论文｜Item2vec中值得品味的8个经典tricks

原创 Thinkgamer 搜索与推荐Wiki 2020-12-08

收录于话题
#论文笔记

18个

▼ 相关推荐 ▼

1、论文｜语言模型：从N-Gram到NNLM、RNNLM
2、论文｜Word2vec：2个模型、2个优化及实战使用
3、美团点评｜深度学习在推荐中的实践

本主题文章将会分为三部分介绍，每部分的主题为：

- word2vec的前奏-统计语言模型
- word2vec详解-风华不减
- 其他xxx2vec论文和应用介绍

后续会更新Embedding相关的文章，可能会单独成系列，也可能会放到《特征工程-Embedding系列中》，欢迎持续关注「**搜索与推荐Wiki**」

Item2vec：论文《Item2Vec：Neural Item Embedding for Collaborative Filtering》

来自于微软2016年发表在RecSys上的，因为word2vec和item2vec是在做推荐系统过程中比较常用的两个算法，所以该部分先介绍item2vec，然后再展开其他xxx2vec。

Item2vec其本质就是Word2vec中的skip-gram+Negative sampling（简称为SGNS），关于什么是SGNS可以参考之前介绍的word2vec篇幅的内容。

下面陈列一些从论文中可以学习到的经典的tricks

1、为什么选择的是SGNS而不是其它的组合

Item2vec为什么采用Skp-Gram + Negative Sampling这种组合呢？因为效果好，而在很多文章中也提到了SGNS这种组合下的实际业务提升要好一些（但并不能一刀切，只是说大多数业务场景下SGNS的效果好，但还是要视具体的情况而定）

2、实验场景的选择

对于这种item相似的算法，如何选择合适的实验场景呢？Item2vec论文中提到的是使用Windows10 App Store的「看了又看」推荐场景，即某个App的相似App推荐场景，这种场景下，对Item相似类算法进行是很合适的。

但是比如把item sim items加入到典型的「recall -> rank」场景中，其实达到的效果并没有那么好，但不能说不合适，这取决于截断的数目，即每个item取多少相似的item。因为在召回中并不会区分 item之间的顺序，比如top100，把100个item全部加到召回池中，并不会区分这100个item之间的顺序，这就会在一定程度上丢失掉这种相似的信息，极端情况下，假设我们的item总数为1000个，而召回时将item的相似item1000全部加入到召回池中，这种极端情况下就失去了个性化的意义。

因此选择一个合适的业务实验场景去评估我们的算法是极其重要的，否则得出的结论也没有什么说服力！

3、负采样不代表是均匀的随机负采样

均匀的随机负采样就代表采样时对所有的负样本采样的概率是一样的，但其实这是不符合实际的数据分布概率的。

因此论文中也使用到了一种非均匀的随机负采样技术，其表达式为：

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

其中：

- $f(w)$ 表示的是$item\ w$的频次
- ρ 表示是人为设定的参数，是一个经验值（论文中针对App Store数据集的设定值是10^{-3}，音乐数据集设定值是10^{-5}）

4、Item2vec基于SGNS的改进点

其改进点为修改window size为set的长度，即从原来的定长变成了变长，其它的保持不变。

Item2vec丢失了集合中的时间和空间信息。

5、Item2vec的等效表达方法

论文中特意指出采用**保持word2vec的SGNS不变，仅将set集合中的item进行随机的排序**可以达到和Item2vec同样的效果，相关的实验也对此进行了佐证。

因此在很大程度上方面了 Item2vec的等效表达，即采样Word2vec的算法代码同意可以得到Item2vec中的Item的embedding表示。

6、基于Item Embedding计算相似性的实验思路

a）聚类数据可视化

依据embedding进行item聚类，并在二维空间中进行数据展示，观察效果。比如论文中使用的t-SNE进行可视化，如下图所示，(a) 为基于Item2vec产出的item embedding进行的聚类可视化，(b)为基于SVD产出的item vector进行的聚类可视化，肉眼可见(a)要比(b)的效果好很多。

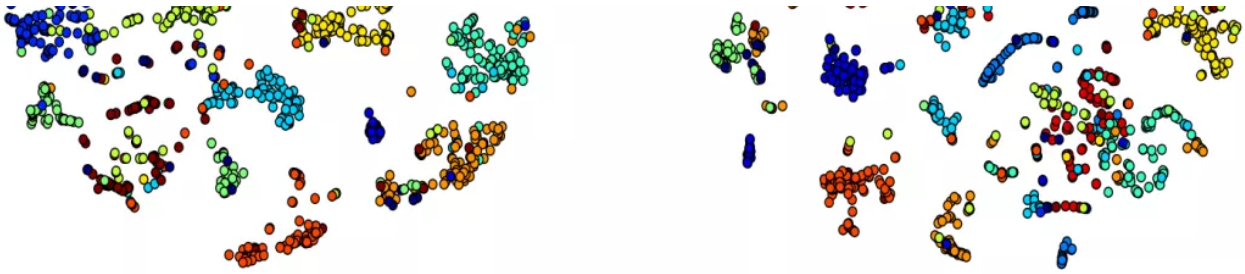

Figure 2: t-SNE embedding for the item vectors produced by Item2Vec (a) and SVD (b). The items are colored according to a web retrieved genre metadata.

Item2vec 聚类的可视化

b）top N item的类型一致性检验

首先基于KNN选取每个item的相似items，top N个，然后根据类型一致性检验统计top N中的一致性占比，继而进行对比分析。

在进行类型一致性检验时，这里选择了一些比较受欢迎的item，论文中关于「受欢迎」和「不受欢迎」的item定义方法为：如果一个item被用户交互次数少于15，则被认为是不受欢迎的。

实验结果如下图所示：

Top (q) popular artists	SVD accuracy	Item2Vec accuracy
2.5K	85%	86.4%
5K	83.4%	84.2%
10K	80.2%	82%
15K	76.8%	79.5%
20K	73.8%	77.9%
10K unpopular (see text)	58.4%	68%

7、聚类可视化的分析和带给我的思考（精华）

在上边提到了使用聚类可视化进行item2vec的效果表达，论文中指出上图(a)中那些同色的簇类中被其它颜色簇类污染的item中，有很多是标签打错或者类型交叉引起的。

那么这里是否可以通过可视化和分析对作品的标签或者其它属性进行反馈和修正呢？比如说上标签打错的item反馈给运营，让运营进行评估和修正，然后继续使用类似的方法进行评估，再反馈修正，以此达到一个良性的循环，从而解决运营打标签，人工再校验的复杂流程和容易出错的问题。

欢迎留言探讨！

8、实验参数的学习

论文中提到了了Item2vec进行实验时的参数值，这也为我们在实际业务中进行相关尝试，提供了初始的、经验上的模型参数值。

- 迭代次数：20
- 负采样数：1:15
- embedding维度：App数据集为40，Music数据集为100
- 人工设定值ρ：App数据集为10^{-3}，Music数据集为10^{-5}

OK，以上就是从Item2vec论文中学习到的知识，希望对你有帮助！点击「**阅读原文**」触达更多精彩内容！