Item2Vec++

模型结构

在学习 SkipGram 模型和 CBOW 模型时，通过训练得到词向量，语义会在低维表征空间有所体现，比如$V(King)-V(Queen\approx V(Man)-V(Woman)$，其中$V(\cdot)$表示一个词在表征空间的向量。这就说明，通过对词序的训练，可以在表征空间得到词与词之间表征的潜在联系。类比在推荐系统中，如果将一个用户的评分记录看作是一段文本内容，每个物品看作是组成文本的词，那么通过训练物品的共同出现的概率模型，应该也可以得到物品在低维表征空间的物品向量，而这份物品向量也会体现出物品间所具有的潜在联系。

Barkan 等人在[4.1.1]中提出了一种训练物品向量的方法，对推荐系统中的物品做了聚类分析，将该方法命名为 Item2Vec。受到 Word2Vec 模型和[4.1.1]的启发，本文作者提出了一种 Item2Vec++模型，用于将评分矩阵中的高维稀疏物品向量映射到一个低维稠密的表征空间，通过训练物品共现概率，来获得一份物品在低维表征空间中的“物品向量”表示，即 Item to Vector（Item2Vec）。

对 Item2Vec++模型，我们做以下定义：在解决推荐问题时，我们会得到一个包含m个用户对n个物品反馈的评分矩阵，我们记为$r\in R^{m\times n}$，用户i对物品j的评分为r_{ij}，分数为$\{1,...,K\}$的K个正整数，用户未评分的用0表示，我们可以得到一个用户的评分向量$r_{i,:}$，同时我们可以对物品进行 one-hot 编码，物品j的 one-hot 编码为v_j，用户评分过的物品记为1，未评分过的物品记为0，用户的评分 one-hot 编码向量为$o_{i,:}$，所有用户的评分 one-hot 编码向量组成的矩阵记为$z\in R^{m\times n}$。比如，我们有以下评分矩阵如表 1 所示：

表 1 评分矩阵表

	物品 1	物品 2	物品 3	物品 4	物品 5	物品 6	物品 7	…
用户 1	0	5	4	2	0	0	3	…
用户 2	1	2	0	0	4	2	1	…
用户 3	2	2	2	0	0	0	0	…
用户 4	0	0	0	0	0	0	5	…
…	…	…	…	…	…	…	…	…

其中，我们以用户 1 和前 7 种物品为例：用户 1 的评分向量为$r_{1,:}=[0,5,4,2,0,0,3]$，用户 1 的评分 one-hot 编码向量为$o_{1,:}=[0,1,1,1,0,0,1]$，物品 1 的 one-hot 编码为$v_1=[1,0,0,0,0,0,0]$。我们根据评分大小并去除未评分物品，对评分向量进行排序，得到用户的物品评分行为序列，记为s，用户 1 的物品评分行为序列为$s_1=[2,3,4,5]$，也可以写作：

$$s_1^*=[item_4,item_7,item_3,item_2]$$

定义“窗”为与一个物品相关联物品的个数，记为w，在一段有序物品评分序列中

$s=\{item_1, item_2, ..., item_j, ..., item_k\}$，中心物品为$item_j$，那么在这段有序序列中与中心物品$item_j$有关的物品的集合为$\{item_{j-l}, item_{j+l} \mid l=1,..., w\}$，当取窗$w=1$时，与中心物品$item_j$相关物品的集合为$\{item_{j-1}, item_{j+1}\}$，当取窗$w=2$时，与中心物品$item_j$相关物品集合为$\{item_{j-2}, item_{j-1}, item_{j+1}, item_{j+2}\}$。我们通过用户 1 的物品评分行为序列$s_1$，进行窗$w=1$的相关物品提取，依次以每个物品为中心物品，提取周围跟它有关的物品，得到用户 1 用于训练的相关物品对的集合$p_{j=1}^{w=1}$，用评分表示物品则该集合可以写为$p_1^1=\{(2,3),(3,2),(3,4),(4,3),(4,5)\}$，同时$p_1^1$也可以写作如下形式：

$$p_1^{1*}=\left\{\begin{array}{lll}(& item_4, & item_7),\\ (& item_7, & item_4),\\ (& item_7, & item_3),\\ (& item_3, & item_7),\\ (& item_3, & item_2),\\ (& item_2, & item_3)\end{array}\right\}$$

将该物品的评分用物品的 one-hot 编码表示，则得到我们可以得到用户i的物品评分行为序列的关联物品的集合，记为c_i，用户 1 的物品评分行为序列的关联物品集合为：

$$c_1=\left\{\begin{array}{lll}(& [0,0,0,1,0,0,0], & [0,0,0,0,0,0,1]),\\ (& [0,0,0,0,0,0,1], & [0,0,0,1,0,0,0]),\\ (& [0,0,0,0,0,0,1], & [0,0,1,0,0,0,0]),\\ (& [0,0,1,0,0,0,0], & [0,0,0,0,0,0,1]),\\ (& [0,1,0,0,0,0,0], & [0,0,1,0,0,0,0]),\\ (& [0,0,1,0,0,0,0], & [0,1,0,0,0,0,0])\end{array}\right\}$$

如果将用户i的物品评分行为序列$s_i=\{item_1, item_2, ..., item_k\}$视为文本，其中每个物品视为词，则根据语言模型，推荐系统问题可以转化为求解概率：

$$P(item_{rec} \mid item_1, item_2, ..., item_k)$$

等同于求解

$$P(item_1, item_2, ..., item_k, item_{rec})$$

其中$item_{rec}$为我们推荐的物品。我们可以根据用户高分物品计算其它高分物品，算法过程如下：

表 1 基于 Item2Vec++模型的推荐算法

基于 Item2Vec++模型的推荐算法：
Step 1. 根据稀疏性选择 n_E 个频繁物品进行物品向量的训练；
Step 2. 根据评分矩阵 r 写出用户的评分 one-hot 编码向量组成的矩阵 z；
Step 3. 初始化 SkipGram 模型或者 CBOW 模型的参数；
Step 4. 随机选择 q 个用户生成训练样本，记为集合 Q，得到集合 Q 中用户 i 的物品评分行为序列的关联物品的集合 c_i，共同组成一组训练样本 C_q，$C_q = \bigcap_{i \in Q} c_i$；
Step 5. 将训练样本带入模型进行训练；
Step 6. 重复 Step4 和 Step5 共 iter_n 次；
Step 7. 将用户的评分较高的物品作为输入带入模型，计算输出概率的最大 n_{rec} 个作为推荐物品。

同时，根据我们得到用户 i 的评分行为序列的关联物品集合 c_i，我们可以带入 SkipGram 模型或者 CBOW 模型，得到一份嵌入向量（Embedding Vector）作为各物品在低维表征空间中的向量表示，通过计算余弦相似度我们可以得到物品之间的相似性，再根据协同过滤算法就可以对用户进行物品推荐。具体算法步骤如下表所示：

表 2 基于 Item2Vec++模型的协同过滤算法

基于 Item2Vec++模型的协同过滤算法：
Step 1. 根据稀疏性选择 n_E 个频繁物品进行物品向量的训练；
Step 2. 根据评分矩阵 r 写出用户的评分 one-hot 编码向量组成的矩阵 z；
Step 3. 初始化 SkipGram 模型或者 CBOW 模型的参数；
Step 4. 随机选择 q 个用户生成训练样本，记为集合 Q，得到集合 Q 中用户 i 的物品评分行为序列的关联物品的集合 c_i，共同组成一组训练样本 C_q，$C_q = \bigcap_{i \in Q} c_i$；
Step 5. 将训练样本带入模型进行训练；
Step 6. 重复 Step4 和 Step5 共 iter_n 次；
Step 7. 得到模型的嵌入矩阵 $W_{embedding} \in R^{K \times n}$，其中 K 为低维表征空间维度，计算每个物品的嵌入向量 $e_i \in R^{K \times 1}, i = 1, ..., n_E$；

Step 8. 计算 n_E 个物品在低维表征空间的相似性，得到相似矩阵 $M_{sim} \in R^{n_E \times n_E}$；

Step 9. 进行基于物品的协同过滤推荐。

与 Item2Vec 模型不同的是，Item2Vec++模型不仅考虑了物品出现的共现概率，同时考虑了评分的影响。同时 Item2Vec++模型既可以套用在 SkipGram 模型上，也可以套用在 CBOW 模型中。

频繁物品选择

在[4.1.1]中 Barkan 给出了一种频繁物品的选择方式：

$$p(discard \mid w) = 1 - \sqrt{\frac{\rho}{f(w)}}$$

其中 $p(discard \mid w)$ 为忽略词 w 的概率，$f(w)$ 为词 w 的频率，ρ 为预定的阈值。本文中直接采用计数的方式选择频繁物品的个数。同时，频繁物品个数不宜过少，当总物品数为 $10^5 \sim 10^6$ 时，频繁物品应选择 $10^3 \sim 10^4$，否则会失去推荐物品的广度，降低推荐系统的推荐效果。

窗大小的选择

窗大小的应该根据评分矩阵中数据的密度进行选择，我们计算的频繁物品数量越少，则相关联的物品选择应越多，本文中窗的大小计算公式为：

$$w = \log(\frac{n}{n_E})$$

其中 $\log(\cdot)$ 为以 10 为底数的对数，n 为物品数量，n_E 为频繁物品数量。

负采样训练

由于在训练中，正负例样本极度不平衡，所以在采用负采样的方法进行训练，即从负例样本中采出一定量的样本作为负例进行迭代训练。在[3.2.3]和[4.1.1]中都采用 Unigram 分布的 $3/4$ 次方作为随机生成的数量。本文中随机选取与正例样本相同数量的负例样本用于训练。

参考

[4.1.1] Barkan O, Koenigstein N. Item2vec: neural item embedding for collaborative filtering[C]//Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on. IEEE, 2016: 1-6.