无中生有：论推荐算法中的Embedding思想

原创 石塔西 推荐道 11月30日

收录于话题
#推荐算法 10 #深度学习 8 #Embedding 2 #图神经网络 5

前言

前段时间面试了许多应界生同学，惊讶地发现很多同学只做深度学习，对于LR/GBDT这样的传统机器学习算法，既不掌握理论，也从未实践过。于是就想写一篇文章，梳理一下推荐算法由传统机器学习，发展到深度学习，再到未来的强化学习、图神经网络的技术发展脉络，因为**「只有了解过去，才能更好地把握当下与未来」**。

无奈这个题目太大，再加上近来分身乏术，实在无暇宏篇大论。于是今日小撰一文，聚焦于深度学习的核心思想Embedding（**「Embedding is all you need」**😃），管中窥豹，梳理一下推荐算法的前世（前深度学习时代）、今生（当下的深度学习时代）和将来（图神经网络处于燎原的前夕）。本文只讨论算法思想，即“道”的部分，至于如何实现具体算法，属于“技”的部分，请移步本人专栏里面的其他文章。

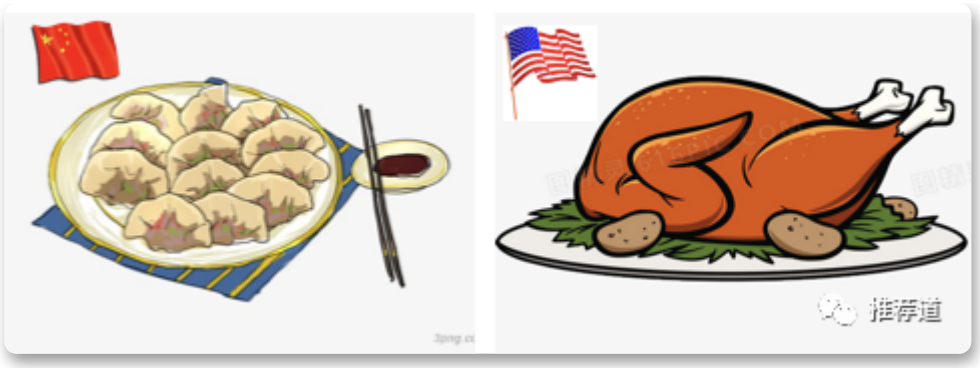
任何一门技术，要想获得互联网打工人的青睐，都必须能够实实在在解决我们面临的问题。那推荐算法面临的经典问题，无非两个，“**「记忆」**”与“**「扩展」**”。

推荐算法的传统机器学习时代：博闻强记

我们希望推荐系统记住什么？能够记住的肯定是那些**「常见、高频」**的模式。举个简单的例子：

- 到了春节，来了中国人，电商网站给他推饺子，大概率能够购买
- 到了感恩节，来了美国人，电商网站给他推火鸡，大概率也能购买

为什么？因为<春节，中国人，饺子>的模式、<感恩节、美国人、火鸡>的模式**「在训练样本中出现得太多太多了，推荐系统只需要记得住」**，下次遇到同样的场景，“照方扒抓药”，就能“药到病除”。



怎么记住？上“评分卡”

Logistic Regression就是一个非常擅于记忆的模型。说是模型，其实就是一个超大规模的“评分卡”。

变量名称	变量范围	得分
基准分	-	223
年龄	18 ≤ 年龄 < 25	-2
	25 ≤ 年龄 < 35	8
	35 ≤ 年龄 < 55	10
	55 ≤ 年龄	5
性别	男	4
	女	2
婚姻状况	已婚	8
	未婚	-2
学历	硕士，博士	10
	本科	8
	大专	5
	中专，技校，高中	1
	初中，小学	-2
月收入	月收入 < 3000	-8
	3000 ≤ 月收入 < 5000	0
	5000 ≤ 月收入 < 8000	5
	8000 ≤ 月收入 < 12000	13
	12000 ≤ 月收入	20

上图的评份卡，是金融风控领域用来评估申请人的信用分。推荐算法的LR，如果形象地画出来，与上面的评分卡类似，只不过卡里面的条目要多得多得多。

- 一个特征（中国、美国），或特征组合（<春节、中国人、饺子>）占据“推荐评分卡”中的一项。可想而知，一个工业级的推荐LR的评分卡里面，条目会有上亿项。
- 每项（i.e.，特征或特征组合）都对应一个分数
- 这个分数是由LR学习出来的，有正有负，代表对最终目标（比如成交，即label=1）的贡献。比如 SCORE(<春节，中国人，饺子>)=5 ，代表这种组合非常容易成交；反之 SCORE(<中国人、鲱鱼罐头>)=-100 ，代表这个组合极不容易成交
  - 简单理解，可以认为在正样本中出现越多的特征（组合）得分越高，反之在负样本中出现越多的特征（组合）得分越低
- 最终给一个<user，context，item>的打分是其命中的评分卡中所有条目的得分总和。比如当一个中国客户来了，预测他对一款“榴莲馅水饺”的购买欲望= SCORE(<春节、中国人、饺子>)+SCORE(<中国人，榴莲>)=5-3.5=1.5 ，即推荐系统猜他还是有可能会购买，但是欲望并不那么强烈。

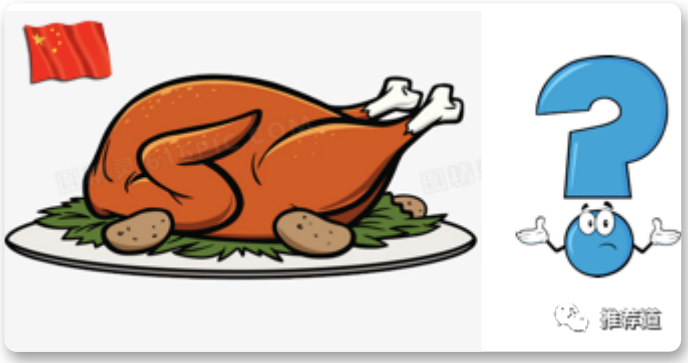
LR("评分卡")模型的特点

- LR的特点就是强于记忆，只要评分卡足够大（比如几千亿项），它能够记住历史上的发生过的所有模式（i.e.，特征及其组合）。
- 所有的模式，都依赖人工输入。
- LR本身并不能够发掘出新模式，它只负责评估各模式的重要性。（通过Cross Entropy Loss + SGD学习得到）
- LR不发掘新模式，反之它能够通过regularization，能够剔除一些罕见模式（比如<中国人，于谦在非洲吃的同款恩希玛>），即避免过拟合，又减少评分卡的规模

LR("评分卡")模型的缺陷

LR强于记忆，弱于扩展。还举刚才的例子

- 中国人来了推饺子，美国人来了推火鸡，都效果不错，毕竟LR记性好。
- 但是，当一个中国人来了，你的推荐系统会给他推荐一只火鸡吗？
- 假设是几前年，当时中国人对洋节接受度不高。如果你的推荐系统只有LR，只有记忆功能，答案是：**「不会」**。因为<中国人，火鸡>属于小众模式，在历史样本罕有出现，LR的L1正则直接将<中国人火鸡>打分置0，从而被从评分卡中剔除。



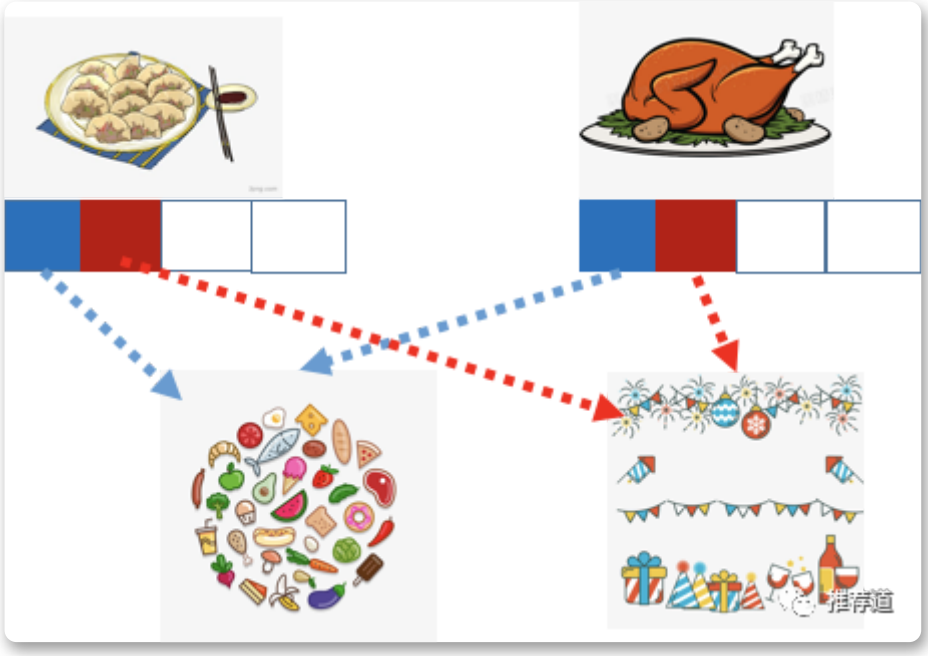
不要小看这个问题，它关乎到企业的生死，也就关系到你老板和你的腰包

- 记忆，记住的肯定是那些常见、高频、大众的模式，能够handle住80%用户的80%的日常需求，但是**「对小众用户的小众需求呢」**（某些中国人喜欢开洋荤的需求、于老师的超级粉丝希望和偶像体验相同美食的需求）？**「无能为力」**，因为缺乏历史样本的支持，换句话说，推荐的个性化太弱。
- 另一个问题是，大众的需求，你能记住，别家电商也能记住。所以你和你的同行，只能在“满足大众需求”的这一片红海里相互厮杀。套用如今最时髦的词，“**「内卷」**”。

推荐算法的刚需：扩展

综上所述，为了避开“大众推荐”这一片内卷严重的红海，而拥抱“**「个性化精准推荐」**”的**「蓝海」**，推荐算法不能只满足于记住“常见、高频”的模式（训练数据中频繁出现的），而必须能够自动挖掘出“**「低频、长尾」**”（训练数据中罕见的）模式。

如何扩展？看似神秘，其实就是将粗粒度的概念，拆解成一系列细粒度的特征，从而“看山非山、看水非水”。还举饺子、火鸡的例子



- 在之前讲记忆的时候，饺子、火鸡都是独立的概念，看似无什么相似性
- 但是，如果我们根据业务知识，将概念拆解，如上图所示。两个特征向量的第一位表示“是否是食物”，从这个角度来看，饺子、火鸡非常相似；两个特征的第二位是“是否和节日相关”，从这个角度来看，饺子、火鸡也非常相似。
- 喂入LR（评分卡)的除了粗粒度模式，<春节，中国人，饺子>和<感恩节，美国人，火鸡>，还有细粒度的模式，比如<节日，节日相关的食物>。这样一来，<春节，中国人，火鸡>这样的**「小众模式，也能够命中评分卡」**，并获得一个中等分数（因为<节日，节日相关的食物>在正负样本中都有出现，所以得分中等）。**「相比于原来被L1正则优化掉，小众模式也有了出头之日，获得了曝光的机会」**。

这样看来，只要我们喂入算法的，不是粗粒度的概念，而是细粒度的特征向量，即便是LR这样强记忆的算法，也能够具备扩展能力。

但是，上述方法依赖于人工拆解，也就是所谓的“特征工程”，有两方面的缺点：

- 工作量大，劳神费力
- 人的理解毕竟有局限性。比如饺子、火鸡，拆解到食物、和节日相关这个级别，就已经算是细粒度了吗？还能不能从其他角度拆解？

既然人工拆解有困难、受局限，即能不能**「让算法自动将概念拆解成特征向量」**？如果你能够想到这一步，恭喜你，你一只脚已经迈入了深度学习的大门。你已经悟到了“道”，剩下的只是“技”而已。

深度学习的核心套路：无中生有的Embedding

学习的过程，就是把书读薄的过程。我曾经提到过，林彪元帅用“剪贴法”来读书：在读书时，选择他认为“有用”的话剪贴起来。一本《共产党宣言》最后被他剪到最后只剩下“大工业、大机器”几个字。

区区不才，而欲效法先贤。到目前为止，我也曾经将两门技术总结成四字成语，并“自鸣得意”。第一个，我将Object-Oriented Programming总结为“求同存异”，即OOP的核心思想就是将不同的实现隐藏在相同的接口后面。另一个就是深度学习，我总结它为“**「无中生有」**”，也就是本文标题的来历。

所谓“无中生有”，

- 就是当你需要用到一个概念的特征 v（比如前面例子里的饺子、火鸡），或者一个函数 f（比如阿里Deep Interest Network中的“注意力”函数、CNN中的filter），但是却不知道如何定义它们，
- 没关系，先将 v 声明为特征向量，将 f 声明为一个小的神经网络，并随机初始化

- 然后让 v 和 f ，随着主目标（最终的分类或回归loss），一同被SGD所优化。
- 当主目标被成功优化之后，我们也就获得了有意义的 v 和 f 。

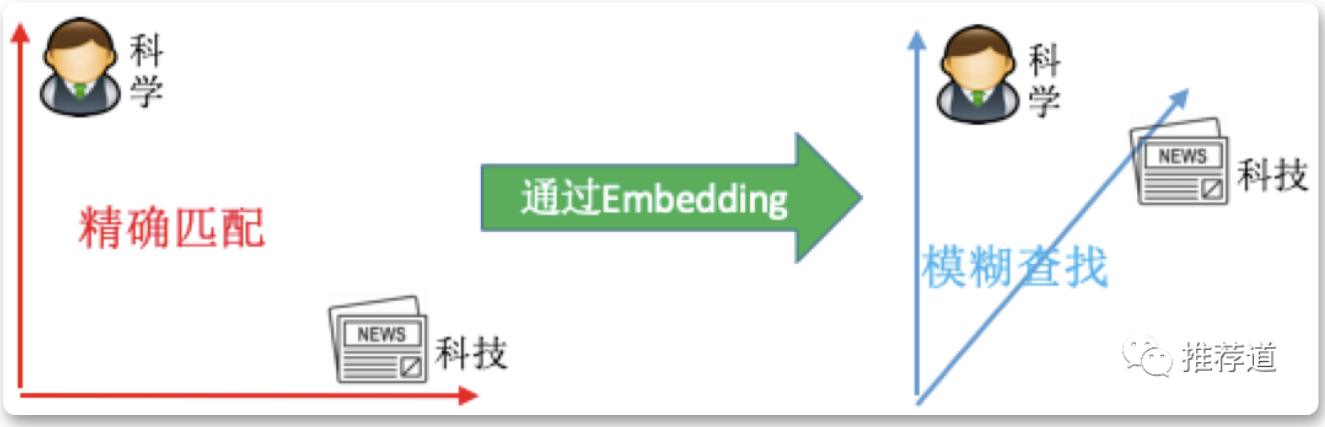
这种“无中生有”的套路，好似“上帝说，要有光，于是便有了光”的神迹。以讹传讹，后来就变成了初学者口中“深度学习不需要特征工程”，给了某些人“我只做深度学习，不做机器学习”的盲目自信。其实这种“**「将特征、函数转化为待优化变量」**”的思想，并不是深度学习发明的，早在**「用矩阵分解进行推荐」**的“古代”就已经存在了，只不过那时候，它不叫Embedding，而叫“**「隐向量」**”。

变“精确匹配”为“模糊查找”

深度学习对于推荐算法的贡献与提升，其核心就在于Embedding。如前文所述，Embedding是一门自动将概念拆解为特征向量的技术，目标是提升推荐算法的扩展能力，从而能够自动挖掘那些低频、长尾、小众的模式，拥抱“个性化推荐”的“蓝海”。

Embedding到底是如何提升“扩展”能力的？简单来说，Embedding将推荐算法从“**「精确匹配」**”转化为“**「模糊查找」**”，从而能够“**「举一反三」**”。

比如在使用倒排索引的召回中，是无法给一个喜欢“科学”的用户，推出一篇带“科技”标签的文章的（不考虑近义词扩展），因为“科学”与“科技”是两个完全独立的词。但是经过Embedding，我们发现“科学”与“科技”两个向量，并不是正交的，而是有很小的夹角。设想一个极其简化的场景，用户向量就用“科学”向量来表示，文章的向量只用其标签的向量来表示，那么用“科学”向量在所有标签向量里做Top-K近邻搜索，一篇带“科技”标签的文章就有机会呈现在用户眼前，从而破除之前“只能精确匹配‘科学’标签”带来的“**「信息茧房」**”



再回到原来饺子、火鸡的例子里，借助Embedding，算法能够自动学习到火鸡与饺子的相似性，从而给<中国人，火鸡>的小众组合打一个不低的分数，使火鸡得到了推荐给中国人的机会，从而能更好地给那些喜欢过洋节的中国人提供更好的个性化服务

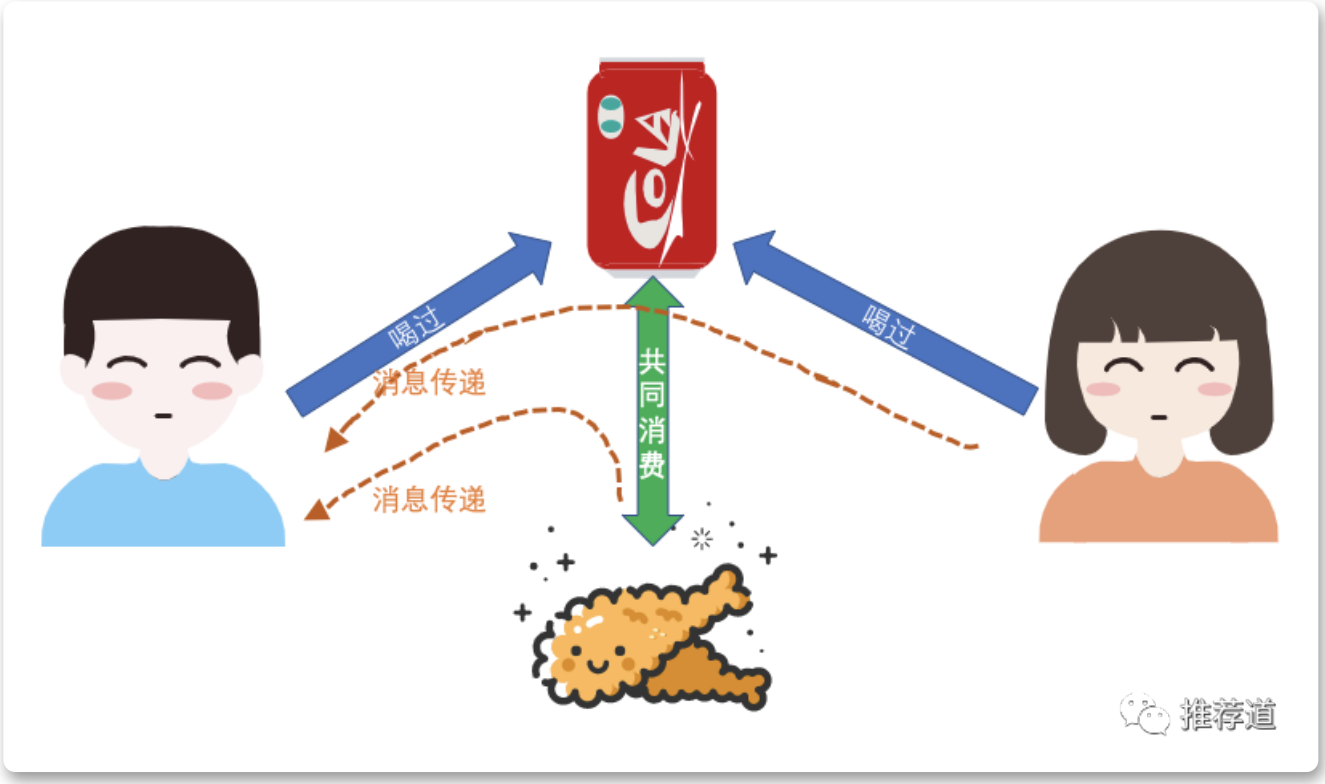
谁来接过Embedding的手中的旗帜？

在以Embedding为核心的深度学习之后，推荐算法的下一个技术方向在哪里？现如今，“图神经网络在推荐领域的应用”的paper层出不穷，在各大厂也已经有落地的成功案例，处于燎原的前夕。但是作为一个合格的炼丹师+调参侠，总要搞清楚GNN为什么火？它到底解决了什么当下技术无法解决的难题？

我在《也评Deep Interest Evolution Network》中曾经提到过，“**「高维、稀疏的categorical/id类特征都是推荐系统中的一等公民」**”。比如，用户购买过的商品、光顾过的店铺、搜索过的关键词、商品的分类与标签，都是这样的ID类特征。包括Embedding在内的很多推荐技术，都是为了更好地伺候好这些一等公民而提出的。

而到了“图计算”或“知识图谱”的阶段，ID类特征换了个名字，变成图上的节点或者知识图谱中的entity。换名字是小事，**「关键是这些ID不再是孤立的，而是彼此关联，从而带来了信息的传递」**。

- 之前，小明喝过“可口可乐”，只有“可口可乐”自己，（通过Embedding）为推荐算法刻画小明贡献信息。
- 如今，因为小红也喝过“可口可乐”，小红的信息也能传递给小明；
- 因为“可口可乐”与“炸鸡”经常一起消费，所以“炸鸡”的信息也能够传递到小明身上。



可以发现，**「如果说Embedding是在提升各ID类特征的内涵，那么GNN就是在扩展各ID类特征的外延」**。

所以，GNN瞄准的改进方向是：

- 之前，像用户访问过的店铺、商品所属分类这样的ID类信息，只是单纯地为刻画user和item贡献了自己本身的信息，但是**「它们背后的“社交”功能还未被开发和利用」**。
- 与当前用户逛同一家商店的其他用户的信息，对于刻画当前用户也非常有帮助。同理还有与当前用户喜欢同一品牌的其他用户的信息、与当前用户使用相同搜索词的其他用户的信息、……。正所谓“**「人以群分」**”，这种类似于**「User Collaborative Filtering」**的思想被实践证明是非常有效的。
- 与当前商品同属一个类别的其他商品的信息，对于刻画当前商品也非常有帮助。同理还有与当前商品属于一个品牌的其他商品的信息，与当前商品使用类似文字描述的其他商品的信息、……。正所谓“**「物以类聚」**”，这种类似于**「Item Collaborative Filtering」**的思想同样被实践证明是相当有效的。
- **「GNN通过图上的信息传递，充分开发、利用了ID类特征的社交功能」**，弥补了短板。GNN不仅能够利用当前user与item自身的信息，还融合了与其类似的user/item的信息，类似User CF或Item CF。可咨利用的信息大大丰富，有助于模型学到更复杂的模式，**「同时也缓解了对低活用户、冷门商品的“冷启动”问题」**。

对于GNN在推荐系统中的应用感兴趣的同学，可以参考我的另一篇文章《知识图谱上的双塔召回：阿里的IntentGC模型》。

简单总结一下，帮助各位调参侠+打工人，更好地掌握推荐算法的“昨天、今天与明天”：

- **「传统机器学习」**，对训练数据中出现的模式，只会**「死记硬背」**
- 以Embedding为核心的**「深度学习」**技术，**「扩展了」**训练时所见模式的**「内涵」**
- 以GNN为代表的**「图计算」**技术，**「扩展」**了训练时所见模式的**「外延」**