个性化召回算法实践(五)——item2vec

item2vec将用户的行为序列转化成item组成的句子，模仿word2vec训练word embedding将item embedding。基本思想是把原来高维稀疏的表示方式(one_hot)映射到低维稠密的向量空间中，这样我们就可以用这个低维向量来表示该项目(电影)，进而通过计算两个低维向量之间的相似度来衡量两个项目之间的相似性。
embedding就是用一个低维的向量表示一个物体，可以是一个词，或是一个商品，或是一个电影等等。这个embedding向量的性质是能使距离相近的向量对应的物体有相近的含义
类似于Word2vec，item2vec有两种方式：CBOW和skip-gram模型。
CBOW使用的是词袋模型，模型的训练输入是某一个特征词的上下文相关的词对应的词向量，而输出就是这特定的一个词的词向量。
Skip-Gram模型和CBOW的思路是反着来的，即输入是特定的一个词的词向量，而输出是特定词对应的上下文词向量。

主流程：

- 从log中抽取用户行为序列
- 将行为序列当成预料训练word2Vec得到item embedding
- 得到item sim关系用于推荐

在代码中，我们直接用gensim库实现。在gensim中，word2vec 相关的API都在包gensim.models.word2vec中。和算法有关的参数都在类gensim.models.word2vec.Word2Vec中。算法需要注意的参数有：

1. sentences: 我们要分析的语料，可以是一个列表，或者从文件中遍历读出。
2. size: 词向量的维度，默认值是100。这个维度的取值一般与我们的语料的大小相关，如果是不大的语料，比如小于100M的文本语料，则使用默认值一般就可以了。如果是超大的语料，建议增大维度。
3. window：即词向量上下文最大距离，window越大，则和某一词较远的词也会产生上下文关系。默认值为5。在实际使用中，可以根据实际的需求来动态调整这个window的大小。如果是小语料则这个值可以设的更小。对于一般的语料这个值推荐在[5,10]之间。
4. sg: 即我们的word2vec两个模型的选择了。如果是0， 则是CBOW模型，是1则是Skip-Gram模型，默认是0即CBOW模型。
5. hs: 即我们的word2vec两个解法的选择了，如果是0， 则是Negative Sampling，是1的话并且负采样个数negative大于0， 则是Hierarchical Softmax。默认是0即Negative Sampling。
6. negative:即使用Negative Sampling时负采样的个数，默认是5。推荐在[3,10]之间。这个参数在我们的算法原理篇中标记为neg。
7. cbow_mean: 仅用于CBOW在做投影的时候，为0，则算法中的x_w为上下文的词向量之和，为1则为上下文的词向量的平均值。在我们的原理篇中，是按照词向量的平均值来描述的。默认值也是1,不推荐修改默认值。
8. min_count:需要计算词向量的最小词频。这个值可以去掉一些很生僻的低频词，默认是5。如果是小语料，可以调低这个值。
9. iter: 随机梯度下降法中迭代的最大次数，默认是5。对于大语料，可以增大这个值。
10. alpha: 在随机梯度下降法中迭代的初始步长。算法原理篇中标记为ηη，默认是0.025。
11. min_alpha: 由于算法支持在迭代的过程中逐渐减小步长，min_alpha给出了最小的迭代步长值。随机梯度下降中每轮的迭代步长可以由iter，alpha， min_alpha一起得出。对于大语料，需要对alpha, min_alpha,iter一起调参，来选择合适的三个值。

训练完模型后，常见的用法如下：

```
#找出某一个词向量最相近的词集合
model.wv.similar_by_word('沙瑞金'.decode('utf-8'), topn =100)
#看两个词向量的相近程度
model.wv.similarity('沙瑞金'.decode('utf-8'), '高育良'.decode('utf-8'))
#找出不同类的词
model.wv.doesnt_match(u"沙瑞金 高育良 李达康 刘庆祝".split())
```

全部代码如下所示：

```
#-*-coding:utf-8-*-
"""
author:jamest
date:20190405
CBOW function
"""
import pandas as pd
from gensim.models import Word2Vec
import multiprocessing
import os

class CBOW:
    def __init__(self,input_file):
        self.model = self.get_train_data(input_file)

    def get_train_data(self,input_file,L=100):
        if not os.path.exists(input_file):
            return
```

```
        score_thr = 4.0
        ratingsDF = pd.read_csv(input_file, index_col=None, sep='::', header=None,
                                names=['user_id', 'movie_id', 'rating', 'timestamp'])
        ratingsDF = ratingsDF[ratingsDF['rating']>score_thr]
        ratingsDF['movie_id'] = ratingsDF['movie_id'].apply(str)
        movie_list = ratingsDF.groupby('user_id')['movie_id'].apply(list).values
        print('training...')
        model = Word2Vec(movie_list, size=L, window=5, sg=0, hs=0, min_count=1, workers=multiprocessing.cpu_count())
        return model

    def recommend(self,movieID,K):
        """
         Args:
             movieID:the movieID to find similar
             K:recom item num
         Returns:
             a dic,key:itemid ,value:sim score
         """
        movieID = str(movieID)
        rank = self.model.most_similar(movieID,topn=K)
        return rank

if __name__ == '__main__':
    moviesPath = '../data/ml-1m/movies.dat'
    ratingsPath = '../data/ml-1m/ratings.dat'
    usersPath = '../data/ml-1m/users.dat'

    rank = CBOW(ratingsPath).recommend(movieID=1,K=30)
    print('CBOW result',rank)
```

参考：

推荐系统概述（一）

Github

标签: Algorithm practice

Jamest
关注 - 3
粉丝 - 24

« 上一篇： 个性化召回算法实践(四)——ContentBased算法
» 下一篇： 稀疏矩阵在Python中的表示方法

posted @ 2019-10-30 17:37 Jamest 阅读(2738) 评论(3) 编辑 收藏

评论列表

#1楼 2019-11-05 15:59 mantch

根据你写的代码，有个问题想问一下，最后的most_similar的输入不应该是movieid吗，怎么会是userid呢，我的理解是训练输入的是moveid，跟userid是没有关系的，最后most_similar应该是关于moveid相近的moveid。

支持(0) 反对(0)

#2楼 [楼主] 2019-11-05 16:16 Jamest

@ mantch
你说的有道理，谢谢指正

支持(0) 反对(0)

#3楼 2020-05-22 11:19 懒惰的星期六

就是movie id

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

1. Re:个性化推荐算法综述
你好，里面的图片可以更新一下

2. Re:个性化召回算法实践(三)算法
好的，不管怎么说，personalra
浅，多谢大佬。保持联系！
--努

3. Re:个性化召回算法实践(三)算法
@努力学python的小白 α是超参
可以试试，不同数据集效果不一
可以上git查查看，我也没有现

4. Re:个性化召回算法实践(三)算法
好的，我明白了，多谢多谢！对
定0.8的呢，我看有的人定的是
你有LDA主题模型困惑度确定的
--努

5. Re:个性化召回算法实践(三)算法
@努力学python的小白 不好意
后指定了K=30，所以输出了30
以有10个用户排名，是因为在
区分用户与物品，即我们可以
与该用户行为最...

阅读排行榜

1. 从目标检测到小目标检测(10
2. 图卷积神经网络(GCN)入门(
3. 机器学习面试问题总结(749
4. 深度学习面试问题总结(682
5. 深度排序模型概述（一）Wi
M(4653)

评论排行榜

1. 个性化召回算法实践(三)——
法(7)
2. 个性化排序算法实践(四)——
3. 个性化召回算法实践(五)——
4. Paper Reading:TridentNet(3
5. 个性化推荐算法综述(1)

推荐排行榜

1. weighted—LR的理解与推广
2. 稀疏矩阵在Python中的表示
3. 深度解析Graph Embedding(
4. [Leetcode Weekly Contest]1
5. 图卷积神经网络(GCN)入门(