

Field Aware Factorization Machines (FFM)

[2016]

1. 回顾FM

如果说FM有什么缺点，那就是FM这个点积形式其实带来了一点限制，两个交叉的embedding会变得越来越相似。

就以FFM论文中的例子，我们有3个embedding需要相互交叉：出版商ESPN，商家Nike，性别男。按照FM的设计，如果ESPN和Nike经常一起出现贡献一个正样本，那它们两个的embedding v 是会变的像的，这是因为交互形式是点积。同理Nike和男同时出现，这两个embedding也应该长得像才对。但是这时候就有问题了，有可能ESPN不应该和男的embedding长得像呢？或者说，现在这三个embedding被捆绑了，互相之间会有拉扯，如果ESPN和男的embedding在实际中长得其实不像，那么Nike的embedding该往那边走呢？

2. FFM

首先，先来定义两个term：

- **fields**(域). 类似Gender, Genre, Region 的字段叫做"域"。
- **feature**(特征). 类似male,female, action,romance 叫做"特征"。

Fields	Gender	Genre	Region	Rating
↑	Male	Action	North America	5
	Female	Action	North America	3
↓	Male	Romance	Asia	1
	Male	Thriller	South America	4
	Male	Sci-Fi	Asia	4

不同域的特征之间，往往具有明显的差异性。而FM忽略了这种差异性：即使是同一个field中的特征，它们两者之间也是完全独立的；每个特征仅有一个隐向量，在对特征与其他特征进行交叉时，始终使用同一个隐向量。这种无差别式交叉方式，并没有考虑到不同特征之间的共性（同域）与差异性（异域）。例如，上图中第一行的二阶特征如果用FM的话表示为：

$$v_{male} \cdot v_{action} + v_{male} \cdot v_{northamerica} + v_{action} \cdot v_{northamerica}$$

FFM认为，一个特征和另一个特征的关系，不仅仅是这两个特征决定，还应该和这两个特征所在的域有关，因此，每个特征，应该针对其他特征的每一种域都学一个隐向量，也就是说，每个feature都要学习F个隐向量(F为field个数)。因此，上图中第一行的FFM二阶特征交互为：

$$v_{male, genre} \cdot v_{action, gender} + v_{male, region} \cdot v_{NAmerica, gender} + v_{action, region} \cdot v_{NAmerica, genre}$$

说白了就是每一个特征都准备多套的embedding，然后在一个合适的field里面，我就用这个域下面的embedding来做交叉。参数的数量多了很多($O(dnF)$), n为特征个数, d为embedding size, F为field size。embedding table大小即为 $O(dnF)$ 。), 但是同时自由度也大了很多，没有了上面所说拉扯的问题。

要注意FFM还有一个很大的不同是，在交叉的时候放到哪个域里面，这个操作是手动指定的，更进一步地，允许哪些特征来交叉，也是可以手动指定的。所以在实践中几乎没有人憨憨的复原FM那样任意两个特征之间都能交互，而是选择人认知中需要交叉的特征来交叉。

总结一下，FFM与FM有两点不同，第一，是人为手工挑选交叉的对象。第二，每一个field里面是一套新的embedding。

3. 复杂度分析

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle v_{i, f_i}, v_{j, f_j} \rangle x_i x_j$$

\downarrow 特征j 对应的 field 编号
 \downarrow 特征i 对应的 field 编号

对比 FM:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j$$

https://blog.csdn.net/qq_41328208

由于引入了Field，公式不能像FM那样进行改写，所以FFM模型进行推断时的时间复杂度为 $O(kn^2)$

将公式简单的展开：

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle v_{i,f_j}, v_{j,f_i} \rangle x_i x_j$$

$$= w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \left[\sum_{f=1}^k v_{i,f_j,f} \cdot v_{j,f_i,f} \right] \cdot x_i \cdot x_j$$

性能: $O(k \cdot n^2)$

https://blog.csdn.net/qq_41320028

由于有多少field我们就要有多少组embedding, 所以这个空间复杂度还是很高的, 所以原始的FFM并不常用。但是, FFM这种多套embedding的思想还是非常有启发性的!

FFM was restricted by the need of large memory and cannot easily be used in Internet companies.