

知乎



如何解决推荐系统中的冷启动问题?

**第四范式...**

已认证的官方帐号

[关注他](#)

36 人赞同了该文章

当新用户或新项目进入内容平台时，就会出现冷启动（Cold Start）问题。

以协同过滤这样的经典推荐系统为例，假设每个用户或项目都有评级，这样我们就可以推断出类似用户/项目的评级，即使这些评级没办法调用。但是，对于新进入的用户/项目，实现这一点很困难，因为我们没有相关的浏览、点击或下载等数据，也就没办法使用矩阵分解技术来“填补空白”。

不过，研究人员已经提出了各种方法来解决冷启动问题。在这篇文章中，我们会简单介绍一些解决推荐系统中冷启动问题的方法，至于这些方法在实践工作中是否奏效，尚无定论。

[▲ 赞同 36 ▼](#)[添加评论](#)[分享](#)[喜欢](#)[收藏](#)[申请转载](#)

知乎

- 基于内容：使用诸如文本、社交网络等的辅助信息；
- 老虎机：考虑新项目中的EE问题（Exploration&Exploitation）；
- 深度学习：使用黑盒子。

详细版

基于代表性

如果没有足够的用户和项目信息，我们可以更多地依赖那些能够“代表”项目和用户的用户。这就是基于代表性的方法背后的哲学。

代表性用户的兴趣偏好线性组合能与其他用户的无限接近。例如，基于代表性的矩阵因子分解（RBMF, Representative Based Matrix Factorization），其实是矩阵因子分解方法的扩展，其附加约束条件是m个项应该由k个项的线性组合表示，如下面的目标函数所示：

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \quad & ||\mathbf{R} - \mathbf{UV}||_F^2 + \alpha ||\mathbf{V}||_F^2 \\ \text{s.t.} \quad & \mathbf{U} \in \mathbb{R}^{n \times k} \text{ and } \mathbf{U} \subset_{n, k} \mathbf{R}, \\ & \mathbf{V} \in \mathbb{R}^{k \times m}, \end{aligned}$$

有了这个附加约束条件，就有了类似于标准MF方法的重建误差。当新用户进入平台时，要求新用户对这k个项进行评级，并用它来推断其他m-k项的评级。这种方法通过让用户对某些项目进行额外评级，从而改进对新用户的推荐。经过改进的RBMF只需要一部分新用户对这些项目评级，而不是所有的新用户。

优点

- 用少数代表性用户来表达新用户，更具可解释性；

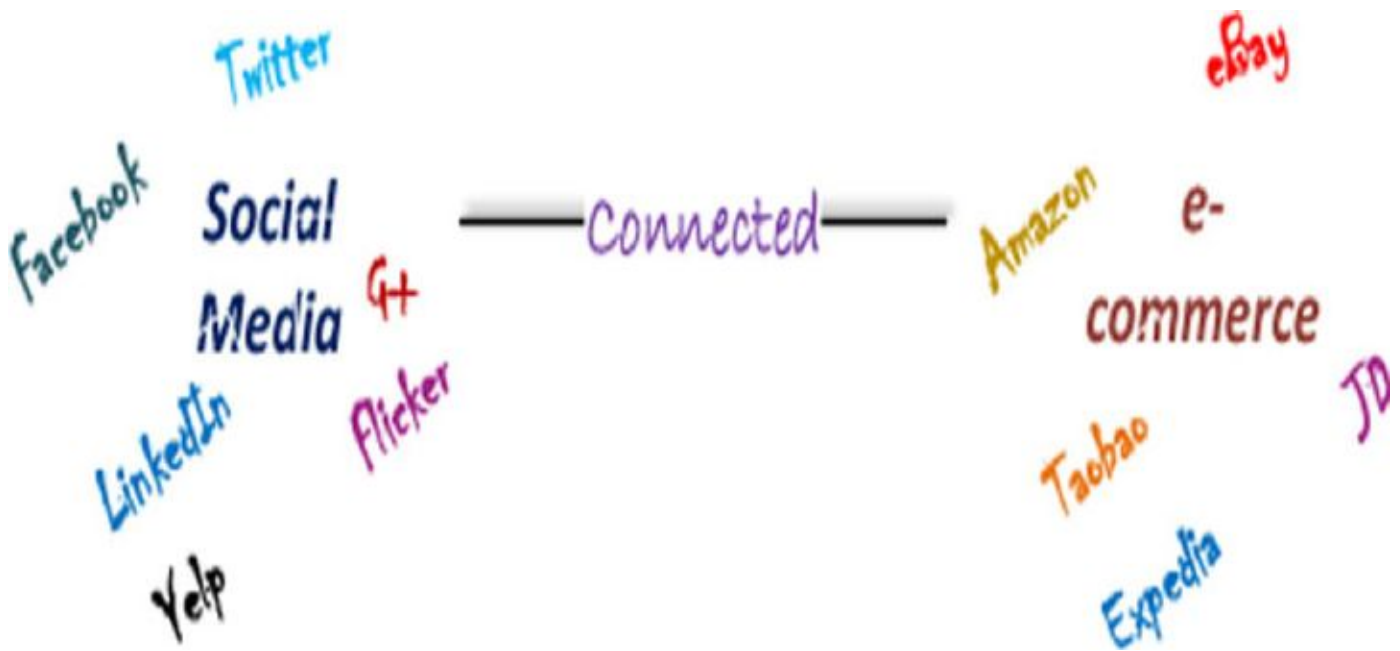
知乎

- 需要改进UI和前端逻辑，以要求用户对代表项进行评级。

基于内容

在推荐系统中还有许多未充分利用的辅助信息。

近年来，电子商务和社交网络之间的界限越来越模糊。许多电子商务网站支持社交登录机制，用户可以使用他们的社交网络账号登录，例如微信、微博、QQ帐户。用户还可以在微博上直接分享他们购买的新产品，其中包含指向产品网页的链接。



那如何在CF方法中处理这些内容呢？经典的CF方法以重建误差为目标，基本上是矩阵填充，从而很难利用这些辅助信息。因此，研究人员提出了将矩阵重建目标和基于内容的目标相结合的混合方法。这些方法并非特定的冷启动，感兴趣的读者可以进一步探讨。

优点

- 可以包含各种内容信息；
- 是MF的简单扩展。

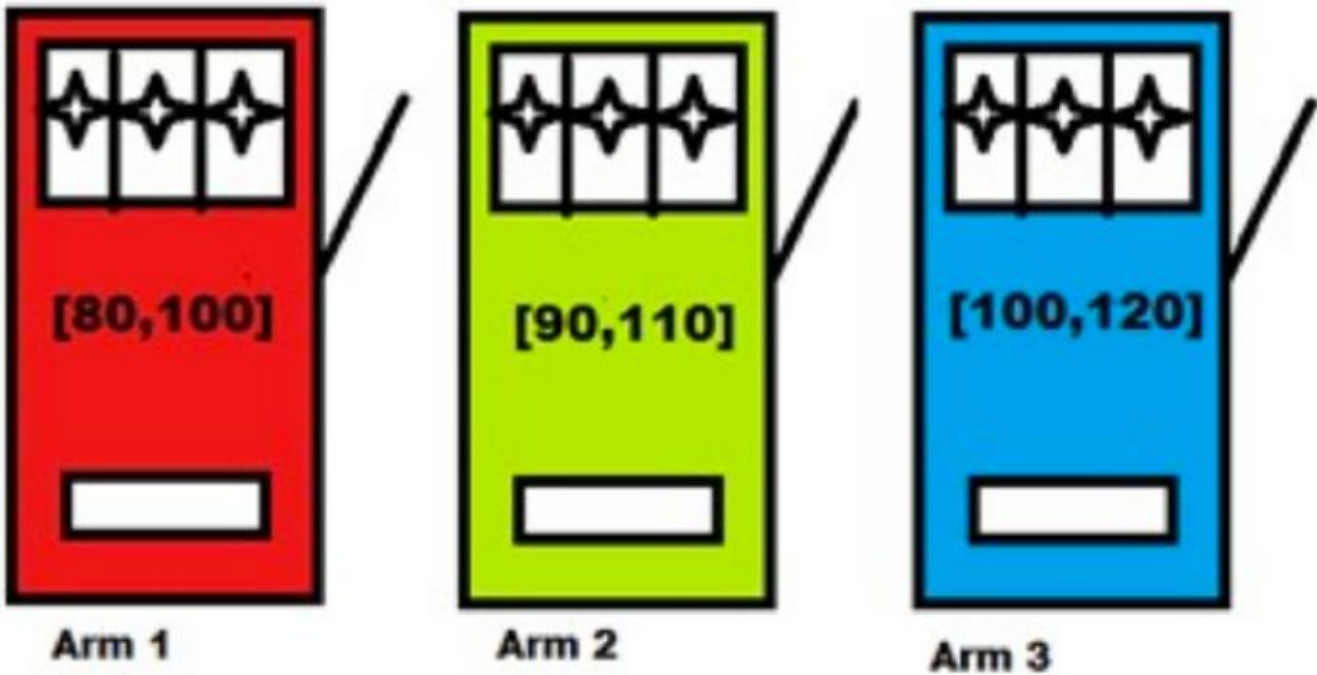
缺点

- 通过深层方法解决，具有不同的缺点，目标函数变得杂乱无章。
- 出现大量的功能工程，需要考虑是否在数据库中插入用户/项目功能。

知乎

冷启动问题可以重新定义并转换为老虎机问题。什么是老虎机（Bandit）问题？一个典型的例子如下：

一个赌徒，要去摇老虎机，走进赌场一看，一排老虎机，外表一模一样，但是每个老虎机吐钱的概率可不一样，他不知道每个老虎机吐钱的概率分布是什么，那么每次该选择哪个老虎机才可以做到收益最大化呢？



老虎机问题和冷启动问题又有什么关系呢？我们可以这样类比：以电子商务平台为例，每天有k个新物品进入平台，我们可以把这k个项看作是具有不同回报（收入、利润）的老虎机。但由于它们是新产品，因此无权访问购买它们的用户数量。推荐哪些项目就像选择要用哪台老虎机一样。

在老虎机问题上，我们会非常关心EE（exploration vs exploitation）问题。如果发现一个销量非常好的新商品，我们希望向用的展示的次数也更多（Exploitation）；但与此同时，我们还想展示其他没有展示过的东西，因为它们有可能比已经展示过的商品（Exploration）更受欢迎。

解决老虎机问题有很多方法：

•**Epsilon-Greedy算法**：这是最简单的一种，选一个（0,1）之间较小的数作为epsilon；每次以概率epsilon做一件事：所有臂中随机选一个；每次以概率1-epsilon 选择截止到当前，平均收益最大的那个臂。epsilon的值可以控制对Exploit和Explore的偏好程度。越接近0，越保守，只想花钱不想挣钱。

$$\bar{x}_j(t) + \sqrt{\frac{2 \ln t}{T_{j,t}}}$$

初始化：先对每一个臂都试一遍；

按照如下公式计算每个臂的分数，然后选择分数最大的臂作为选择：

观察选择结果，更新 t 和 T_{jt} 。其中加号前面是这个臂到目前的收益均值，后面的叫做bonus，本质上是均值的标准差， t 是目前的试验次数， T_{jt} 是这个臂被试次数。

这个公式反映一个特点：均值越大，标准差越小，被选中的概率会越来越大，同时哪些被选次数较少的臂也会得到试验机会。

•**Thompson sampling算法**：这种算法简单实用，因为它只要一行代码就可以实现。简单介绍一下它的原理，要点如下：

每个臂是否产生收益，其背后有一个概率分布，产生收益的概率为 p 。

我们不断地试验，去估计出一个置信度较高的“概率 p 的概率分布”就能近似解决这个问题了。

怎么能估计“概率 p 的概率分布”呢？答案是假设概率 p 的概率分布符合 $\text{beta}(\text{wins}, \text{lose})$ 分布，它有两个参数: wins , lose 。

每个臂都维护一个 beta 分布的参数。每次试验后，选中一个臂，摇一下，有收益则该臂的 wins 增加1，否则该臂的 lose 增加1。

每次选择臂的方式是：用每个臂现有的 beta 分布产生一个随机数 b ，选择所有臂产生的随机数中最大的那个臂去摇。

上下文老虎机问题

上下文老虎机问题是老虎机问题的一般化，更接近现实世界的情况。唯一的区别是，除了动作（例如，要显示的项目）和奖励（例如，销售）之外，还有上下文（例如项目描述）。因此，用上下文中的

知乎

优点

- 易于实现：与复杂的机器学习方法相比，简单的Bandit算法非常容易实现；
- 良好的理论保证：由于其简单性，已经对平均情况和最坏情况的奖励进行了大量的理论分析。

缺点

- 太过简单？

深度学习

这里把深度学习视为解决冷启动问题的一大类，但其实使用深度学习的方式非常多样化，以下只是其中的一些方法：

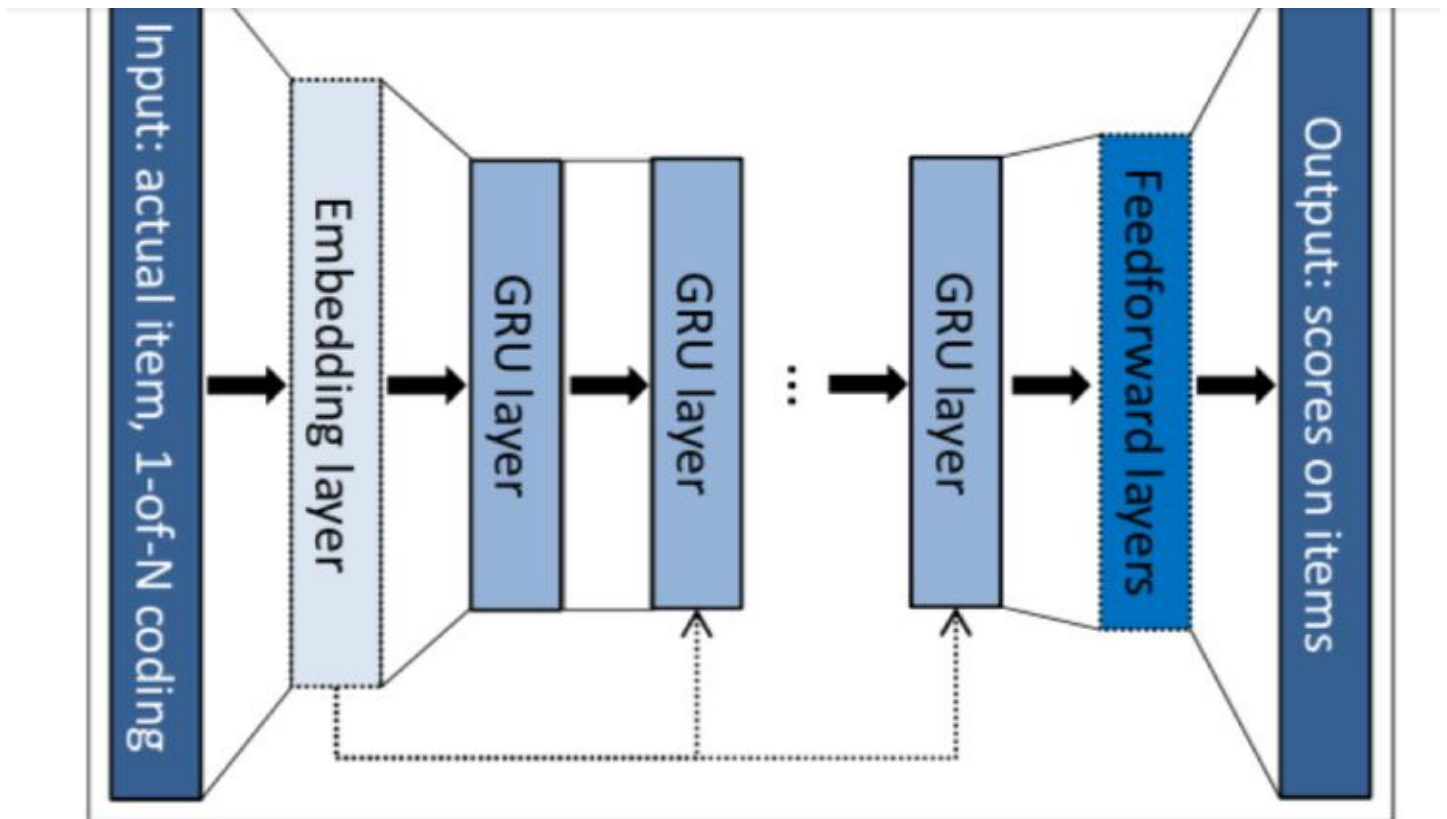
DropoutNet:

基本思路很简单，但很有效果。具体方法是，在训练基于深度学习的推荐系统（如多层的协同过滤）时，我们可以随机舍去某些项目和用户的评级，使其对新项目具有鲁棒性。与神经网络训练中的标准dropout相反，神经网络训练会舍去特征，而DropoutNet会舍去节点。这样做可以减少神经网络对某些评级的依赖，更普遍地适用于评级较少的项目/用户。这种方法的优势是可以与任何基于神经网络的推荐系统一起使用，同时还适用于用户和物品的冷启动。

基于会话的RNN:

这种方法试图把每个用户会话反馈到RNN中，具体来说，它训练了GRU（Gated Recurrent Unit）的变体，其中输入是会话的当前状态，输出是会话中下一个事件的项目。在只有少数用户会话可用的小型电子商务网站中，这种方法很有用。

知乎



优点

- 如果能够找到适合当前目的的特定域的模型，则有可能显著提高推荐性能。

缺点

- 实施和调整模型花费时间；
- 部署深层模型可能需要更多时间，具体情况取决于当前的堆栈；
- 具体效果无法保证。

原文链接：[Tackling the Cold Start Problem in Recommender Systems](#)

以上内容由第四范式先荐编译，仅用于学习、交流，版权归作者所有。