

摘要

数据稀疏是推荐系统中最具有挑战性的问题之一。一个有希望解决该问题的方法是跨领域推荐，通过从多种领域拓展反馈或评分来提高协同推荐的表现。在本文中，我们提出一种嵌入和映射（Embedding and Mapping）的跨领域推荐框架，称为EMCDR。EMCDR框架在两个方面不同于已知的跨领域推荐模型。首先，一个多层感知机被用于捕获跨领域的非线性映射函数，它为在每个领域学习领域性的实体特征提供了很大灵活性。其次，只有拥有充足数据的实体被用于学习映射函数，保证了其面对因数据稀疏而引起的噪音的鲁棒性。在两个跨领域推荐情景中的扩展实验证明了EMCDR算法超过了当前最高水平的跨领域推荐方法。

2 EMCDR框架

在这一部分，我们定义了跨领域推荐问题并且呈现了EMCDR框架。

假设我们有两个拥有相同用户与（或）物品的领域。在一个领域出现的用户/物品意味着在另一个领域也将会出现。在这个意义上来说，这两个领域也拥有相同的用户和物品。不失一般性的，我们可以指定一个领域为源领域（source domain），一个领域为目标领域（target domain）。让 $\mu = \{\mu_1, \mu_2, \dots\}$ 和 $V = \{v_1, v_2, \dots\}$ 分别表示用户集和物品集， R^s 和 R^t 分别表示源领域和目标领域的评分矩阵，其中 R_{ij}^s 是源领域中用户 u_i 给物品 v_j 的评分， R_{ij}^t 是在目标领域中对应的评分。

给定两个被观测到的部分矩阵， R^s 和 R^t ，以及跨领域的用户集和物品集， μ 和 v ，我们的目标是通过跨领域的扩充，给只有少量信息的目标领域的用户和物品进行推荐。为了实现这个目的，我们提出一个框架EMCDR。该框架包含三个主要步骤，即潜变量建模，隐空间映射以及跨领域推荐。

在第一步中，我们的目标是利用潜变量建模来找到用户/物品在隐空间中的表达。潜变量模型假定有一个元素与每个用户和每个物品相关。并且可观察到的用户-物品评分矩阵是用户和物品的潜变量之间交互的结果。在隐空间的用户和物品因子可以被表示为特征向量。我们把从源领域学习到的潜变量 R^s 表示为 $\{U^s, V^s\}$ ，目标领域为 $\{U^t, V^t\}$ 。在第二步中，我们通过映射函数来为用户/物品之间的跨领域关系进行建模。我们认为在源领域和目标领域的用户/物品特征向量之间有一种潜在的映射关系，因此可以用一个映射函数来表达这种关系。通常的，给定两个领域中用户 u_i 的特征向量， U_i^s 和 U_i^t ，我们采用映射函数 $f(\cdot; \theta)$ 来表示这种跨领域的关系 $U_i^t = f(U_i^s; \theta)$ ，其中 θ 是映射函数的参数。最后，我们为目标领域新用户/物品进行推荐。利用在源领域的潜变量和映射函数，我们可以获得其在目标领域的潜变量。EMCDR框架表示如下。

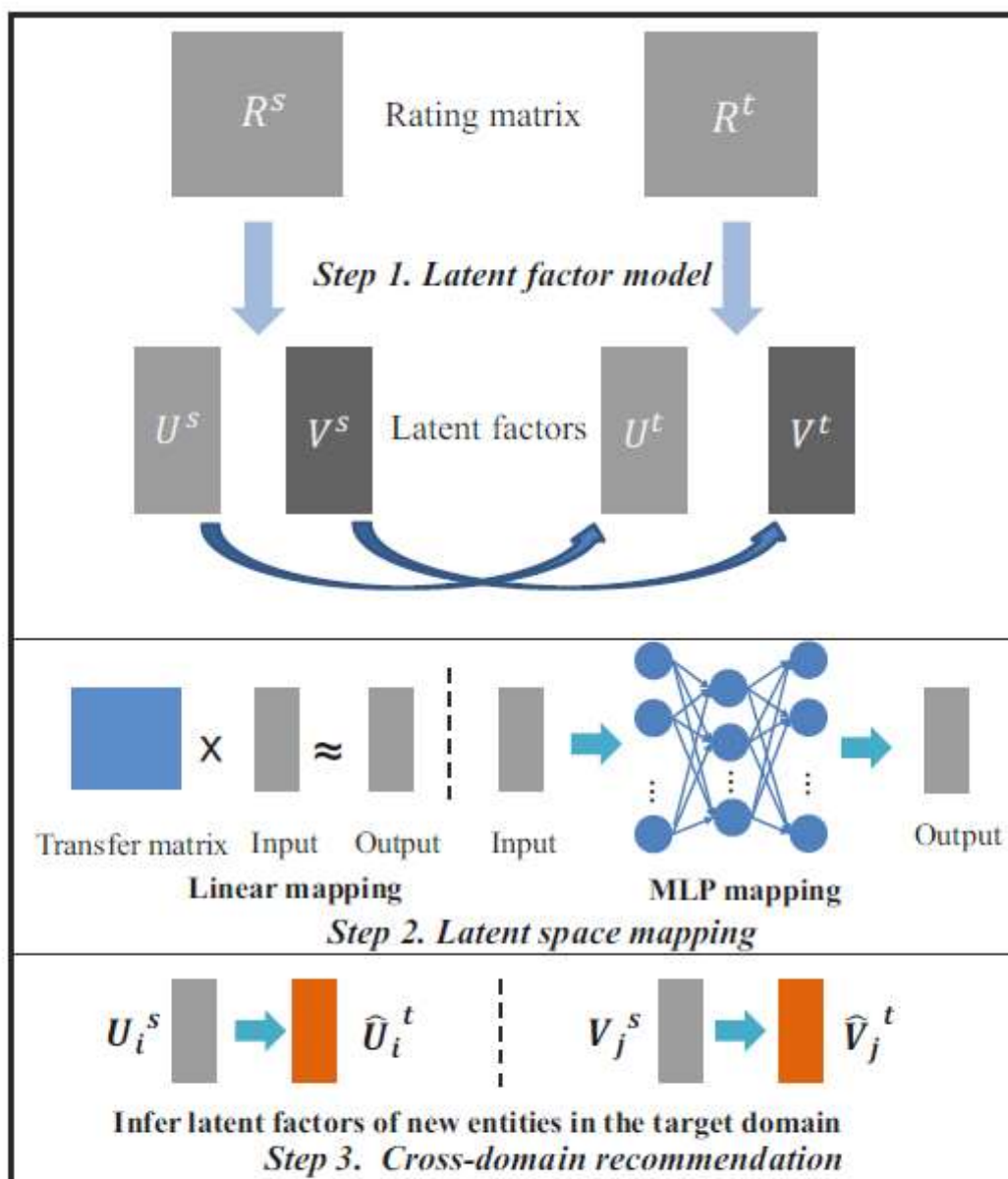


Figure 1: Illustrative diagram of the EMCDR framework.

3 EMCDR模型

在这一部分，我们使用两种不同的潜变量模型和两种不同的隐空间映射函数来完成提出的EMCDR框架。

Algorithm 1 The EMCDR framework.

Require:

Source domain R^s , target domain R^t ;
User set \mathcal{U} , item set \mathcal{V} ;

Ensure:

Make recommendation for entities in the target domain:
 $\{u \in R^s \& u \notin R^t\} / \{v \in R^s \& v \notin R^t\}$;

Latent Factor Model

- 1: Learn $\{U^s, V^s\}$ from R^s ;
- 2: Learn $\{U^t, V^t\}$ from R^t ;

Latent Space Mapping

- 3: Learn the mapping function $f_U(\cdot)$ ($f_V(\cdot)$) by users across domains

Cross-domain Recommendation

- 4: Get affine factors \hat{U}^t (\hat{V}^t) of target users
 - 5: Make recommendation for target users (items)
-

3.1 潜变量模型

如前文所述，EMCDR框架的第一步是要从源领域和目标领域分别学习实体的潜变量。在本文中，我们使用两种不同的模型，分别为MF（以评分为导向）和BPR（以排名为导向）。MF将评分矩阵分解为两个低维的矩阵。让 R 作为一个 $|U| \times |V|$ 的评分矩阵， K 是潜变量的维度， U 是一个 $K \times |U|$ 维的用户潜变量矩阵并且其中第 i 列表示用户 u_i 的潜变量， V 是一个 $K \times |V|$ 维的物品潜变量矩阵，其中第 j 列 V_j 表示物品 v_j 的潜变量。利用高斯噪声，某一个用户-物品对 (u_i, v_i) 的评分 R_{ij} 被观测到的可能性建模为：

$$p(R_{ij}|U_i, V_j; \sigma^2) = N(R_{ij}|U_i^T V_j, \sigma^2)$$

其中 $N(x|\mu, \sigma^2)$ 是平均值为 μ ，方差为 σ^2 的高斯分布的概率密度函数。

MF方法在已有的评分矩阵的基础上最大化同分布的可能性。

$$\min_{U, V} \left(\sum_i \sum_j I_{ij} \cdot \|R_{ij} - U_i^T V_j\|_F^2 + \lambda_U \sum_i \|U_i\|_F^2 + \lambda_V \sum_j \|V_j\|_F^2 \right)$$

如果用户 i 对物品 j 有评分的话指示变量 $I_{ij} = 1$ ，否则的话， $I_{ij} = 0$ 。 $\|\cdot\|_F^2$ 表示范数， λ_U 和 λ_V 是正则化项的系数。

不同于MF基于目标函数优化评分，BPR采用了一种选择顺序的方法基于目标函数来优化排名。BPR首先创建一个训练集 $D: U \cdot V \cdot V$ ，它是由原始评分矩阵中的偏好对组成的。

$$D := \{(u_i, v_j, v_l) | R_{ij} > R_{il}\}$$

给定潜变量 U_i, V_j, V_l ，偏好对 (R_{ij}, R_{il}) 的可能性被建模为

$$p(R_{ij} > R_{il}) = \sigma(U_i^T V_j - U_i^T V_l)$$

其中 $\sigma(\cdot)$ 是sigmoid函数。BPR之后优化下面这个来源于潜变量后验概率的目标函数

$$\min_{U, V} \left(\sum_{(u_i, v_j, v_l) \in D} -\ln \sigma(U_i^T V_j - U_i^T V_l) + \lambda_U \sum_i \|U_i\|_F^2 + \lambda_V \sum_j \|V_j\|_F^2 \right)$$

MF和BPR模型的参数可以对目标函数的随机梯度下降来优化。MF或BDR模型所学到的潜变量可以被看做是在隐空间中的坐标集。之后，我们使用这些坐标作为一个隐式特征来学习一个跨领域的映射函数。对于MF模型，给定用户-物品对 (u_i, v_j) 的评分 R_{ij} ，更新规则是

$$U_i \leftarrow U_i - \eta \cdot \{(R_{ij} - U_i^T V_j) V_j + \lambda_U U_i\}$$

$$V_j \leftarrow V_j - \eta \cdot \{(R_{ij} - U_i^T V_j) U_i + \lambda_V V_j\}$$

其中 U_i 和 V_j 分别代表用户 u_i 和物品 v_j 的潜变量， λ_U 和 λ_V 是正则化项，并且 η 是学习率。

对于BPR模型，首先要创建一个由偏好对组成的训练集。之后，在学习步骤中，给定一个偏好对 $R_{ijl} = \{R_{ij} > R_{il}\}$ ，更新的规则是

$$\Theta \leftarrow \Theta + \eta \cdot \left(\frac{e^{-\hat{x}_{ijl}}}{1 + e^{-\hat{x}_{ijl}}} \cdot \frac{\sigma}{\sigma \Theta} \hat{x}_{ijl} - \lambda_{\Theta} \Theta \right)$$

在这里， $\Theta \in \{U_i, V_j, V_l\}$ ， $\hat{x}_{ijl} = U_i^T V_j - U_i^T V_l$ ， λ_{Θ} 是潜变量的正则化项， η 是学习率。

3.2 隐空间映射

利用潜变量模型，我们可以获得源领域和目标领域中的用户潜变量和物品潜变量 $\{U^s, V^s, U^t, V^t\}$ 。我们假设可以通过一个映射函数来获取领域之间的关系。在下文中，我们提出两种不同的映射函数：一种是线性函数，另一种是基于多层感知机的非线性函数。

为了获得映射函数，我们可以将学习的过程形式化为一个监督回归问题。特别的，我们最小化映射损失来获得映射函数。

$$\min_{\theta} \sum_{u_i \in U} L(f_U(U_i^s; \theta), U_i^t)$$

其中 $L(\cdot, \cdot)$ 是定义在源领域和目标领域对应的特征向量上的损失函数。因为映射函数的输入和输出都是多维度的数值向量，我们选择平方误差作为损失函数。

相似的，物品映射函数的损失问题也可以被表达为

$$\min_{\theta} \sum_{v_j \in V} L(f_v(V_j^s; \theta), V_j^t)$$

3.2.1 线性映射

隐空间映射问题可以使用线性映射的问题来解决。映射函数被定义为转移矩阵 M 。以用户为例，我们的目标是寻找一个 M 使得对于所有跨领域的用户 $M \times U_i^s$ 近似于 U_i^t 。实际上， M 可以通过如下的优化问题来学习

$$\min_M \sum_{u_i \in U} L(M \times U_i^s, U_i^t) + \Omega(M)$$

其中 $L(\cdot, \cdot)$ 是损失函数而 $\Omega(M)$ 是转移方程的正则化项。

3.2.2 基于MLP的非线性映射

在本研究中，我们还使用了MLP来处理隐空间匹配问题。采用MLP作为映射函数有如下优点。首先，因为输入和输出项都是K维向量，MLP可以方便的处理这种输入-输出结构。其次，MLP是一个非线性转换，它比线性映射函数更加灵活。最后，MLP可以使用后向传播来优化。特别的，这个优化问题被形式化为

$$\min_{\theta} \sum_{u_i \in U} L(f_{mlp}(U_i^s; \theta), U_i^t)$$

其中 $f_{mlp}(\cdot; \theta)$ 是MLP映射函数， θ 是参数，包括层之间的权重矩阵和偏差项。

为了获得MLP的映射函数，在本文中，我们采用随机梯度下降来学习参数。通过在训练集上的循环，更新MLP的参数。这个反向传播的算法被用于计算参数的梯度。这个过程会一直进行到模型收敛。

3.3 跨领域推荐

给定目标领域的用户/物品，没有足够的信息我们就没有办法预测出一个可靠的潜变量来做出推荐。然而，利用在源领域中学习到的潜变量以及源领域和目标领域的映射函数，我们可以获得一个仿造的目标领域潜变量。例如，在目标领域有一个用户 u_i ，我们可以获得它的仿变量 \hat{U}_i^t 通过如下等式

$$\hat{U}_i^t = f(U_i^s; \theta)$$

其中 U_i^s 是源领域的潜变量， $f(\cdot; \theta)$ 是映射函数。之后，我们就可以依据这个伪变量来做出推荐。

4 实验

4.1 实验准备

- 数据集。我们使用两个真实的数据集来进行测试。第一个数据集，MovieLens-Netflix，它包含来自于两个公开基准数据集Netflix Prize和MovieLens的用户和对齐（aligned）电影。根据IMDB上的信息，超过5000部电影被Netflix和MovieLens共享。用户在这两个平台上是十分不同的，它形成了跨领域的物品共享情景。第二个数据集是从在线社交平台豆瓣上爬取的，在这个网站上，用户给电影和书籍进行评分，这两个数据集的统计指标如表1和2所示，在本文中，我们把MovieLens视作源领域，Netflix视作目标领域作为物品分享情景。在用户共享情景中，电影被视作源领域，书籍被视作目标领域。
- 实验准备。为了评估这个框架在跨领域推荐任务上的正确性和效率，我们随机目标领域中的一部分实体的全部评分信息，并利用他们作为跨领域冷启动实体来进行推荐。为了实验的严密性，我们设置不同比例的冷启动实体，10%，20%，30%，40%以及50%。此外，应为不同的冷启动实体或许会影响最终的推荐结果，我们把样例用户/物品重复L次来产生差分集。我们报告这些L差分集平均结果和标准差。特别的，我们在本实验中，设置L为10。潜因子的维度K被设置为20，50和100，学习率和正则化系数通过在训练集上的5折交叉验证来确定。对于线性映射函数，置换矩阵的大小为 $K \times K$ ，正则化系数 λ_M 被设置为0.01；对于MLP映射函数，我们选择MLP的结构为一层隐含层，输入和输出的维度被设置为K，隐藏层结点的数目被设置为 $2 \times K$ ，权重和偏差参数根据[Glorot and Bengio, 2010]中的方法来初始化。我们设置的批（minibatch）的大小为16并且不使用动量。最终，使用一个tan-sigmoid函数作为激活函数。