FFM：将相同特性的特征归到同一Field的「领域意识」因子分解机，相比于FM，其优势在哪？

原创 去远方看太阳 随时学丫 3月24日

这是小娌的第 10 期分享

作者 | 小娌

来源 | 小娌（ID: a18581391726）

本文共 1705 字，预计阅读需 5 分钟

FFM 算法，全称是 Field-aware Factorization Machines，是 FM（Factorization Machines）的改进版。FFM 最初的概念来自 Yu-Chin Juan（阮毓钦，毕业于中国台湾大学，现在美国 Criteo 工作）与其比赛队员。通过引入 field 的概念，FFM 把相同性质的特征归于同一个 field。

在 FFM 中，每一维特征 x_i，针对其它特征的每一种 field f_j，都会学习一个隐向量 V_{i,f_j}。因此，隐向量不仅与特征相关，也与 field 相关。这也是 FFM 中 “Field-aware” 的由来。

FFM 相比于 FM，做了哪些优化？

回顾一下 FM

$$y\hat{} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j$$

样本 x 是 n 维向量，xi 是第 i 个维度上的值。vi 是 xi 对应的长度为 K 的隐向量，V 是模型参数，所以所有样本都使用同一个 V，即 $x_{1,1}$与 $x_{2,1}$都使用 v_1。

FFM 中的 Field

在 FFM 中，每一维特征 feature 都归属于一个特定的 field，field 和 feature 是一对多关系。

filed	**field1=年龄**	**field2=城市**			**field3=性别**	
feature	x1=年龄	x2=北京	x3=上海	x4=深圳	x5=男	x5=女
user1	23	1	0	0	1	0
user2	25	0	1	0	0	1

对于连续特征，一个特征就对应一个 Field，或者对连续特征离散化，一个分箱成为一个特征，比如：

filed	**field1=年龄**			
feature	<20	20-30	30-40	>40
user1	0	1	0	0
user2	0	1	0	0

对于离散特征，采用 One-Hot 编码，同一属性的归到一个 Field，不论是连续特征还是离散特征，它们都有一个共同点：同一个 field 下只有一个 feature 的值不为 0，其它 feature 的值都为 0。

FFM 模型认为 v_i 不仅跟 x_i 有关系，还跟与 x_i 相乘的 x_j 所属的 field 有关系，即 v_j 成了一个二维向量 $v_{F.K}$，F 是 field 的总个数，FFM 只保留了 FM 式子中的二次项。

$$\hat{y} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} v_{i,f_j}, v_{j,f_i} x_i x_j$$

以上文的表格数据为例，计算 user1 的 $\overline{y}$。

$$\hat{y} = v_{1,f_2} . v_{2,f_1} x_1 x_2 + v_{1,f_3} . v_{3,f_1} x_1 x_3 + v_{1,f_4} . v_{4,f_1} x_1 x_4 + ...$$

由于 x_2, x_3, x_4 属于同一个 field，所以 f_2, f_3, f_4 可以用同一个变量来代替，比如就用 f_2。

$$\hat{y} = v_{1,f_2} \cdot v_{2,f_1} x_1 x_2 + v_{1,f_2} \cdot v_{3,f_1} x_1 x_3 + v_{1,f_2} \cdot v_{4,f_1} x_1 x_4 + \ldots$$

我们来算一下 $\hat{y}$ 对 v_{1,f_2} 的偏导。

$$\frac{\partial \hat{y}}{\partial v_{1,f2}} = v_{2,f1} x_1 x_2 + v_{3,f1} x_1 x_3 + v_{4,f1} x_1 x_4$$

等式两边都是长度为 K 的向量。

注意 $x2, x3, x4$ 是同一个属性的 one-hot 表示，即 $x2, x3, x4$ 中只有一个为 1，其他都为 0。在本例中 $x3 = x4 = 0, x2 = 1$，所以

$$\frac{\partial \hat{y}}{\partial v_{1,f2}} = v_{2,f1} x_1 x_2$$

推广到一半情况：

$$\frac{\partial \hat{y}}{\partial v_{i,fj}} = v_{j,fi} x_i x_j$$

xj 属于 Field fj，且同一个 Field 里面的其他 xm 都等于 0。实际项目中 x 是非常高维的稀疏向量，求导时只关注那些非 0 项即可。

你一定有个疑问：v 是模型参数，为了求 v 我们采用梯度下降法时需要计算损失函数对 v 的导数，为什么这里要计算 $\hat{y}$ 对 v 的导数？

在实际预测点几率的项目中，我们是不会直接使用公式 $\hat{y} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} v_{i,f_j}, v_{j,f_i} x_i x_j$ 的，通常会再套一层 sigmoid 函数，公式中的 $\hat{y}$ 用 z 来取代。

$$z = \phi(v, x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} v_{i,fj} \cdot v_{j,fi} xixj$$

由 $\frac{\partial \hat{y}}{\partial v_{1,fj}} = v_{2,fj} x_i x_j$ 得

$$\frac{\partial z}{\partial v_{i,fj}} = v_{j,fi} x_i x_j$$

用 a 表示对点击率的预测值

$$a = \sigma(z) = \frac{1}{1+e^(-z)} = \frac{1}{1+e^(-\phi(v,x))}$$

令 y=0 表示负样本，y=1 表示正样本，C 表示交叉熵损失函数。根据神经网络调优公式可得：

$$\frac{\partial C}{\partial z} = a - y = \begin{cases} -\frac{1}{1+e^{z}} & if\ y\text{是正样本} \\ \frac{1}{1+e^{-z}} & if\ y\text{是负样本} \end{cases}$$

$$\frac{\partial C}{\partial v_{i,fj}} = \frac{\partial C}{\partial z}\frac{\partial z}{\partial v_{i,fj}}$$

看完了本博客再去看论文《Field-aware Factorization Machines for CTR Prediction》中的公式推导应该就比较容易了吧，在该论文中他是以y=1y=1代表正样本，y=−1 代表负样本，所以才有了 3.1 节中的$k = \frac{\partial C}{\partial z} = \frac{-y}{1+e^(yz)}$

有了上面的推导，我们再来看 FFM 的公式。

设样本一共有 n 个特征，f 个 field，那么，FFM 的二次项有 nf 个隐向量。而在 FM 模型中，每一维特征的隐向量只有一个。FM 可以看做 FFM 的特例，是把所有特征都归属到一个 Field 时的 FFM 模型。根据 FFM 的 Field 敏感特性，可以导出其模型方程。

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

其中，f_j 是第 j 个特征所属的字段，如果隐向量的长度为 k，那么 FFM 的二次参数有 nfk 个，远远多于 FM 模型的 nk 个，此外，由于隐向量与 Field 相关，FFM 二次项并不能够化简，时间复杂度是 $O(kn^2)$。

需要注意的是由于 FFM 中的 laten vector 只需要学习特定的 Field，所以通常：KFFM << KFM。

FFM 由于引入了 Field，使得每两组特征交叉的隐向量都是独立的，可以取得更好的组合效果， FM 可以看做只有一个 Field 的 FFM。

FFM的论文地址：https://www.csie.ntu.edu.tw/~cjlin/papers/ffm.pdf

END