

070 | 高级推荐模型之二：协同矩阵分解

2018-03-14 洪亮劫

AI技术内参

[进入课程 >](#)



讲述：初明明

时长 06:40 大小 3.06M



周一我们讨论了“张量分解”模型。这种模型的特点是能够把不同的上下文当作新的维度，放进一个张量中进行建模。虽然张量分解是矩阵分解在概念上的一种直觉扩展，但其在现实建模的过程中有很大难度，最大的问题就是张量分解的求解过程相对比较复杂，不同的分解方法又带来不同的建模选择。

今天，我们来看另外一种思路，来解决融合多种渠道信息的问题，这就是**协同矩阵分解** (Collective Matrix Factorization) 。

为什么需要协同矩阵分解

在解释什么是协同矩阵分解之前，我们先来看一看为什么需要这样一种思路。我们还是需要回到矩阵分解本身。

矩阵分解的核心就是通过矩阵，这个二维的数据结构，来对用户和物品的交互信息进行建模。因为其二维的属性，矩阵往往只能对用户的某一种交互信息直接进行建模，这就带来很大的局限性。

在之前的讨论中，我们看到了一系列不同的思路来对这样的基本结构进行扩展。

思路一，就是通过建立显式变量和隐变量之间的回归关系，从而让矩阵分解的核心结构可以获得更多信息的帮助。

思路二，则是采用分解机这样的集大成模型，从而把所有的特性，都融入到一个统一的模型中去。

思路三，就是我们这周已经讲到的，利用张量，把二维的信息扩展到 N 维进行建模。

这些已有的思路都各有利弊，需要针对具体的情况来分析究竟什么样的模型最有效果。

然而在有一些应用中，除了用户和物品这样很明显的二元关系以外，还有其他也很明显的二元关系，如何把这些二元关系有效地组织起来，就变成了一个有挑战的任务。

什么意思呢？比如，我们是一个社交媒体的网站，既有用户对于物品（比如帖子）的交互信息，又有用户之间的互相连接信息（谁与谁是好友等）。那么，如何来显式地表达这两类不同的二元关系呢？

在前面的思路里面可以看到，我们似乎需要选择一个主要的关系来作为这个模型的基础框架，然后把其他的信息作为补充。在这样两类关系中，选择哪一个作为主要关系，哪一个作为补充关系，就显得有一点困难了。

更进一步说，对于用户与用户之间的建模，以及用户与物品之间的建模，我们其实会有不同的模型去构造。例如，用户与物品之间的评分，往往用整数来代表评分的数值，或者是用实数来代表喜好度。而用户与用户之间的友好关系，则往往是 0 或者 1，象征是否有连接。因此，我们可能需要不同的模型对这些不同的数值进行建模。

这也就让研究人员想出了协同矩阵分解的思路。

协同矩阵分解的基本思路

协同矩阵分解的基本思路其实非常直观，那就是**有多少种二元关系，就用多少个矩阵分解去建模这些关系**。

用我们刚才所说的社交媒体的例子。如果我们有用户与用户的关系，用户与物品的关系，那我们就组织两个矩阵分解，分别来对这两种关系进行建模。最早对这个思想进行得比较完整的表述，我在文末列出了参考文献 [1]。

这里的一个核心就是，如果两个没有关系的矩阵，各自做矩阵分解，那么分解出来的信息，一般来说，是没有任何关联的。

再来看刚才的例子，如果有一个用户与用户的矩阵需要分解，然后有一个用户与物品的矩阵需要分解。那从这两个矩阵分解中，我们分别可以得到至少两组不同的**用户隐变量**。一组是从用户与用户的关系而来，一组是从用户与物品的关系而来。这两组用户的隐变量是不一样的。同时，因为两个矩阵没有关联，所以无法达到我们希望这两种关系互相影响的效果。

要想在两个矩阵分解之间建立联系，我们必须有其他的**假设**。这里的其他假设就是，两组不同的用户隐变量其实是一样的。也就是说，我们假设，或者认定，**用户隐变量在用户与用户的关系中，以及在用户与物品的关系中，是同一组用户隐变量在起作用**。

这样，虽然表面上还是两个矩阵分解，但其实我们限定了其中某一部分参数的取值范围。说得直白一些，我们认定从两个矩阵分解出来的两组来自同一个因素（这里是用户）的隐变量是完全一样的。用更加学术的语言来说，这就是**将两组矩阵分别投影到了相同的用户空间和物品空间**。

这样做的好处，自然就是对于多种不同的关系来说，我们使用“**相同隐变量**”这样的假设，可以把这些关系都串联起来，然后减少了总的变量数目，同时也让各种关系互相影响。

那么，这样的假设有没有潜在的问题呢？

一个比较大的潜在问题就是，使用同样的一组隐变量去表达所有的同类关系，这样的假设存在一定的局限性。比如上面的例子，用同样一组用户隐变量去解释用户和用户之间的关系，同时也需要去解释用户和物品之间的关系，能够找到这样一组用户隐变量其实是有一定难度的。

而在实际应用中，不同关系的数据量会有很大的差距。比如，用户和物品关系的数据总量可能要比用户与用户的多。所以，由于用户和物品关系的数据多，两个矩阵分解用的同一组用户隐变量，很可能会更多地解释用户和物品的部分，从而造成了学到的隐变量未必能够真正表达所有的关系。

对于这样的情况，自然已经有一些学者想到了对策，我们今天就不在这里展开了。

最后，需要提一下，在协同矩阵分解的场景中，学习各个隐变量的参数的过程，和一般的单个矩阵分解相比，没有太多本质性的变化。最简单的学习过程，依然是利用**随机梯度下降法** (SGD, Stochastic Gradient Descent) 去学习。只不过，每一个隐变量会存在于多个矩阵分解中，这在更新变量时增加了所需的计算量。

小结

今天我为你讲了推荐系统的另一个高级模型，协同矩阵分解，用来对不同类型的二元信息进行建模。

一起来回顾下要点：第一，我们简要介绍了为什么需要协同矩阵分解；第二，我们详细介绍了协同矩阵分解的原理、潜在问题和解法。

最后，给你留一个思考题，从概念上来看，协同矩阵分解和张量分解之间有怎样的关系？是不是所有的张量分解都可以化为多个协同矩阵分解呢？

欢迎你给我留言，和我一起讨论。

参考文献

1. Ajit P. Singh and Geoffrey J. Gordon. [Relational learning via collective matrix factorization](#). Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08). ACM, New York, NY, USA, 650-658, 2008.


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 069 | 高级推荐模型之一：张量分解模型

下一篇 071 | 高级推荐模型之三：优化复杂目标函数

精选留言 (1)

 写留言



林彦

2018-03-14



我的理解张量分解里面矩阵的值对应(1)N维或N元信息同时关联发生的事件或概率，(2)把N维完全拆开每维发生的事件或概率。协同矩阵分解里面对应的是二维或二元之间的关联，同时假定其中的某个(或某些?)维度有一些隐藏元素影响这些关联。

张量分解能拆成或组合成两两关联，我觉得不大可能所有的两两关联都能协同。协同矩...

展开 ▾