

067 | 基于隐变量的模型之二：基于回归的矩阵分解

2018-03-07 洪亮劼

AI技术内参

[进入课程 >](#)



讲述：初明明

时长 07:15 大小 3.33M



本周我们主要来分享“矩阵分解”的点点滴滴，这是过去 10 年里推荐系统中最流行的一类模型。周一我们讨论了这类方法中最基础的基于隐变量的矩阵分解。这类模型的优势是显式地对用户和物品信息中的隐含结构进行建模，从而能够挖掘更加深层次的用户和物品关系。矩阵分解的流行起源于 10 年前的 Netflix 大赛，当时各类矩阵分解模型都在实际数据中起到了很好的效果。

今天我们要分享的模型，叫作“**基于回归的隐变量模型**”（Regression-based Latent Factor Model）。这是在基本矩阵分解的基础上衍生出来的一类模型。

基本矩阵分解的问题

我们先来看看基本矩阵分解模型的潜在问题。

首先，我们来回顾一下矩阵分解的基本表达。假设我们在对用户和物品的评分进行建模。对于每一个用户，用一个向量来表达其对于所有可能物品的评分，把所有用户的向量堆积起来，就可以得到一个矩阵。这个矩阵的每一行代表一个用户，每一列代表一个物品，每一个交叉的元素代表某一个用户对于某一个商品的评分。对于每一个用户和物品的隐向量都要少于原有的物品数目，因此，我们也说矩阵分解的模型实现了“降维”（降低建模维度）的目的。

虽然矩阵分解的模型对于挖掘用户和物品的内在联系有比较强的作用，但是这类模型的劣势也十分明显。

第一，矩阵分解的矩阵仅仅是对用户和物品的喜好进行了“编码”（Encode）。我们之前在解释基于内容的推荐系统时说过，对于一个复杂的工业级推荐系统来说，有很多灵感或者直觉，都很难仅仅依赖用户和物品的喜好来捕捉到。有大量的信号，无法很好地被融合到这个矩阵分解的模式里。因此，矩阵分解虽然是不错的“独立”模型（Standalone），但在融合多种不同的推荐元素方面，表现却很一般。

第二，矩阵分解的核心是学习用户的隐向量和物品的隐向量。原则上，这两类隐向量的学习仅能通过训练过程获得。

什么意思呢？就是说，如果在训练集合中有 100 个用户，200 个物品，那么通过训练，我们可以学习到这 100 个用户和 200 个物品对应的隐向量，用于以后的预测或者排序。但我们无法获得新来用户或者新来物品的隐向量了，因为这些用户和物品并不在训练集里。

在推荐系统中，这种情况就叫作不能处理“冷启动”（Cold Start）问题，也就是不能处理“冷”用户和“冷”物品。这个特点是好是坏取决于具体的应用。但是，在有些应用的场景下，这是灾难性的一种特性。比如，新闻网站推荐文章，一般来说新闻文章滚动得都很快，随时随地都可能有新文章涌入，每过几天，整个网站的文章可能都被换掉，在这样的场景下，直接使用矩阵分解就会有问题。

矩阵分解的基本模型存在这两个问题，那有没有什么办法可以扩展矩阵分解的形式，改善在这两种情况下的处理呢？

基于回归的矩阵分解

基于回归的矩阵分解，其核心思路其实非常简单直观。矩阵分解不过是定义了一种从用户隐变量和物品隐变量到可以观测到的评分，这一显变量的产生过程。如果我们掌握了这么一个

思路，那么，基于回归的矩阵分解就很容易理解了。

首先，有一组用户特性和物品特性来表述每一个用户和物品。这些特性不是隐变量，是显式表达的特性。用户特性比如用户的年龄、性别、经常出现的地区、已经表达了喜好的类别等；物品特性比如物品的种类、描述等等。这两组显式的特性就是为了解决我们刚才说的第一个问题，矩阵分解无法抓住更多的信号。而在这里，我们明确定义这两组信号，一组来自于用户，一组来自于物品，希望能够通过这些信号抓住更多有价值的信息。

解决了这一步后，现在我们有两个独立的，看上去不相关的部分。一个是基于矩阵分解的部分，这一部分是分解一个已知的评分矩阵，从而学习到用户和物品的隐向量。另外一个部分，就是用户特性和物品特性，暂时还不知道如何与矩阵分解关联上。

基于回归的矩阵分解假定，用户的隐向量，其实是从用户的显式特性变换而来的。同理，物品的隐向量，其实是从物品的显式特性变换而来的。而这两种变换，本质上就是两个回归模型 (Regression Model)。就是说，**我们建立一个从显式特性到隐向量的回归模型，使得隐向量受到两方面的制约：从评分矩阵的分解得来的信息和从显式特性回归得来的信息。**这样，我们就把这两部分信息嫁接到了一起。

把这两部分信息嫁接到一起的好处就是，我们不再怕“冷启动”了。或者说，在有一部分“冷启动”的情况下，这样的模型可以处理得更好。原因就是使用了显示特性来回归隐向量。对于新用户和新物品而言，我们虽然没有直接得到他们的隐向量，但是有这些用户或者物品的显式特性，而这些显式特性可以帮助我们估计到新用户和新物品的隐向量，即使我们并没有它们的交互信息。这就是基于回归的矩阵分解的强大之处。

我们还可以从贝叶斯的角度来理解基于回归的矩阵分解。把用户的隐向量和物品的隐向量看作是两个随机变量。我们可以认为这些随机变量加上了先验概率分布。只不过，这个先验概率分布的均值不是我们经常使用的 0，而是一个以回归函数的结果为均值的高斯分布，这个回归函数就是我们由显式特性得到的。本质上，我们认为显示特性的某种变换成为了隐向量的先验信息。

学习基于回归的矩阵分解

基于回归的矩阵分解，其实就相当于我们定义了“**层次的贝叶斯模型**” (Hierarchical Bayesian Model)。模型的求解就变得更加复杂。

有一种学习这个模型的简单思路。第一，针对已知的用户和物品，先通过简单的矩阵分解学习到用户和物品的隐变量。在这个部分里，用户的显式信息并不参与学习。第二，学习好了这部分变量以后，再来学习从显式信息到隐变量的回归部分。这部分基本上可以当作线性回归。当然，用这种两个阶段的学习流程，并不能保证学习到的参数是最优的，这仅仅是一种简化的学习方法。

如果走比较正统的学习过程，这样的模型需要采用“**期望最大化**” (Expectation Maximization) 的流程。也就是说，我们先用一组随机的参数来决定回归的部分，学习到最佳的隐向量，然后再根据隐向量更新回归部分的参数。因篇幅的原因我们就不展开细节了。不过，即便如此，我们也仅仅能够学习到一个局部解。

小结

今天我为你讲了隐变量模型中基于回归的矩阵分解。

一起来回顾下要点：第一，我们简要介绍了矩阵分解的一些问题；第二，我们详细介绍了基于回归的矩阵分解的基本思路，以及这样的模型如何解决了传统矩阵分解关于“冷启动”的难题；第三，我们简要地讲解了如何求解基于回归的矩阵分解里的参数。

最后，给你留一个思考题，基于回归的矩阵分解的短板是什么？

欢迎你给我留言，和我一起讨论。

AI 技术内参

你的360度人工智能信息助理

洪亮劼

Etsy 数据科学主管
前雅虎研究院资深科学家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 066 | 基于隐变量的模型之一：矩阵分解

下一篇 068 | 基于隐变量的模型之三：分解机

精选留言 (5)

 写留言



惜心 (伟祺...)

2018-04-02



相当于先用矩阵分解得到隐变量向量 作为y

用用户和物品的显变量作为数据x

拟合求出 theta

在用的时候 利用新来用户物品显变量信息点乘参数求出一个向量

后面求相似度就和矩阵分解出的物品矩阵和用户矩阵类似了

展开



米乐乐果

2018-03-25



希望老师可以顺带指出比较经典的论文

展开 ∨



林彦

2018-03-08



基于回归的矩阵分解需要在实践中尝试不同的回归模型。回归模型中的特征工程在实践中要结合领域知识，这里要推断和隐式特征的更好的关系不太容易。在大规模数据中这种方法的收敛性，找到最优解是不是有一定难度。

展开 ∨



微微一笑

2018-03-07



文章的更新顺序不是按照目录来的吧？推荐系统架构剖析的文章找不到

展开 ∨



Peter

2018-03-07



老师，请问这块有没有一些资料或者论文可以看呀

展开 ∨

拼课微信：171614366!