

069 | 高级推荐模型之一：张量分解模型

2018-03-12 洪亮劼

AI技术内参

[进入课程 >](#)



讲述：初明明

时长 07:13 大小 3.31M



上周我们分享了推荐系统中矩阵分解的点点滴滴，简单复习一下讨论过的三个模型。

第一，“基于隐变量的矩阵分解”，其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。

第二，“基于回归的隐变量模型”，这是在基本的矩阵分解基础上衍生出来的一类模型。这种基于回归的矩阵分解模型把显式特性和隐变量结合起来，对解决“冷启动”问题有一定作用。

第三，“分解机”是由基于回归的隐变量模型衍生出来的，并且吸纳了基于信息的推荐系统的一些观点，成为了一种可以融合多种信息的强有力的工具。

这周，我们跟随着这个脚步，进一步来讨论一些比较高级的模型，看这些模型如何抓住更多的用户和物品之间的关系。

今天，我们先来聊一种叫作“**张量分解**”（Tensor Factorization）的模型。

为什么需要张量分解

在我们探讨张量分解的一些基本思想之前，先来看一看，推荐系统在什么场景下需要张量分解。

我们还是要从矩阵分解说起。矩阵分解的核心思想，是用矩阵这种数据结构来表达用户和物品的相互关系。这里，我们一般谈论的都是一些最简单的关系，例如评分、点击、购买等（本文我们依然只是讨论评分）。在这种二元的模式下，矩阵就是最好的表达用户和物品之间关系的数据结构。

然而，在真实的场景中，用户和物品的关系以及产生这种关系的周围环境是复杂的。一个矩阵并不能完全描述所有的变量。例如，用户对于某个物品的评分是发生在某个地点、某个时间段内的。这种所谓的“上下文关系”（Context）往往会对评分产生很大影响。遗憾的是，一个矩阵无法捕捉这样的上下文关系。

我们之前讨论过的“基于回归的矩阵分解”和“分解机”，本质上都是在某种程度上绕开这个问题。采用的方法就是，依然用矩阵来表达二元关系，但是把其他信息放进隐变量中，或者是采用基于信息的推荐系统的思路来得到相关信息的建模。

除了这种思路，还有没有别的方法，可以把上下文关系融入到对用户和物品的建模中去呢？

这时“张量”就该上场了。

从本质上来说，**张量就是矩阵的推广**。我们可以这么理解，**矩阵是对二维关系建模的一种工具；而张量，就是对 N 维关系的一种建模**。在二维关系中，用户和物品的评分是唯一能够被建模的变量；而到了 N 维关系中，理论上，我们可以对任意多种上下文关系进行建模。

比如，我们刚才提到的时间，就可以组成一个三维的张量，分别为用户、物品和时间。然后，在这个三维的张量中，每一个单元代表着某一个用户对于某一个物品在某一个时间段的评分。

基于张量分解的推荐模型

我们已经讲了张量的作用。那么，如何使用张量来进行推荐模型的建模呢？

我们还是用刚才所说的“用户、物品和时间”作为例子。在这个三维的张量里，每一个元素代表的是用户对于物品在某个时间段上的评分。那么，根据矩阵分解的思路，我们怎么来对这个张量进行分解呢？

遗憾的是，张量的分解要比矩阵分解更为复杂。与矩阵分解不同，张量分解至少有两种不同的形式，这两种形式会引导出不同的分解模型和算法。

第一种分解模式叫 CP 分解 (CANDECOMP/PARAFAC)。

拿我们刚才所说的三维张量来讲，CP 分解是把一个三维张量分解为三个矩阵。具体来说，比如我们的三维张量是 N 维用户乘以 M 维的物品乘以 R 维的时间段。那么，分解出来的三个矩阵就分别为 N 维乘以 K 维的用户矩阵， M 维乘以 K 维的物品矩阵，以及 R 维乘以 K 维的时间矩阵。这三个矩阵中的每一个向量都代表某一个用户、某一个物品和某一个时间段。 K 在这里是一个参数，类似于矩阵分解中的隐变量的维度，我们也就可以把这个 K 理解成为隐变量的维度。

那么在原始的三维张量中，某一个元素就是这三个矩阵的某三个向量对应元素乘积相加的结果。CP 分解的一大好处就是，分解出来的三个矩阵的隐变量维度是一样的，这也就减少了需要调整的参数个数。

第二种分解模式叫作 HOSVD 分解 (High Order Singular Value decomposition)。

这种分解和 CP 分解最大的不同就是分解出来的三个矩阵的维度不再一样。也就是说，在我们之前的例子中，用户矩阵的维度可以是 N 乘以 A ，物品矩阵的维度是 M 乘以 B ，时间段矩阵的维度是 R 乘以 C 。当然，这样就无法还原之前的 N 乘以 M 乘以 R 的三维张量了。

于是在技术上，还需要乘以一个 A 乘以 B 乘以 C 的小张量才能对原始数据进行复原。所以，通俗地讲，HOSVD 分解就是把一个三维的张量，分解成为三个矩阵和一个更小的张量的乘积。

那 HOSVD 相比 CP 分解有什么好处呢？好处自然就是给不同的数据以不同的自由度，因为不再限制用户、物品和时间段都必须有一样的维度。但同时也带来了难点，那就是有了更

多的“超参数”需要调整。

值得一提的是，我们仅仅是讲了讲张量分解的一个基本思路。在实际的运作中，还有一个重要的步骤，那就是**设计目标函数**，或者说是**定义损失函数**。在一般的分解过程中，我们可以定义“**平方差**”（Squared Loss），也就是原始数值和预测数值之间的平方差来作为损失函数，当然也可以使用其他的损失函数，这一点和矩阵分解是一样的。

求解张量分解

虽然张量是对矩阵的自然延伸，张量分解是矩阵分解的某种延伸，但是求解张量分解则是一个相对比较难的问题。

一种比较简单的模式依然是使用“**随机梯度下降**”法（SGD, Stochastic Gradient Descent），也就是把张量的分解问题看作是一个一般的优化问题。关于这种方法的细节我在这里不详细讲解了，感兴趣的话，建议阅读参考文献 [1]，这是一篇很经典的论文，是第一次把张量分解使用在推荐系统中。

另外一种方法，也是在矩阵分解中可以使用的，叫作 **ALS**（Alternating Least Square）方法。这种方法则是在优化每一轮的时候，按住所有其他的矩阵变量不动，单独优化一个变量。举例来说，如果我们要优化用户矩阵，那么我们就按住物品矩阵和时间段矩阵不动，不去更新里面的参数，而仅仅更新用户矩阵里面的参数。同理，在下一轮的优化中，可以仅仅更新物品矩阵里的参数。这种方法也是行之有效的一种优化方法。

小结

今天我为你讲了推荐系统的一个高级模型：张量分解。张量分解模型主要用来对上下文的信息进行建模。

一起来回顾下要点：第一，我们简要介绍了为什么需要张量分解；第二，我们详细介绍了张量分解的基本原理；第三，我们简要地讨论了如果求解张量分解。

最后，给你留一个思考题，从概念上来看，用张量分解对上下文信息进行建模的最大问题是什么？

欢迎你给我留言，和我一起讨论。

参考文献

1. Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. [Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering](#). Proceedings of the fourth ACM conference on Recommender systems (RecSys '10). ACM, New York, NY, USA, 79-86, 2010.



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 068 | 基于隐变量的模型之三：分解机

下一篇 070 | 高级推荐模型之二：协同矩阵分解

精选留言 (2)

写留言



Jack_Saini...

2018-03-17

稀疏性在从矩阵扩展到张量后变得更为严重

展开 ∨

1



林彦

2018-03-12

👍 1

我能想到的问题：张量分解所要用到的“上下文关系”（Context），如时间，本来是一个连续变量，怎么映射成合理的离散变量。某些时间段没有评分数据如何处理？

展开 ▼