Low-rank Linear Cold-Start Recommendation from Social Data

会议信息：AAAI-2017

文章链接：http://users.cecs.anu.edu.au/~akmenon/papers/loco/loco-paper.pdf

摘要

推荐系统的冷启动问题是如何向系统中没有任何历史记录的新用户推荐，而我们常用的协同过滤算法是完全依赖用户与物品之间的历史交互信息的，这成为了一个挑战。近期相关的工作表明，有效利用社交信息（如用户的好友关系、公共主页喜好）能够很好地缓解这一问题。本文将三种已有的冷启动模型总结推广为一种线性的基于内容的模型，在此基础上，提出了 LoCo 模型，实验表明此模型要优于 state-of-the-art 冷启动推荐模型。

介绍

1992 年，协同过滤方法一经提出就经久不衰，却面临着冷启动问题，并且在实际应用中更为严重。为此，开始引入关于用户的外部信息和特征。基于内容的过滤方法，则是从外部补充信息中获取用户的喜好特征，在实际应用中，也常常作为混合式推荐系统的一部分。基于此，这些外部信息是否准确，而如果是准确的，怎么高效地利用就是我们研究的方向。

通常的用户的外部信息是来源于社交网络的，比如用户的好友关系、他喜欢的某些主页，这些都能够用来缓解冷启动问题。有许多方式把这些信息融合进近邻推荐模型、矩阵分解推荐模型中，并且取得了一定的效果，但还存在一些问题：

- 基于近邻的推荐方法没有一个显式的优化目标函数，因此缺乏模型上的直观可解释性，也可能带来了次优化的结果
- 矩阵分解，或称隐特征的方法，在实现过程中需要经过多次耗时的迭代来优化一个**非凸**的目标函数，需要手动调参的数量也很多，还隐含着维数灾难

本文提出了一种高效的、准确的基于学习的方法，利用社交数据来解决冷启动问题。主要的贡献有：

- 将已有的三种冷启动模型总结为一个线性基于内容模型的特例，通过比较分析了已有模型中各自存在的问题
- 提出具有更一般形式的 LoCo 模型，通过三种手段解决已有模型的不足：
    - 使用多元线性回归从社交数据中学习一个更优的用户喜好权重
    - 使用低阶参数化方法学习回归权重（隐层表示）来应付社交数据的高维度
    - 使用可高度规模化的低阶分解方法，randomised SVD（参见 Halko, Martinsson, and Tropp 2011），可以适应高维用户社交数据

我的看法

本文并未涉及到深度学习的方法，算是在2017年环境中的一股清流了，而且读完感觉叙述很清晰，前人工作也总结的很不错，给人一种很好的条理感，这是我们自己写文章时要学习的。本文关于相关工作的总结十分出彩，很有作者自己的观点和相关的数学推导，并且在这之上抽象出了一个更为general的模型，还针对这些特例方法在这普适方法中做了一些优化。实验部分的分析比较充分，而且在现在推荐系统关注的另一方面：效率问题上，通过降维提高了运行效率。

不使用深度学习，一方面，我觉得可以根据社交数据，通过用户的社交相似度，据此加权平均计算出新用户的特征向量，相当于新用户被拆解为其他老用户的特征组合，就像一个人身上有很多其他用户的影子一样，这是一种保守的特征估计。那么这个新用户的特征向量和物品的向量表示空间还是一致的，可以回到简单的内积预测喜好。

那么可以在 model-based filtering 方法上加上一些新的限制，我觉得还是可以在最终矩阵分解的优化目标中加上新的损失，这属于 collective matrix factorization 的做法，从打分信息中获取的用户特征作为输入，去优化社交数据中用户的特征，反过来社交数据有作为补充信息的输入用户的特征，与物品特征联系起来做预测。

从文中所用的数据来看，用的是用户与他们喜欢的页面信息，也是一个矩阵，那么我希望能够学习出一个用户的特征表示，借助这个社交视角下的表示，可以完全还原出原本的社交矩阵信息，那么这个表示的质量还是比较好的，借鉴了自编码器的思想，那么这个社交视角下的表示怎么回到原本打分喜好的特征空间中，上一段的论述是直接从打分的特征空间中迁移过来的用户特征向量，其实还可以加入一个限制（损失函数控制）：社交视角下的表示与打分视角下的表示尽可能接近，究竟哪一个好或者没什么差别，可能还是和数据相关。

问题描述

用户数 U、商品数 I，喜好（购买）0-1矩阵：$R \in \{0,1\}^{U\times I}$，外部辅助信息：$X \in \mathbb{R}^{U\times P}$，表示用户对某个网页的喜欢与否，冷启动问题下，热启动用户作为训练数据，在测试数据中的用户都不在训练数据中出现，就是冷启动用户，分别用 tr 和 te 表示训练集 R_{tr} 和测试集 R_{te}，外部社交信息也可以划分为 X_{tr} 和 X_{te}。

事实上，在模型学习过程中 $R_{te} \equiv 0$，最终的目的是计算估计值 $\hat{R}_{te}$。已有的相关做法如下：

1. 社交相似近邻：$\hat{R}_{te} = X_{te} \odot X_{tr}^T \star R_{tr} = S \star R_{tr}$，这里通过某种矩阵运算（内积），由 S 表示用户之间的相似度度量，完成一个转化过程；
2. Collective Matrix Factorization 方法：通过一个统一的最优化目标：
$$\min_{U,V,Z} \|R - UV\|_F^2 + \mu\|X - UZ\|_F^2 + \Omega(U,V,Z)$$
求解出对应的同一空间下的特征表示矩阵，$\hat{R}_{te} = U_{te}V$；
3. 特征映射方法，BPR-LinMap，假设 $U_{te} = X_{te}T$，且 $U_{tr} = X_{tr}T$，通过共享的 T 的语义空间来映射；
4. 类似分类法：学习一个抽象的函数 f，$\hat{R}_{te} = f(X_{te})$。

统一模型

我们可以抽象化为一个线性模型：$\hat{R}_{te} = X_{te}W$，总结了前三种做法都可以用 W 显式表达为：

Method	Weight W
Social neighbourhood	$\mathbf{X}_{\mathrm{tr}}^T\mathbf{R}_{\mathrm{tr}}$
CMF	$\mathbf{Z}_*^T(\mathbf{V}_*\mathbf{V}_*^T+\mathbf{Z}_*\mathbf{Z}_*^T)^{-1}\mathbf{V}_*$
BPR-LinMap	$(\mathbf{X}_{\mathrm{tr}}^T\mathbf{X}_{\mathrm{tr}})^{-1}\mathbf{X}_{\mathrm{tr}}^T\hat{\mathbf{R}}_{\mathrm{tr}}$
Linear regression	$(\mathbf{X}_{\mathrm{tr}}^T\mathbf{X}_{\mathrm{tr}})^{-1}\mathbf{X}_{\mathrm{tr}}^T\mathbf{R}_{\mathrm{tr}}$
LoCo	$\mathbf{V}_K(\mathbf{V}_K^T\mathbf{X}_{\mathrm{tr}}^T\mathbf{X}_{\mathrm{tr}}\mathbf{V}_K)^{-1}\mathbf{V}_K^T\mathbf{X}_{\mathrm{tr}}^T\mathbf{R}_{\mathrm{tr}}$

Table 1: Summary of choice of weight matrix $\mathbf{W}$ in linear model $\hat{\mathbf{R}}_{\mathrm{te}} = \mathbf{X}_{\mathrm{te}}\mathbf{W}$ that yields various cold-start recommenders. See text for definitions of $\mathbf{Z}_*, \mathbf{V}_*, \mathbf{V}_K$.

分析这三种方法的利弊

1. 基于社交的近邻方法，W 的选择并没有一个显式的、有实际意义的优化目标，因此不同视角下的特征关系无法体现，存在 **欠拟合** 问题；
2. BPR-LinMap，对于高维的辅助信息 X，相关的实验表明，通过高维回归等方法学习出的映射矩阵 T 很容易 **过拟合**；
3. CMF，适合高维社交信息，但是规模上不好扩展，且需要手动调参的数量过多；

总结就是，这些方法无法同时满足：

- 有较好的可解释性和明确、有意义的优化目标；
- 不同的特征之间具有某种联系
- 适合高维特征数据输入
- 训练和调参工作上的规模可扩展

LoCo 模型

该模型是上述三种模型的统一表示（根据 W 的不同选择有不同的解释），因此我们有一个多元线性回归的目标函数：

$$\min_W \|R_{tr} - X_{tr}W\|_F^2 + \frac{\lambda}{2}\|W\|_F^2$$

W 的显式解形式 Table 1中的 Linear Regression 已经给出。

为了能够使得上面的统一表示能够适应高维社交数据，我们需要限制 W 的秩比较小，可是这样带来的问题是，上述优化目标非凸，同时效率会比较低下。

我们有 rank-K SVD 分解 $X_{tr} \approx U_K S_K V_K^T$，令 $W = V_K Z$，显然 $rank(W) < K$，优化目标可化为：

$$\min_{Z} \|R_{tr} - X_{tr} V_K Z\|_F^2 + \frac{\lambda}{2} \|Z\|_F^2$$

最终求解的 W ，由 Table 1中的 LoCo 一行给出。

有了这样的显式解形式，那么最终就是选择一种近似的计算方式，借助 randomised SVD 可以解决这个问题。至此，LoCo 模型的提出，亮点就在于：

- 线性模型的低阶化参数与权重
- 使用 randomised SVD 适应大规模的、高维的社交网络数据
- 用统一的多元线性回归问题求解参数

作者通过一个 subsection 论述了 LoCo 模型的时间复杂性，也在实验结果部分给出了时间性能比较。

实验结果与分析

数据集：Ebook，Flickr，Blogcatalog 和 LastFM

其他模型主要是作者在稳重提到的三种方法，实验结果是 LoCo 模型是优于其他方法的，从理论和实验上取得了一致的结果。关于一些参数的分析（如隐层表示的维数设置，训练数据的规模）等也都是符合我们常识的。值得一提的是，在社交网络中，如果用户的数据越丰富，那么在冷启动用户的推荐结果上会更好，同样符合我们的直觉：冷启动用户提供的外部信息越多，我们对他的特征学习就更好。

时间比较上，LoCo 在最大数据集 Ebook 上的运行时间在分钟级别，只有社交邻近方法在秒级，其他方法都是小时级别，也算是 LoCo 模型在降维方面做得比较好，运算的开销和模型的设计比较巧妙的体现。

结论与（我能想到的）改进

本文综合已有的方法，提出一种更为一般形式的线性基于内容的模型来解决推荐系统中冷启动用户的问题。作者认为的改进一方面是使用非线性模型，我觉得可以引入神经网络模型完成非线性学习；另一方面是流数据的处理，即实时推荐系统，这是基于用户的喜好很可能随着时间的流逝而改变，也是需要考虑的一个问题，通常的做法是，时间久远的数据计算损失时设计一个较小的正则化参数，可是作者这里的做法是直接通过显式的表达式求解，所以即便能找到显式求解形式，可能形式上会更为复杂，计算的开销也不得而知。如果不考虑一定要低阶分解的方法，即采用通过梯度下降来更新权重，那还是会好操作一些。

至于如何提升 LoCo 模型的运行时间，我觉得已经可以接受了，可能还需要对模型进行进一步的抽象与近似求解过程。