u\cdot}\|^2 + \|V_{i\cdot}\|^2) \right] + \lambda \frac{\beta}{2} \|\tilde{B}\|^2$, 给定 B 和 V , 可以得到 CMTF 中用户的潜在特征矩阵 U 的解析解, 推导过程如下所示:

$$\begin{aligned} \frac{\partial f_u}{\partial U_{u\cdot}} &= \sum_{i=1}^m y_{ui} [(-r_{ui} + \hat{r}_{ui}) V_{i\cdot} B^T + \alpha U_{u\cdot}] + \lambda \sum_{i=1}^m \tilde{y}_{ui} [(-\tilde{r}_{ui} + \hat{\tilde{r}}_{ui}) V_{i\cdot} \tilde{B}^T + \alpha U_{u\cdot}] \\ &= - \sum_{i=1}^m (y_{ui} r_{ui} V_{i\cdot} B^T + \lambda \tilde{y}_{ui} \tilde{r}_{ui} V_{i\cdot} \tilde{B}^T) + \alpha U_{u\cdot} \sum_{i=1}^m (y_{ui} + \lambda \tilde{y}_{ui}) + \\ &\quad U_{u\cdot} \sum_{i=1}^m (y_{ui} B V_{i\cdot}^T V_{i\cdot} B^T + \lambda \tilde{y}_{ui} \tilde{B} V_{i\cdot}^T V_{i\cdot} \tilde{B}^T) \end{aligned} \quad (5-4)$$

令 $\frac{\partial f_u}{\partial U_{u\cdot}} = \mathbf{0}$, 推导出用户 u 的潜在特征向量 $U_{u\cdot}$ 的计算公式如下所示:

$$U_{u\cdot} = \mathbf{b}_u C_u^{-1} \quad (5-5)$$

其中, $C_u = \sum_{i=1}^m (y_{ui} B V_{i\cdot}^T V_{i\cdot} B^T + \lambda \tilde{y}_{ui} \tilde{B} V_{i\cdot}^T V_{i\cdot} \tilde{B}^T) + \alpha \sum_{i=1}^m (y_{ui} + \lambda \tilde{y}_{ui}) I$; $\mathbf{b}_u = \sum_{i=1}^m (y_{ui} r_{ui} V_{i\cdot} B^T + \lambda \tilde{y}_{ui} \tilde{r}_{ui} V_{i\cdot} \tilde{B}^T)$ 。

类似地, 给定 B 和 U , 可以得到 CMTF 中物品的潜在特征矩阵 V 的解析解, 物品 i 的潜在特征向量 $V_{i\cdot}$ 的计算公式如下所示:

$$V_{i\cdot} = \mathbf{b}_i C_i^{-1} \quad (5-6)$$

其中, $C_i = \sum_{u=1}^n (y_{ui} B^T U_{u\cdot}^T U_{u\cdot} B + \lambda \tilde{y}_{ui} \tilde{B}^T U_{u\cdot}^T U_{u\cdot} \tilde{B}) + \alpha \sum_{u=1}^n (y_{ui} + \lambda \tilde{y}_{ui}) I$; $\mathbf{b}_i = \sum_{u=1}^n (y_{ui} r_{ui} U_{u\cdot} B + \lambda \tilde{y}_{ui} \tilde{r}_{ui} U_{u\cdot} \tilde{B})$ 。需要说明的是, 式(5-5)和式(5-6)中解析解的更新规则可以视为最小二乘(alternating least squares, ALS)方法^[6]的推广。

在 CSVD 中, 值域 D_{\perp} 的限制起到了与正则化项类似的效果, 将式(5-3)中的正则化项去掉, 可以得到新的目标函数:

$$\begin{aligned} \min_{U, V} & \frac{1}{2} \|Y \odot (R - UB V^T)\|_F^2 + \frac{\lambda}{2} \|\tilde{Y} \odot (\tilde{R} - U \tilde{B} V^T)\|_F^2, \\ \text{s. t. } & U^T U = I, \quad V^T V = I \end{aligned} \quad (5-7)$$

其中, \odot 表示矩阵中的元素逐项相乘。

令 $f = \frac{1}{2} \|Y \odot (R - UB V^T)\|_F^2 + \frac{\lambda}{2} \|\tilde{Y} \odot (\tilde{R} - U \tilde{B} V^T)\|_F^2$, U 的梯度计算公式如下所示:

$$\frac{\partial f}{\partial U} = (Y \odot (UB V^T - R)) V B^T + \lambda (\tilde{Y} \odot (U \tilde{B} V^T - \tilde{R})) \tilde{V} \tilde{B}^T \quad (5-8)$$

通过格拉斯曼流形(Grassmann manifold)梯度下降算法^[7]可以得到 U 的更新公式如下

所示:

$$\mathbf{U} \leftarrow \mathbf{U} - \gamma(\mathbf{I} - \mathbf{U}\mathbf{U}^T) \frac{\partial f}{\partial \mathbf{U}} = \mathbf{U} - \gamma \nabla \mathbf{U} \quad (5-9)$$

将式(5-9)代入目标函数(见式(5-7))中,可以得到如下公式:

$$\begin{aligned} g(\gamma) &= \frac{1}{2} \|\mathbf{Y} \odot [\mathbf{R} - (\mathbf{U} - \gamma \nabla \mathbf{U}) \mathbf{B} \mathbf{V}^T]\|_{\text{F}}^2 + \frac{\lambda}{2} \|\tilde{\mathbf{Y}} \odot [\tilde{\mathbf{R}} - (\mathbf{U} - \gamma \nabla \mathbf{U}) \tilde{\mathbf{B}} \mathbf{V}^T]\|_{\text{F}}^2 \\ &= \frac{1}{2} \|\mathbf{Y} \odot (\mathbf{R} - \mathbf{U} \mathbf{B} \mathbf{V}^T) + \gamma \mathbf{Y} \odot (\nabla \mathbf{U} \mathbf{B} \mathbf{V}^T)\|_{\text{F}}^2 + \frac{\lambda}{2} \|\tilde{\mathbf{Y}} \odot (\tilde{\mathbf{R}} - \mathbf{U} \tilde{\mathbf{B}} \mathbf{V}^T) + \gamma \tilde{\mathbf{Y}} \odot (\nabla \mathbf{U} \tilde{\mathbf{B}} \mathbf{V}^T)\|_{\text{F}}^2 \end{aligned} \quad (5-10)$$

令 $\mathbf{t}_1 = \mathbf{Y} \odot (\mathbf{R} - \mathbf{U} \mathbf{B} \mathbf{V}^T)$, $\tilde{\mathbf{t}}_1 = \tilde{\mathbf{Y}} \odot (\tilde{\mathbf{R}} - \mathbf{U} \tilde{\mathbf{B}} \mathbf{V}^T)$, $\mathbf{t}_2 = \mathbf{Y} \odot (\nabla \mathbf{U} \mathbf{B} \mathbf{V}^T)$, $\tilde{\mathbf{t}}_2 = \tilde{\mathbf{Y}} \odot (\nabla \mathbf{U} \tilde{\mathbf{B}} \mathbf{V}^T)$, 有 $g(\gamma) = \frac{1}{2} \|\mathbf{t}_1 + \gamma \mathbf{t}_2\|_{\text{F}}^2 + \frac{\lambda}{2} \|\tilde{\mathbf{t}}_1 + \gamma \tilde{\mathbf{t}}_2\|_{\text{F}}^2$, 以及梯度如下:

$$\frac{\partial g(\gamma)}{\partial \gamma} = \text{tr}(\mathbf{t}_1^T \mathbf{t}_2) + \gamma \text{tr}(\mathbf{t}_2^T \mathbf{t}_2) + \lambda [\text{tr}(\tilde{\mathbf{t}}_1^T \tilde{\mathbf{t}}_2) + \gamma \text{tr}(\tilde{\mathbf{t}}_2^T \tilde{\mathbf{t}}_2)] \quad (5-11)$$

其中, $\text{tr}(\cdot)$ 是矩阵的迹(trace)。令 $\frac{\partial g(\gamma)}{\partial \gamma} = 0$, 可以得到 $\gamma = \frac{-\text{tr}(\mathbf{t}_1^T \mathbf{t}_2) - \lambda \text{tr}(\tilde{\mathbf{t}}_1^T \tilde{\mathbf{t}}_2)}{\text{tr}(\mathbf{t}_2^T \mathbf{t}_2) + \lambda \text{tr}(\tilde{\mathbf{t}}_2^T \tilde{\mathbf{t}}_2)}$ 。

类似地, 可以得到物品的潜在特征矩阵 \mathbf{V} 的更新规则如下所示:

$$\mathbf{V} \leftarrow \mathbf{V} - \gamma \nabla \mathbf{V} \quad (5-12)$$

其中, $\nabla \mathbf{V} = (\mathbf{I} - \mathbf{V} \mathbf{V}^T) \frac{\partial f}{\partial \mathbf{V}}$, $\frac{\partial f}{\partial \mathbf{V}} = (\mathbf{Y} \odot (\mathbf{U} \mathbf{B} \mathbf{V}^T - \mathbf{R})) \mathbf{U} \mathbf{B} + \lambda (\tilde{\mathbf{Y}} \odot (\mathbf{U} \tilde{\mathbf{B}} \mathbf{V}^T - \tilde{\mathbf{R}})) \mathbf{U} \tilde{\mathbf{B}}$ 。

给定 \mathbf{U} 和 \mathbf{V} , 目标方和辅助方可以分别计算 \mathbf{B} 和 $\tilde{\mathbf{B}}$ 。令 $\mathcal{F}(\mathbf{R} \sim \mathbf{U} \mathbf{B} \mathbf{V}^T) = \sum_{u=1}^n \sum_{i=1}^m y_{ui} \left[\frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha}{2} (\|\mathbf{U}_u\|^2 + \|\mathbf{V}_i\|^2) \right] + \frac{\beta}{2} \|\mathbf{B}\|_{\text{F}}^2$, 有:

$$\begin{aligned} \mathcal{F}(\mathbf{R} \sim \mathbf{U} \mathbf{B} \mathbf{V}^T) &\propto \sum_{u=1}^n \sum_{i=1}^m y_{ui} \left[\frac{1}{2} (r_{ui} - \hat{r}_{ui})^2 \right] + \frac{\beta}{2} \|\mathbf{B}\|_{\text{F}}^2 \\ &= \frac{1}{2} \|\mathbf{Y} \odot (\mathbf{R} - \mathbf{U} \mathbf{B} \mathbf{V}^T)\|_{\text{F}}^2 + \frac{\beta}{2} \|\mathbf{B}\|_{\text{F}}^2 \end{aligned} \quad (5-13)$$

因此, 可以得到如下等价的目标函数:

$$\min_{\mathbf{B}} \frac{1}{2} \|\mathbf{Y} \odot (\mathbf{R} - \mathbf{U} \mathbf{B} \mathbf{V}^T)\|_{\text{F}}^2 + \frac{\beta}{2} \|\mathbf{B}\|_{\text{F}}^2 \quad (5-14)$$

其中, \mathbf{B} 的计算可以与最小二乘 SVM 问题^[8]中的 \mathbf{w} 的计算完全相同; $\mathbf{w} = \text{vec}(\mathbf{B}) = [\mathbf{B}^T_{\cdot 1} \cdots \mathbf{B}^T_{\cdot d}]^T \in \mathbb{R}^{d^2 \times 1}$ 是由矩阵 \mathbf{B} 的列拼接而成的向量。当 $y_{ui} = 1$ 时, 可以构造样本 $\{(\mathbf{x}_{ui}, r_{ui})\}$, 其中 $\mathbf{x}_{ui} = \text{vec}(\mathbf{U}_u^T \cdot \mathbf{V}_i \cdot \cdot) \in \mathbb{R}^{d^2 \times 1}$ 。因此, 我们可以得到与最小二乘 SVM 问题等价的目标函数, 如下所示:

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{r} - \mathbf{X} \mathbf{w}\|_{\text{F}}^2 + \frac{\beta}{2} \|\mathbf{w}\|_{\text{F}}^2 \quad (5-15)$$

其中, 数据矩阵 $\mathbf{X} = [\cdots \mathbf{x}_{ui} \cdots]^T \in \mathbb{R}^{\rho \times d^2}$ (当 $y_{ui} = 1$ 时), 评分 $\mathbf{r} \in \{1, 2, 3, 4, 5\}^{\rho \times 1}$, $\rho =$

$\sum_{u=1}^n \sum_{i=1}^m y_{ui}$ 。令 $\nabla \mathbf{w} = -\mathbf{X}^T (\mathbf{r} - \mathbf{X} \mathbf{w}) + \beta \mathbf{w} = \mathbf{0}$, 我们可以得到:

$$w = (\mathbf{X}^T \mathbf{X} + \beta \mathbf{I})^{-1} \mathbf{X}^T \mathbf{r} \quad (5-16)$$

需要说明的是, \mathbf{B} 和 w 可以被视为线性紧算子(linear compact operator), 它们能够使用各种现有的离线工具来有效求解。

5.2.2 算法流程

构建 TCF^[4] 模型的流程如下: ①对目标方的等级评分数据的范围进行处理, 使得它与辅助方的二值评分数据的范围一致; ②在算法的第一次迭代中, 对 \mathbf{U} 和 \mathbf{V} 进行随机初始化(对于 CMTF)或使用辅助方的二值评分数据的训练集 $\tilde{\mathcal{R}}$ 的 SVD 结果^[9] (对于 CSVD), 然后通过式(5-16)计算 \mathbf{B} 和 $\tilde{\mathbf{B}}$; ③在算法的第 t ($t > 1$) 次迭代中, 更新 $\mathbf{U}, \mathbf{V}, \mathbf{B}, \tilde{\mathbf{B}}$ 。重复步骤②和步骤③, 直到收敛。在算法 5-1 中, 我们可以看到 TCF 算法流程的伪代码。

算法 5-1 TCF 算法

输入: 目标方的等级评分数据的训练集 \mathcal{R} , 辅助方的二值评分数据的训练集 $\tilde{\mathcal{R}}$

输出: 每个用户 u 对测试集 \mathcal{I}_E 中的物品 i 的预测评分

1. 通过 $r_{ui} = \frac{r_{ui} - 1}{4}$, $r_{ui} \in \mathcal{R}$ 对目标方的等级评分进行处理
2. **if** $t = 1$ **then**
3. 对于 CMTF, 随机初始化 \mathbf{U} 和 \mathbf{V} ; 对于 CSVD, 使用 $\tilde{\mathcal{R}}$ 的 SVD 结果
4. 通过式(5-16)计算 \mathbf{B} 和 $\tilde{\mathbf{B}}$
5. **end if**
6. **for** $t = 2, 3, \dots, T$ **do**
7. 对于 CMTF, $T_1 = 1$; 对于 CSVD, $T_1 = 20$
8. **for** $t_1 = 1, 2, \dots, |T_1|$ **do**
9. 固定 \mathbf{V} 和 \mathbf{B} , 对于 CMTF, 通过式(5-5)更新 \mathbf{U} ; 对于 CSVD, 通过式(5-9)更新 \mathbf{U}
10. 固定 \mathbf{U} 和 \mathbf{B} , 对于 CMTF, 通过式(5-6)更新 \mathbf{V} ; 对于 CSVD, 通过式(5-12)更新 \mathbf{V}
11. if 模型已收敛 **then**
12. 跳出当前循环
13. **end if**
14. **end for**
15. 固定 \mathbf{U} 和 \mathbf{V} , 通过式(5-16)更新 \mathbf{B} 和 $\tilde{\mathbf{B}}$
16. **if** 模型已收敛 **then**
17. 跳出当前循环
18. **end if**
19. **end for**

5.2.3 代码实现

TCF 代码的运行环境为 MATLAB, 算法 5-1 的第 7~15 行对应的核心代码如下:

```

//CMTF
//固定  $U$  和  $B$ , 通过式(5-6)更新  $V$ 
U2 = [U * B; U * B_aux];
V = MF_U(vec_V(:, [2 1 3]), tradeoff_alpha_V, V, U2, tradeoff_lambda, num_user);

//固定  $V$  和  $B$ , 通过式(5-5)更新  $U$ 
V2 = [V * B'; V * B_aux'];
U = MF_U(vec_U, tradeoff_alpha_U, U, V2, tradeoff_lambda, num_item);

//固定  $U$  和  $V$ , 通过式(5-16)更新  $B$  和  $\tilde{B}$ 
B = EstimateB(U, V, train_vec, tradeoff_beta);
B_aux = EstimateB(U, V, aux_vec, tradeoff_beta_aux);

//CSVD
for iterUV = 1:20

    //固定  $V$  和  $B$ , 通过式(5-9)更新  $U$ 
    GradU = CalGradU(U, V, B, B_aux, train_vec, aux_vec, tradeoff_lambda);
    gamma = StepSize_gamma( U, GradU, V, B, B_aux, train_vec, aux_vec, tradeoff_lambda );
    U = U - gamma * GradU;

    //固定  $U$  和  $B$ , 通过式(5-12)更新  $V$ 
    GradV = CalGradU(V, U, B', B_aux', train_vec(:, [2 1 3]), aux_vec(:, [2 1 3]), tradeoff_lambda);
    gamma = StepSize_gamma( V, GradV, U, B', B_aux', train_vec(:, [2 1 3]), aux_vec(:, [2 1 3]), tradeoff_lambda );
    V = V - gamma * GradV;
end

//固定  $U$  和  $V$ , 通过式(5-16)更新  $B$  和  $\tilde{B}$ 
B = EstimateB(U, V, train_vec, tradeoff_beta);
B_aux = EstimateB(U, V, aux_vec, tradeoff_beta_aux);

```

5.2.4 实验设置

对于 CMTF 和 CSVD, Pan 等^[4] 设置模型的潜在特征向量的维度 $d=10$, 算法的总迭代次数 $T=100$, 辅助方的二值评分数据的权重 λ 从 $\{0.01, 0.1, 1\}$ 中进行选择, 对所有的数据集都有 $\lambda=0.1$ 。对于 CMTF, 正则化项上的权衡参数 $\beta=1$, 正则化项上的权衡参数 α 从 $\{0.01, 0.1, 1\}$ 中进行选择, 对所有的数据集都有 $\alpha=0.1$ 。

5.2.5 讨论

Pan 等^[4] 针对协同过滤中的数据稀疏性问题, 提出了一个基于迁移学习的共同矩阵分解(TCF)算法。TCF 在目标方和辅助方的用户和物品都对齐的情况下, 将辅助方的二值评分数据(喜欢或不喜欢)中的知识迁移到目标方, 以提高目标方的等级评分预测效果。

在 MoviePilot^[10] 和 Netflix 这两个数据集上的实验结果表明,在不同的数据稀疏程度下, CMTF 和 CSVD 的效果都优于基准方法。

未来还可以在其他的一些问题设置中进一步拓展 TCF 框架,例如,当目标方和辅助方的用户和物品只有部分对齐时、当需要考虑目标方和辅助方之间的用户隐私和数据安全时^[11]等。

5.3 偏好感知迁移

迁移混合分解模型(transfer by mixed factorization, TMF)^[12] 利用辅助方的二值评分数据中用户的隐式偏好上下文来帮助提高目标方的评分预测效果,而 SVD++^[13] 和 MF-MPC^[14] 利用了目标方的等级评分数据中用户的隐式偏好上下文来更好地学习用户的个性化偏好。为了结合两者的优点,Liang 等^[5] 基于迁移学习框架提出共享目标方和辅助方的评分信息,并以互补的方式将等级评分、二值评分和目标方与辅助方的隐式偏好上下文这三者所包含的信息融合在一起,进而设计了一个偏好感知迁移学习(PAT)算法,以提高用户对物品的评分预测效果。

5.3.1 技术细节

共同矩阵分解(collective matrix factorization, CMF)算法^[15] 利用目标方和辅助方的两种不同类型的显式反馈数据来联合构建推荐模型。在模型的构建过程中,目标方和辅助方共享物品的潜在特征向量 \mathbf{V}_i 。目标方和辅助方的评分预测公式分别如下所示:

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{U}_u \cdot \mathbf{V}_i^T \quad (5-17)$$

$$\hat{r}_{ui} = \mathbf{W}_u \cdot \mathbf{V}_i^T \quad (5-18)$$

我们可以发现,CMF 在以等级评分预测为目标来构建模型时,并没有考虑隐式偏好上下文的信息。

根据 SVD++^[13]、MF-MPC^[14] 和 TMF^[12] 的建模方式,Liang 等^[5] 得到如下偏好上下文:

$$\bar{\mathbf{C}}_u^o = \delta_o \frac{1}{\sqrt{|\mathcal{J}_u \setminus \{i\}|}} \sum_{i' \in \mathcal{J}_u \setminus \{i\}} \mathbf{C}_{i'}^o \quad (5-19)$$

$$\bar{\mathbf{C}}_u^g = \delta_g \sum_{g \in \mathbf{G}} \frac{1}{\sqrt{|\mathcal{J}_u^g \setminus \{i\}|}} \sum_{i' \in \mathcal{J}_u^g \setminus \{i\}} \mathbf{C}_{i'}^g \quad (5-20)$$

$$\bar{\mathbf{C}}_u^p = \delta_p \omega_p \frac{1}{\sqrt{|\mathcal{P}_u|}} \sum_{j \in \mathcal{P}_u} \mathbf{C}_j^p \quad (5-21)$$

$$\bar{\mathbf{C}}_u^n = \delta_n \omega_n \frac{1}{\sqrt{|\mathcal{N}_u|}} \sum_{j \in \mathcal{N}_u} \mathbf{C}_j^n \quad (5-22)$$

其中, $\bar{\mathbf{C}}_u^o$ 表示单类偏好上下文(one-class preference context, OPC),其认为用户 u 对物品 i 的预测评分除了与物品 i 有关以外,还与用户 u 评过分的其他物品 $i' \in \mathcal{J}_u \setminus \{i\}$ 有关; $\bar{\mathbf{C}}_u^g$ 表示多类偏好上下文(multiclass preference context, MPC),其在单类偏好上下文的

基础上将用户 u 对除物品 i 以外的其他物品 $i' \in \mathcal{J}_u \setminus \{i\}$ 根据评分等级 $g \in \mathbb{G}$ 进行了分类； \bar{C}_u^p 表示正反馈偏好上下文，其认为对物品集合具有相似喜好的用户在对其他物品的喜好上也是相似的； \bar{C}_u^n 表示负反馈偏好上下文，其认为对物品集合具有相似不喜好的用户在对其他物品的不喜好上也是相似的； $\delta_o, \delta_g, \delta_p, \delta_n \in \{0, 1\}$ 是指示变量，为 1 时表示引入对应的偏好上下文，为 0 时表示不引入对应的偏好上下文； w_p 和 w_n 分别表示正反馈和负反馈上的权重。

Liang 等^[5] 将这四类偏好上下文融入迁移学习框架中，分别得到以下目标方和辅助方的预测公式：

$$\hat{r}_{ui} = \mu + b_u + b_i + (\mathbf{U}_u + \bar{C}_u^o + \bar{C}_u^g + \bar{C}_u^p + \bar{C}_u^n) \mathbf{V}_i^T \quad (5-23)$$

$$\hat{\tilde{r}}_{ui} = \mathbf{W}_u \cdot \mathbf{V}_i^T \quad (5-24)$$

其中， \bar{C}_u^o 是 SVD++^[13] 中的偏好上下文，它会使得具有相似单类偏好上下文的用户的潜在特征向量接近。需要说明的是，Liang 等将偏好上下文融入目标方的预测公式中而不是辅助方的预测公式中，这是因为 Liang 等的研究目标是等级评分预测，而不是二值评分预测。在图 5-2 中，可以看到目标方和辅助方的整体预测规则，图中的绿色虚线表示引入目标方和辅助方之间的交互。需要说明的是，图 5-2 中的 C_j^g 表示 MF-MPC^[14] 中的多类偏好上下文，如果该项不考虑评分的等级，就会退化为 SVD++^[13] 中的单类偏好上下文 C_j^o ，对应地，PAT 也退化为 PAT-OPC^[5]。

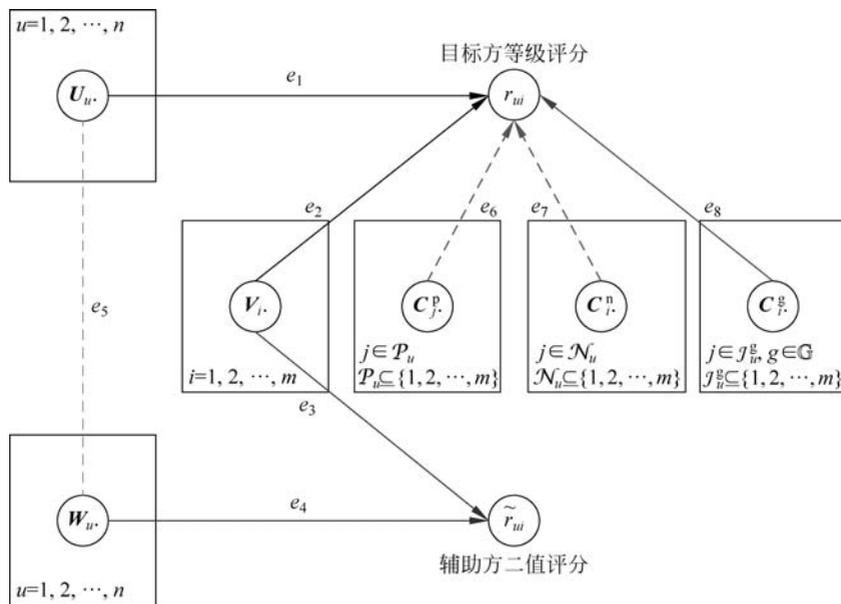


图 5-2 PAT 算法的模型结构示意图(见彩插)

PAT^[5] 的目标函数与 TMF^[12] 的类似，如下所示：

$$\min_{\theta} \sum_{u=1}^n \sum_{i=1}^m y_{ui} f_{ui} + \lambda \sum_{u=1}^n \sum_{i=1}^m \tilde{y}_{ui} \tilde{f}_{ui} \quad (5-25)$$

其中, y_{ui} 和 \tilde{y}_{ui} 分别是目标方和辅助方的指示变量, 为 1 时表示用户 u 对物品 i 有评分, 为 0 时表示用户 u 对物品 i 没有评分, $f_{ui} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha}{2}(\|U_{u\cdot}\|^2 + \|V_{i\cdot}\|^2 + b_u^2 + b_i^2 + \delta_p \sum_{j \in \mathcal{P}_u} \|C_j^p\|^2 + \delta_n \sum_{j \in \mathcal{N}_u} \|C_j^n\|^2 + \delta_o \sum_{i' \in \mathcal{J}_u \setminus \{i\}} \|C_{i'}^o\|^2 + \delta_g \sum_{g \in \mathbf{G}} \sum_{i' \in \mathcal{J}_u^g \setminus \{i\}} \|C_{i'}^g\|^2)$ 是目标方定义在用户 u 和物品 i 上的目标函数, $\tilde{f}_{ui} = \frac{1}{2}(\tilde{r}_{ui} - \hat{\tilde{r}}_{ui})^2 + \frac{\alpha}{2}(\|W_{u\cdot}\|^2 + \|V_{i\cdot}\|^2)$ 是辅助方定义在用户 u 和物品 i 上的目标函数, $\Theta = \{\mu, b_u, b_i, U_{u\cdot}, V_{i\cdot}, C_j^p, C_j^n, C_{i'}^o, C_{i'}^g, | u=1, 2, \dots, n; i, i', j=1, 2, \dots, m\}$ 表示模型的参数。

关于 f_{ui} , 模型参数的梯度如下所示:

$$\nabla \mu = \frac{\partial f_{ui}}{\partial \mu} = -e_{ui} \quad (5-26)$$

$$\nabla b_u = \frac{\partial f_{ui}}{\partial b_u} = -e_{ui} + ab_u \quad (5-27)$$

$$\nabla b_i = \frac{\partial f_{ui}}{\partial b_i} = -e_{ui} + ab_i \quad (5-28)$$

$$\nabla U_{u\cdot} = \frac{\partial f_{ui}}{\partial U_{u\cdot}} = -e_{ui} V_{i\cdot} + \alpha U_{u\cdot} \quad (5-29)$$

$$\nabla V_{i\cdot} = \frac{\partial f_{ui}}{\partial V_{i\cdot}} = -e_{ui}(\omega U_{u\cdot} + (1-\omega)W_{u\cdot} + \bar{C}_u^p + \bar{C}_u^n + \bar{C}_u^o + \bar{C}_u^g) + \alpha V_{i\cdot} \quad (5-30)$$

$$\nabla C_{i'}^o = \frac{\partial f_{ui}}{\partial C_{i'}^o} = \delta_o(-e_{ui} \frac{1}{\sqrt{|\mathcal{J}_u \setminus \{i\}|}} V_{i\cdot} + \alpha C_{i'}^o), \quad i' \in \mathcal{J}_u \setminus \{i\} \quad (5-31)$$

$$\nabla C_{i'}^g = \frac{\partial f_{ui}}{\partial C_{i'}^g} = \delta_g(-e_{ui} \frac{1}{\sqrt{|\mathcal{J}_u^g \setminus \{i\}|}} V_{i\cdot} + \alpha C_{i'}^g), \quad i' \in \mathcal{J}_u^g \setminus \{i\}, g \in \mathbf{G} \quad (5-32)$$

其中, $e_{ui} = r_{ui} - \hat{r}_{ui}$ 是目标方中真实评分与预测评分之间的差, $\omega U_{u\cdot} + (1-\omega)W_{u\cdot}$ 用于引入目标方和辅助方的用户潜在特征向量之间的交互^[16]。

关于 \tilde{f}_{ui} , 模型参数的梯度如下所示:

$$\nabla C_j^p = \frac{\partial \tilde{f}_{ui}}{\partial C_j^p} = \delta_p(-e_{ui} \omega_p \frac{1}{\sqrt{|\mathcal{P}_u|}} V_{i\cdot} + \alpha C_j^p), \quad j \in \mathcal{P}_u \quad (5-33)$$

$$\nabla C_j^n = \frac{\partial \tilde{f}_{ui}}{\partial C_j^n} = \delta_n(-e_{ui} \omega_n \frac{1}{\sqrt{|\mathcal{N}_u|}} V_{i\cdot} + \alpha C_j^n), \quad j \in \mathcal{N}_u \quad (5-34)$$

$$\nabla W_{u\cdot} = \frac{\partial \tilde{f}_{ui}}{\partial W_{u\cdot}} = \lambda(-\tilde{e}_{ui} V_{i\cdot} + \alpha W_{u\cdot}) \quad (5-35)$$

$$\nabla V_{i\cdot} = \frac{\partial \tilde{f}_{ui}}{\partial V_{i\cdot}} = \lambda(-\tilde{e}_{ui}(\omega W_{u\cdot} + (1-\omega)U_{u\cdot}) + \alpha V_{i\cdot}) \quad (5-36)$$

其中, $\tilde{e}_{ui} = \tilde{r}_{ui} - \hat{\tilde{r}}_{ui}$ 是辅助方中真实评分与预测评分之间的差。

目标方和辅助方的模型参数的更新规则如下所示：

$$\theta = \theta - \gamma \nabla \theta \quad (5-37)$$

其中, γ 为学习率, $\gamma > 0$ 。

5.3.2 算法流程

构建 PAT^[5] 模型的流程如下：①先从目标方数据和辅助方数据的并集 $\mathcal{R} \cup \tilde{\mathcal{R}}$ 中随机挑选一条评分记录 $(u, i, r_{ui}, \tilde{r}_{ui})$, 如果评分记录来源于目标方, 那么该评分记录用于计算目标方中的模型参数 $(\mu, b_u, b_i, \mathbf{U}_u, \mathbf{V}_i, \mathbf{C}_j^p, \mathbf{C}_j^n, \mathbf{C}_i^o, \text{和 } \mathbf{C}_i^s)$ 的梯度, 如果评分记录来源于辅助方, 那么该评分记录用于计算辅助方中的模型参数 $(\mathbf{V}_i, \text{和 } \mathbf{W}_u)$ 的梯度；②利用步骤①计算得到的模型参数的梯度更新对应的模型参数；③更新学习率。重复上述三个步骤, 直到收敛。在算法 5-2 中, 我们可以看到 PAT 算法流程的伪代码。

算法 5-2 PAT 算法

输入：目标方的等级评分数据训练集 \mathcal{R} , 辅助方的二值评分数据训练集 $\tilde{\mathcal{R}}$

输出：每个用户 u 对测试集 T_E 中物品 i 的预测评分

1. **for** $t=1, 2, \dots, T$ **do**
2. **for** $t_1=1, 2, \dots, |\mathcal{R} \cup \tilde{\mathcal{R}}|$ **do**
3. 从目标方和辅助方的数据中随机挑选一条评分记录
4. 根据评分记录的来源方, 计算对应方中的模型参数梯度
5. 利用模型参数梯度通过式(5-37)更新模型参数
6. **end for**
7. 通过 $\gamma = \gamma \times 0.9$ 更新学习率
8. **end for**

5.3.3 代码实现

PAT 代码的运行环境为 JDK 1.8 和 MyEclipse 2020, 算法 5-2 的第 3~5 行对应的核心代码如下：

```
//从目标方和辅助方的数据并集中随机挑选一条评分记录
int rand_case = (int) Math.floor( Math.random() * Data.num_train );
int userID = Data.indexUserTrain[rand_case];
int itemID = Data.indexItemTrain[rand_case];
float rating = Data.ratingTrain[rand_case];

//如果随机挑选的评分记录属于辅助方, 那么利用该评分计算辅助方的用户和物品梯度( $\nabla \mathbf{W}_u$  和  $\nabla \mathbf{V}_i$ ), 并更新对应的用户和物品的潜在特征向量( $\mathbf{W}_u$  和  $\mathbf{V}_i$ )
if( rand_case > Data.num_train_target && Data.lambda > 0 )
{
```

```

float auxiliary_pred = 0;
float auxiliary_err = 0;
for (int f = 0; f < Data.d; f++)
{
    //计算辅助方的预测评分  $\hat{r}_{ui}$ 
    auxiliary_pred += Data.W[userID][f] * Data.V[itemID][f];
}

for (int f = 0; f < Data.d; f++)
{
    //计算用户梯度  $\nabla \mathbf{w}_u$ .
    float grad_W_f = Data.lambda * ( - auxiliary_err * Data.V[itemID][f] +
Data.alpha_w * Data.W[userID][f] );
    //计算物品梯度  $\nabla \mathbf{v}_i$ .
    float grad_V_f = Data.lambda * ( - auxiliary_err * ( Data.rho * Data.W
[userID][f] + (1 - Data.rho) * Data.U[userID][f] ) + Data.alpha_v * Data.V[itemID][f] );
    //更新用户和物品的潜在特征向量 ( $\mathbf{w}_u$  和  $\mathbf{v}_i$ .)
    Data.W[userID][f] = Data.W[userID][f] - Data.gamma * grad_W_f;
    Data.V[itemID][f] = Data.V[itemID][f] - Data.gamma * grad_V_f;
}
}

//如果随机挑选的评分记录属于目标方,那么利用该评分计算目标方的用户和物品梯度 ( $\nabla \mathbf{u}_u$  和
// $\nabla \mathbf{v}_i$ ),并更新对应的用户和物品的潜在特征向量 ( $\mathbf{u}_u$  和  $\mathbf{v}_i$ .)
if( rand_case <= Data.num_train_target )
{
    float [] tilde_Uu_g = new float[Data.d];
    float [] tilde_Uu = new float[Data.d];

    if(Data.delta_g!= 0) //如果引入了多类偏好上下文  $\mathbf{c}_i^g$ .
    {
        for(int g = 1; g <= Data.num_rating_types; g++)
        {
            if( Data.user_graded_rating_number[userID][g]>0 )
            {
                HashSet < Integer > itemSet = Data.Train_ExplicitFeedbacksGraded.get
(userID).get(g);
                //计算等级评分个数并开根号
                float explicit_feedback_num_u_sqrt = 0;
                if(itemSet.contains(itemID) )
                {
                    if( itemSet.size()>1 )
                    {
                        explicit_feedback_num_u_sqrt
= (float) Math.sqrt ( Data.user_graded_rating_number
[userID][g] - 1 );

```



```

    }
}
//计算正则化项
for (int f = 0; f < Data.d; f++)
{
    Pu[f] = Pu[f] / Data.UserPositiveNumAuxiliary[userID];
}
}

// "不喜欢"偏好上下文  $C_j^o$  的计算与"喜欢"偏好上下文  $C_j^p$  的计算类似,省略

float pred = 0;
float err = 0;
for (int f = 0; f < Data.d; f++)
{
    //计算目标方的预测评分  $\hat{r}_{ui}$ 
    pred += ( Data.U[userID][f] + Data.delta_p * Data.weight_p * Pu[f] +
Data.delta_n * Data.weight_n * Nu[f] + Data.delta_g * tilde_Uu[f] + Data.delta_o * Ou
[f]) * Data.V[itemID][f];
}
    pred += Data.g_avg + Data.biasU[userID] + Data.biasV[itemID];
    err = rating - pred; //计算真实评分和预测评分的误差

//更新  $\mu$ 
Data.g_avg = Data.g_avg - Data.gamma * ( -err );

//更新  $b_u$  和  $b_i$ 
Data.biasU[userID] = Data.biasU[userID] - Data.gamma * ( -err + Data.beta_u
* Data.biasU[userID] );
Data.biasV[itemID] = Data.biasV[itemID] - Data.gamma * ( -err + Data.beta_v
* Data.biasV[itemID] );

//更新用户和物品的潜在特征向量  $U_u$  和  $V_i$ 
float [] V_before_update = new float[Data.d];
for(int f = 0; f < Data.d; f++)
{
    V_before_update[f] = Data.V[itemID][f];
    float grad_U_f = -err * Data.V[itemID][f] + Data.alpha_u * Data.U
[userID][f];
    float grad_V_f = -err * ( ( Data.rho * Data.U[userID][f] + (1 - Data.rho) *
Data.W[userID][f] ) + Data.delta_p * Data.weight_p * Pu[f] + Data.delta_n * Data.
weight_n * Nu[f] + Data.delta_g * tilde_Uu[f] + Data.delta_o * Ou[f] ) + Data.alpha_v
* Data.V[itemID][f];
    Data.U[userID][f] = Data.U[userID][f] - Data.gamma * grad_U_f;
    Data.V[itemID][f] = Data.V[itemID][f] - Data.gamma * grad_V_f;
}
}

//更新"喜欢"偏好对应的模型参数  $C_j^p$ 

```

```

if( Data.delta_p!= 0 && Data.UserPositiveNumAuxiliary[userID]> 0 )
{
    HashSet< Integer> itemSet = Data.AuxiliaryDataPositive.get(userID);
    for(int auxiliary_itemID : itemSet)
    {
        for (int f = 0; f < Data.d; f++)
        {
            Data.P[auxiliary_itemID][f] = Data.P[auxiliary_itemID][f] - Data.
gamma * Data.delta_p * ( - err * Data.weight_p * V_before_update[f] / Data.
UserPositiveNumAuxiliary[userID] + Data.alpha_p * Data.P[auxiliary_itemID][f] );
        }
    }
}
//更新"不喜欢"偏好对应的模型参数  $c_j^n$ . 与更新"喜欢"偏好对应的模型参数  $c_j^p$ . 类似,
//省略

//更新单类偏好上下文对应的模型参数  $c_i^o$ . 与更新多类偏好上下文对应的模型参数  $c_i^g$ .
//类似, 省略

//更新多类偏好上下文对应的模型参数  $c_i^g$ .
if(Data.delta_g!= 0)
{
    for(int g = 1; g <= Data.num_rating_types; g++)
    {
        if( Data.user_graded_rating_number[userID][g]> 0 )
        {
            HashSet< Integer> itemSet = Data.Train_ExplicitFeedbacksGraded.get
(userID).get(g);

            float explicit_feedback_num_u_sqrt = 0;
            if(itemSet.contains(itemID) )
            {
                if( itemSet.size(>1) )
                {
                    explicit_feedback_num_u_sqrt = (float) Math.sqrt( Data.
user_graded_rating_number[userID][g] - 1 );
                }
            }
            else
            {
                explicit_feedback_num_u_sqrt = (float) Math.sqrt( Data.user_
graded_rating_number[userID][g] );
            }

            if(explicit_feedback_num_u_sqrt > 0)
            {

```


factorization, iTCF)^[16]和 TMF^[12]可以看作 PAT 的特例,这些模型与 PAT 的关系(以及在图 5-2 中对应的边)如表 5-2 所示。

表 5-2 CMF、iTCF 和 TMF 与 PAT 的关系

方 法	边
CMF	$\{e_1, e_2, e_3, e_4\}$
iTCF	$\{e_1, e_2, e_3, e_4, e_5\}$
TMF	$\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$
PAT	$\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$

从表 5-2 可以看出,CMF 的目标方和辅助方共享了物品的潜在特征向量,iTCF 在 CMF 的基础上引入了目标方和辅助方的用户潜在特征向量之间的交互,TMF 在 iTCF 的基础上引入了辅助方中用户对物品的喜欢或不喜欢的偏好,而 PAT 在 TMF 的基础上又进一步引入了目标方数据中的偏好上下文。它们的模型效果随着模型组件数量的增多而变好,一方面,这表明未来可以通过增加模型组件的数量来进一步提高模型的效果,另一方面,这也表明了 PAT 是一个较为灵活和通用的框架。

5.4 本章小结

本章针对基于异构反馈的评分预测问题,对 TCF^[4]和 PAT^[5]进行了较为详细的介绍,其中,TCF 较早在迁移学习框架中联合建模了目标方的等级评分数据和辅助方的包含喜欢和不喜欢的二值评分数据,进而缓解了协同过滤中的数据稀疏性问题,而 PAT 可以视为 TCF 的扩展。与 TCF 相比,PAT 引入了更多的模型组件,具有更好的推荐效果。另外,还讨论了 PAT 的一些特例(CMF^[15]、iTCF^[16]和 TMF^[12]),感兴趣的读者可以阅读原论文进一步了解。

需要说明的是,HCF 问题可以看作一个特殊的跨域推荐^[17-18]问题,其中的目标方和辅助方分别对应目标领域和辅助领域,因此从跨域推荐的角度对 HCF 问题进行扩展和求解也是一个值得关注的点。

随着人们对隐私保护的重视以及相关隐私保护法律法规的颁布,面向 HCF 问题的推荐算法中存在的用户隐私泄露问题也越来越值得关注,例如,目标方和辅助方的用户潜在特征向量之间的交互过程会导致两方的用户偏好被泄露。关于个体用户的隐私保护和组织间的数据安全,我们将会在第 9 章和第 10 章进行介绍。

5.5 参考文献

- [1] KOREN Y. Factor in the neighbors: Scalable and accurate collaborative filtering [J]. ACM Transactions on Knowledge Discovery from Data, 2010, 4(1): 1: 1-1; 24.
- [2] LIU N, XIANG E W, ZHAO M, et al. Unifying explicit and implicit feedback for collaborative filtering [C]//HUANG J, KOUDAS N, JONES G J F, et al. Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10). New York:

- ACM,2010: 1445-1448.
- [3] ZHANG Y, NIE J. Probabilistic latent relational model for integrating heterogeneous information for recommendation (Technical report: UCSC-SOE-10-22) [R]. Santa Cruz: School of Engineering of UCSC, 2010.
 - [4] PAN W, YANG Q. Transfer learning in heterogeneous collaborative filtering domains [J]. *Artificial Intelligence*, 2013, 197: 39-55.
 - [5] LIANG F, DAI W, HUANG Y, et al. PAT: Preference-aware transfer learning for recommendation with heterogeneous feedback [C]//Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN'20). Piscataway: IEEE, 2020: 1-7.
 - [6] BELL R M, KOREN Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights [C]//Proceedings of the 7th IEEE International Conference on Data Mining (ICDM'07). Piscataway: IEEE, 2007: 43-52.
 - [7] EDELMAN A, ARIAS T A, SMITH S T. The geometry of algorithms with orthogonality constraints [J]. *SIAM Journal on Matrix Analysis and Applications*, 1998, 20(2): 303-353.
 - [8] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers [J]. *Neural Processing Letters*, 1999, 9(3): 293-300.
 - [9] CREMONESI P, KOREN Y, TURRIN R. Performance of recommender algorithms on top-N recommendation tasks [C]//AMATRIAIN X, TORRENS M, RESNICK P, et al. Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10). New York: ACM, 2010: 39-46.
 - [10] SAID A, BERKOVSKY S, LUCA E. Putting things in context: Challenge on context-aware movie recommendation [C]//Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa'10). ACM: New York, 2010: 2-6.
 - [11] YANG Q, LIU Y, CHEN T, et al. Federated machine learning: Concept and applications [J]. *ACM Transactions on Intelligent Systems and Technology*, 2019, 10(2): 12: 1-12: 19.
 - [12] PAN W, XIA S, LIU Z, et al. Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks [J]. *Information Sciences*, 2016, 332: 84-93.
 - [13] KOREN Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model [C]//LI Y, LIU B, SARAWAGI S. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08). New York: ACM, 2008: 426-434.
 - [14] PAN W, MING Z. Collaborative recommendation with multiclass preference context [J]. *IEEE Intelligent Systems*, 2017, 32(2): 45-51.
 - [15] SINGH A P, GORDON G J. Relational learning via collective matrix factorization [C]//LI Y, LIU B, SARAWAGI S. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08). New York: ACM, 2008: 650-658.
 - [16] PAN W, MING Z. Interaction-rich transfer learning for collaborative filtering with heterogeneous user feedback [J]. *IEEE Intelligent Systems*, 2014, 29(6): 48-54.
 - [17] 陈雷慧, 匡俊, 陈辉, 等. 跨领域推荐技术综述 [J]. *华东师范大学学报(自然科学版)*, 2017, (5): 101-116.
 - [18] 刘真, 田靖玉, 苑宝鑫, 等. 基于知识聚合和迁移的跨领域推荐算法 [J]. *电子学报*, 2020, 48(10): 1928-1932.

5.6 习题

1. 本章研究的问题(即基于异构反馈的评分预测问题)与第2章中研究的问题(即基

于显式反馈的评分预测问题)有哪些联系和区别?

2. 请说一说,在一个真实的应用中,包含的异构反馈的形式通常有哪些?如果要模拟生成这样的数据,应该怎么做?

3. 请简要阐述迁移学习用于目标数据和辅助数据联合建模的基本思路。

4. TCF 的两个变体(CMTF 和 CSVD)的主要区别是什么?

5. SVD++和 MF-MPC 的主要区别是什么?

6. 能否在 PAT 的基础上加入更多的组件?

7. TCF 和 PAT 是如何用辅助方的二值评分信息来提高目标方的等级评分预测效果的?

8. TCF 和 PAT 的主要区别是什么?

9. 请想一想,当有超过两种类型的显式反馈时,如何扩展 TCF 和 PAT?

10. 请说一说,在本章研究的问题中,什么是目标方,什么是辅助方?

11. 在 RecSys、SIGIR 等知名国际会议的论文集中,查阅最新的基于迁移学习的推荐算法(近 3 年),并从迁移的知识和迁移的方式两方面对相关算法进行总结。