Embedding向量召回在vivo个性化推荐中的实践

原创 vivo AI Lab vivo人工智能技术 2月27日

JEEK
容易脸红的编程大神

本文首先介绍Embedding召回框架和封装的算法，然后介绍Embedding召回在vivo手机的阅图锁屏、以及内置i主题App这2个推荐场景中的应用效果。

背景

推荐系统的算法部分一般分为两个阶段：召回和排序。召回主要负责从全量的物品池中快速筛选出用户可能会感兴趣的物品，缩小备选范围，为排序做准备，而排序阶段则是从召回的物品池中找出用户最感兴趣的物品。召回的结果决定了排序的上限，是推荐系统中非常重要的一个环节，本文主要介绍Embedding方法在召回阶段的应用。

召回一般要满足3个条件：
(1)高效性：在短时间内完成物品的召回
(2)相关性：尽可能召回匹配用户兴趣的物品
(3)多样性：召回物品尽可能多样化
因此召回一般采用多路召回，常见的有用户标签召回、热点召回、用户历史行为召回等。不同的召回策略从不同的角度出发，尽可能保证结果的多样性。这些传统的召回方式没有充分利用用户和物品的信息，召回效果可能不太理想。而利用复杂模型去做召回在性能上不能满足要求。因此，就出现了一种折中的方式，即Embedding向量召回。
Embedding一词最早出现在NLP领域，用一个稠密向量表示一个词，能解决传统One-Hot编码中维度太高的问题，是一种降维方式，同时Embedding携带了词的语义信息。这种方式不局限于词，也可以用于其他实体，比如推荐系统中的用户和物品，用户和物品的Embedding能够很好地表示内在信息和彼此之间的关联。Embedding向量召回就是利用这种思想，通过模型训练将用户和物品映射到同一个向量空间，即学习出用户和物品的Embedding向量表示。在召回阶段，计算用户和物品Embedding向量之间的相似度，保证召回的高效性。
因此，Embedding向量召回既可以利用较为丰富的特征和复杂的模型学习到用户的兴趣，又可以满足召回的高效性要求。本文首先介绍Embedding召回框架和封装的算法，然后介绍Embedding召回在vivo手机的阅图锁屏、以及内置i主题App这2个推荐场景中的应用效果。

Embedding向量召回

召回框架

我们实现的通用Embedding召回框架如图1所示。最底部是数据处理层，根据不同召回模型生成相应的数据；模型层封装了一些业界常用且效果较好的召回模型，模型训练完之后生成用户和物品Embedding向量；然后在离线评价层，我们会对用户和物品向量进行召回率、准确率、AUC等指标的评价，来判断模型的离线效果；最后是向量检索层，我们在向量检索工具Faiss中建立好物品向量的索引，然后通过用户向量去检索得到召回列表。推荐列表的生成有离线和在线两种方式，离线方式即利用 Faiss 工具离线计算好每个用户的召回列表，在线方式即实时采集用户数据并构建用户特征，请求模型计算出用户Embedding向量，最后使用Faiss查询得到物品召回列表。我们目前在项目中使用的是离线召回的方式。

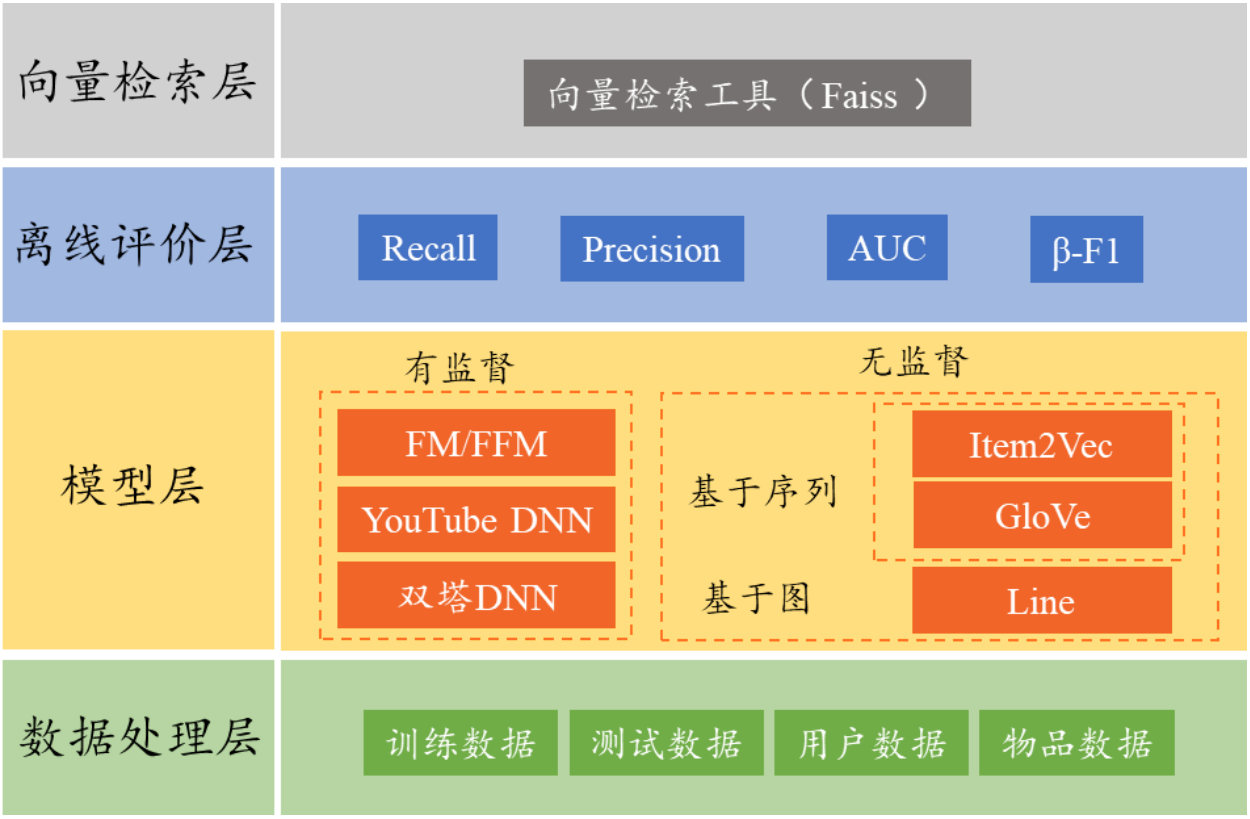


图1 Embedding召回框架

召回模型算法及实现

本小节主要介绍目前已封装好的召回算法，以及实现过程中的一些细节。首先我们尝试了几种传统的序列Embedding方法，包括Item2Vec、GloVe，之后进一步使用了Graph Embedding来做物品的Embedding，使用的算法是Line；除了这些常用的Embedding算法，我们也实现了业界常用的FM/FFM算法，以及深度学习Embedding算法，包括YouTube的单塔DNN模型和双塔DNN模型。下面按照无监督和有监督两大类来介绍。

无监督

1）Item2Vec

Item2Vec[1]由Oren Barkan和Noam Koenigstein于2016年提出，论文把 Word2Vec 的 Skip-Gram with Negative Sampling (SGNS)算法思路迁移到基于物品的协同过滤上。它是一种通过用户行为来理解内容的方式，使用包含一个隐层的神经网络，并基于一段时间内被点击物品具有内在相似性的假设，来学习物品Embedding向量。训练数据是用户的点击物品序列，物品间的共现为正样本，并按照物品的频率分布进行负样本采样。

实际应用时，我们基于 Skip-Gram和负采样来做 Item2Vec ，由于用户的点击序列不像自然语言那样具有严格的局部空间句法结构，并且实际业务中用户点击序列比较短，我们选取了7天的数据生成点击序列，另外，考虑到序列带来的影响，我们参照论文提到的方式，对

了大的数据生成点击序列。另外，为了避免序列带来的影响，我们参照论文提到的方式，对点击序列进行了Shuffle，并将上下文窗口设置为5，正负样本控制在1:10，隐层的Embedding Size 取32, 这些取值是效果和性能的折中，根据具体的业务可以调整。

2）GloVe

GloVe[2]（Global Vectors for Word Representation），是一个基于全局词频统计的词Embedding算法，Embedding向量捕捉到了单词之间一些语义特性。类似Item2Vec，我们可以把用户的点击序列看成一个句子，每个物品就是单词。其实现主要可以分为三步：

1. 根据点击序列数据构建共现矩阵，其权重根据两个物品在上下文窗口的距离d进行衰减1/d；

2. 构建词向量和共现矩阵之间的近似关系 $w_i^T \bar{w}_j + b_i + \bar{b}_j = \log(X_{ij})$ ；

3. 构造损失函数，作者在平方损失上加了一个权重函数，保证共现次数多的物品之间的权重不会过大，其公式如下：

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & if \ x < x_{max} \\ 1 & otherwise \end{cases}$$

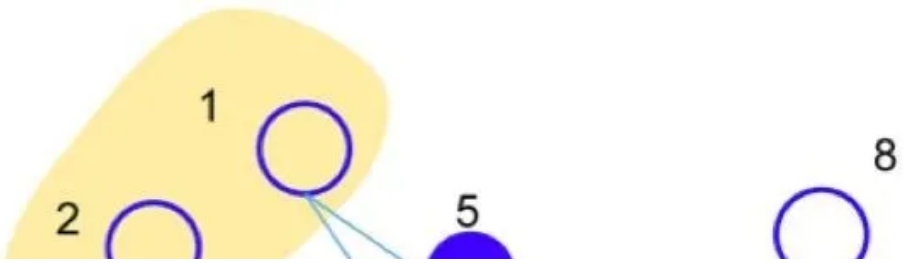
更多细节可以参考论文。

3）Line

Item2Vec和GloVe都是建立在“序列”样本（如点击列表）的基础上，但在推荐场景下，物品之间的关系更多呈现的是图（网络）结构，比如典型的场景是通过用户行为数据生成物品关系图，节点表示物品，边表示物品之间的关系，用低维、稠密、实值的向量表示网络中的节点，即物品向量。

其中比较有代表性的是微软亚洲研究院在 2015 年发布的Line[3]（Large-scale Information Network Embedding），它定义了两种相似度来表征节点之间的关系，即一阶相似度（存在直接相连的边，衡量两个节点自身的相似度）和二阶相似度（是否存在相同的邻居节点，衡量两个节点邻近网络结构之间的相似度），使得训练出来的Embedding向量既保留局部网络结构信息又保留了一定的全局网络结构信息。

在业务中我们将用户的点击序列输入模型，构建图网络，然后分别训练一阶相似度模型和二阶相似度模型，得到两个向量，最后将其拼接得到物品向量。



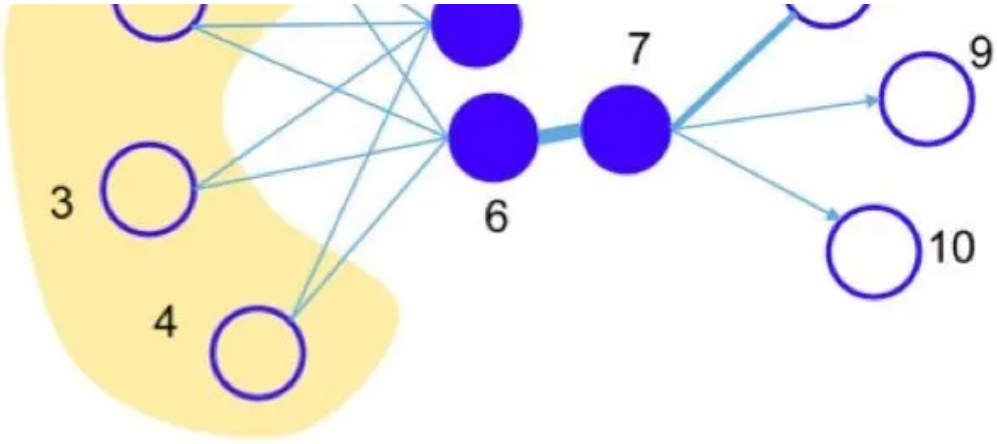


图 2 二阶相似度，例如5和6共享邻居，是二阶相似的

4）小结

上述三种模型都会产生物品向量，那用户向量又是如何得到的呢？我们的做法比较简单，直接选取用户最近一段时间内点击/下载的物品，查询对应物品的物品向量，然后取平均得到用户向量，即Average Pooling的思想，以此反映用户最近的兴趣。

上述模型学习出来的物品向量具有很好的相似度，但缺点也比较明显：

- 只考虑了训练样本中物品与物品之间的局部/全局共现关系，没有利用丰富的用户侧特征，用户个性化不够；
- 需要用到用户的行为数据，无法解决冷启动问题，另外使用短期的用户行为进行训练，只能覆盖活跃用户，且只能评估用户的短期兴趣。

有监督

1）FM

FM算法（Factorization Machine）是在CTR预估中常用的模型，它主要解决特征的二阶组合问题，对于稀疏的数据具有很好的学习能力。它其实就是LR模型加上特征之间的二阶交叉项，公式如下

$$\hat{y}(x) = w_0 + \sum_{i=1}^{F} w_i x_i + \sum_{i=1}^{F-1} \sum_{j=i+1}^{F} < v_i, v_j > x_i x_j$$

x是特征的值，v是特征的Embedding向量，每个特征学习一个Embedding向量。特征交叉时，以两个特征的Embedding向量的内积作为交叉项的权重。

但从原始的FM模型中我们无法得到用户和物品的Embedding向量。因此，我们采用[9]中提出的方法，简化FM模型，只考虑用户特征和物品特征之间的交叉，目的是为了抽离出用

户和物品的Embedding向量。如果我们只保留用户特征和物品特征之间的交叉，二阶交叉项可以改写为：

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \approx \sum_{i\in U}\sum_{j\in I} < v_i, v_j > x_i x_j = < \sum_{i\in U} v_i x_i , \sum_{j\in I} v_j x_j >$$

$\sum_{i\in U} v_i x_i$ 是各用户特征乘上对应的Embedding之后求和，维度与每个用户特征的Embedding维度相同，

$\sum_{j\in I} v_j x_j$ 是各物品特征乘上对应的Embedding之后求和。假设特征Embedding的长度都是n维，最后拼接上一阶项和bias，用户和物品的Embedding向量长度为n+3，如图3所示。

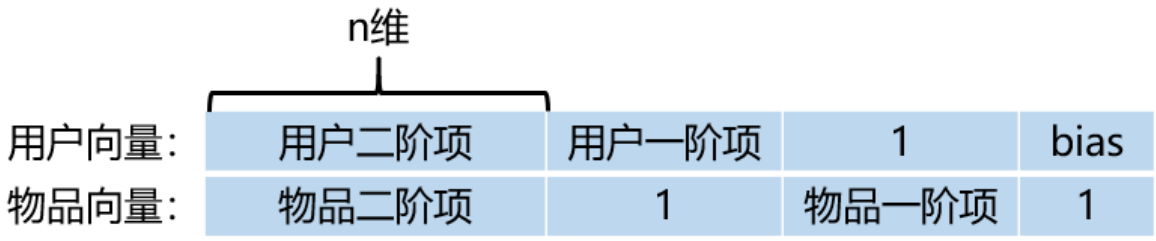


图 3 FM用户和物品Embedding向量

2）FFM

FFM算法（Field-aware Factorization Machine）是FM算法的改进版。FFM相对于FM更加细致，它认为每个特征与其他不同域的特征交叉时使用的Embedding向量不同，即对于每一个特征域学习一个Embedding向量。如果一共有F个特征域，那么每个特征都会学习F-1个Embedding向量（除了自己所在的特征域）。类似于FM的简化方法，为了简化FFM得到用户的Embedding向量和物品的Embedding向量，我们按照[10]中提出的方法，只考虑用户特征和物品特征之间的交叉。

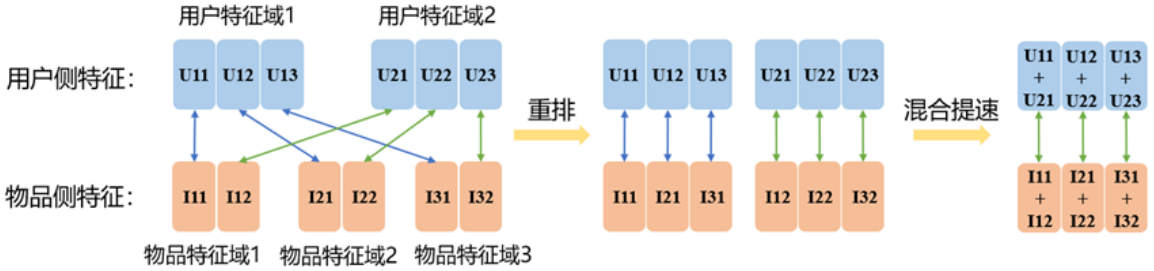


图 4 FFM用户和物品向量二阶部分

如图4所示，假设用户侧有M个特征域，物品侧有N个特征域，则每一个用户特征都会学习N

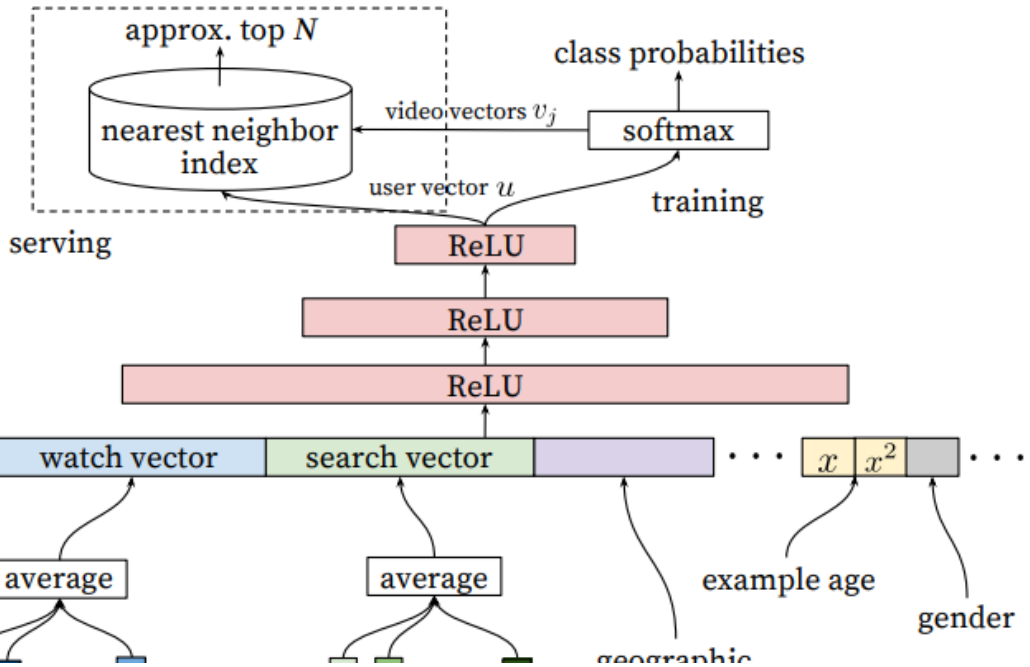
个Embedding向量，每个物品特征会学习M个Embedding向量。经过重新排序后用户侧Embedding向量可以和物品侧Embedding向量一一对应。将向量拼接就可以得到用户和物品的Embedding向量，向量的长度为M*N*K（K为特征Embedding向量的长度）。但是一般特征的数量会比较大，因此拼接得到的用户和物品的向量长度太长，影响向量检索的速度。[10]中进一步提出了几种简化的方法，我们采用了“混合提速”策略。即用户侧特征对应同一物品侧特征域的Embedding先相加再拼接，物品侧同一特征域的Embedding先相加，不同特征域之间再拼接。另外，一阶项和bias项的添加方式和FM相同，最后用户和物品Embedding向量的长度为N*K+3。

3）YoutTube DNN

YouTube DNN算法来自于YouTube 2016年发表在RecSys上的文章。它把推荐问题看成一个超大规模多分类问题。网络结构如图5所示，用户历史行为特征由物品Embedding求平均表示，另外拼接上用户的属性特征。经过一个全连接网络之后得到用户的Embedding向量。深度神经网络的目标就是在给定用户历史行为、用户特征以及上下文的情况下，学习出用户的Embedding向量，然后用于Softmax分类器来召回物品。在物品库V里选择的物品为第i个视频的概率，其数学表达式如下：

$$P(\omega = i|U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

其中 $u \in R^N$ 表示（用户，上下文）的embedding向量， $v_j \in R^N$ 表示每个候选物品的embedding向量。



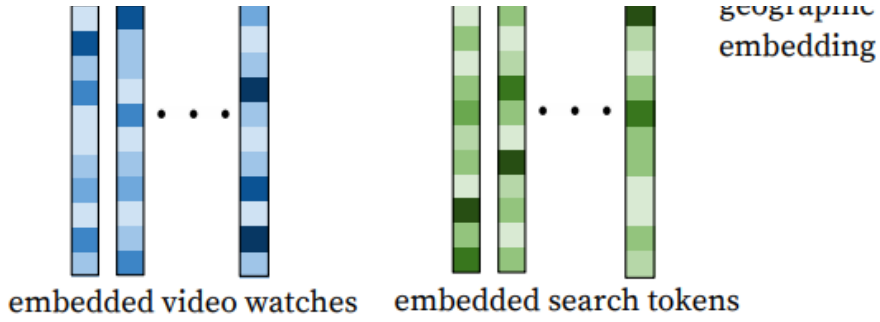


图 5 YouTube DNN模型

在我们的实现中，模型输入层物品的Embedding没有经过预训练，而是共用Softmax矩阵中的物品Embedding向量，发现有更好的效果。此模型的特点是可以同时利用用户的历史行为特征以及用户的属性特征，对用户的兴趣有更好的挖掘。

4）双塔DNN

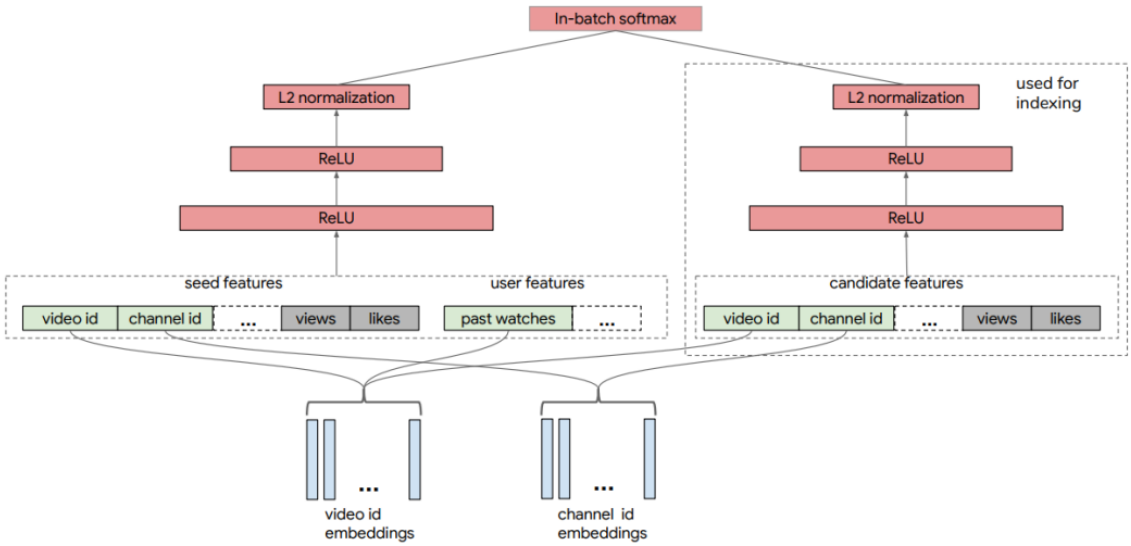


图 6 双塔 DNN模型

双塔DNN模型来自于YouTube 2019年发表在RecSys上的文章，在业界其实也早有应用。相对于只有用户塔的DNN模型，它加入了物品塔。用户塔还是输入用户的历史行为特征和用户属性特征，物品塔可以输入物品的id类特征和属性特征。两个塔的最后一层输出分别代表用户的Embedding向量和物品的Embedding向量，然后通过Softmax函数得到点击概率。相对于单塔的DNN模型，双塔模型加入了更多物品侧的特征，能够更好地刻画物品。

5）小结

FM/FFM算法为了实现Embedding召回做了一定的改造，与原始的FM/FFM算法有一定区

FM/FFM算法为了实现Embedding召回做了一定的改造，与原始的FM/FFM算法有一定区别。在FM算法实现中，我们省略了上下文特征，以及用户特征、物品特征的内部交叉，后续可以考虑加上这些特征；在FFM算法实现中除了部分特征的省略，为了减小向量维度，用了混合提速的方法，也会损失一定的模型精确度。

在YouTube DNN和双塔DNN的实现中，需要对id特征做Embedding操作，常规方法是用一个[Index_size，Embedding_size]的Variable矩阵做Embedding参数，然后根据id查表得到相应的Embedding向量。但是实际的id特征往往不是从0 开始，甚至不一定是整数。利用Tensorflow的lookup.index_table_from_tensor函数可以将原始id特征映射到从0开始的index，但是还是需要设置Variable矩阵的大小，而且需要预留一定的数量给新增的id。因此我们使用了vivo的Embedding Table工具，在模型中直接使用物品的原始id，并且Embedding Table可以动态增加。另外，考虑到性能、训练数据量和模型参数量之间的关系，当物品数量非常大的时候，我们会筛选出较为活跃的物品进行Embedding，使得到的Embedding能较好地反映物品的特征。尽量保证模型参数和训练样本数的比例在1:20~1:10。

以上模型训练完之后并不能直接得到用户和物品的Embedding向量，除了YouTube DNN模型参数中包含物品的Embedding向量。其余模型需要为每一个用户或物品构造一条预测数据输入到模型中，通过predict过程将模型的最后一层输出结果作为用户或者物品的Embedding向量。如果是离线处理，则选择用户或者物品最新的特征。

算法优缺点比较：

- 有监督学习算法目前都已经实现了增量更新，可以提升模型更新速度，但无监督算法目前还没有实现；
- 无监督算法存在的普遍问题是，偏向热门物品，对长尾物料挖掘不足；
- 有监督的算法通过利用丰富的物品侧、用户侧特征，可以进一步实现个性化，另外像双塔DNN、FM/FFM通过加入物品特征，可以解决物品冷启动问题，并且通过实时获取新物品特征，可以让新物品得以曝光。

实际业务落地

1）阅图锁屏

vivo手机的阅图锁屏功能为用户推荐手机锁屏界面的壁纸，通过推荐算法，我们可以给用户推荐个性化的壁纸，提升用户的使用体验。阅图锁屏主要有两个场景，一个是用户点亮屏幕时显示的锁屏壁纸，我们统一称为锁屏；另一个是用户滑动壁纸时显示的滑屏壁纸，我们统一称为滑屏，因此滑屏是用户主动的行为，更能表现出用户的兴趣。

目前，我们在阅图锁屏业务上线了4个Embedding向量召回算法，效果如图7所示，效果提升是相对于传统协同召回算法。在滑屏场景中，YouTube DNN效果最好，因为相对于其他算法，YouTube DNN中也加入了用户的部分画像特征，后续可以考虑加入更多的用户侧特征。在锁屏场景下，用户的行为比较随机，存在大量的无效曝光，因此算法的效果不太符合预期，但是基本都有提升，其中GloVe的效果最优，Line算法还在优化中。

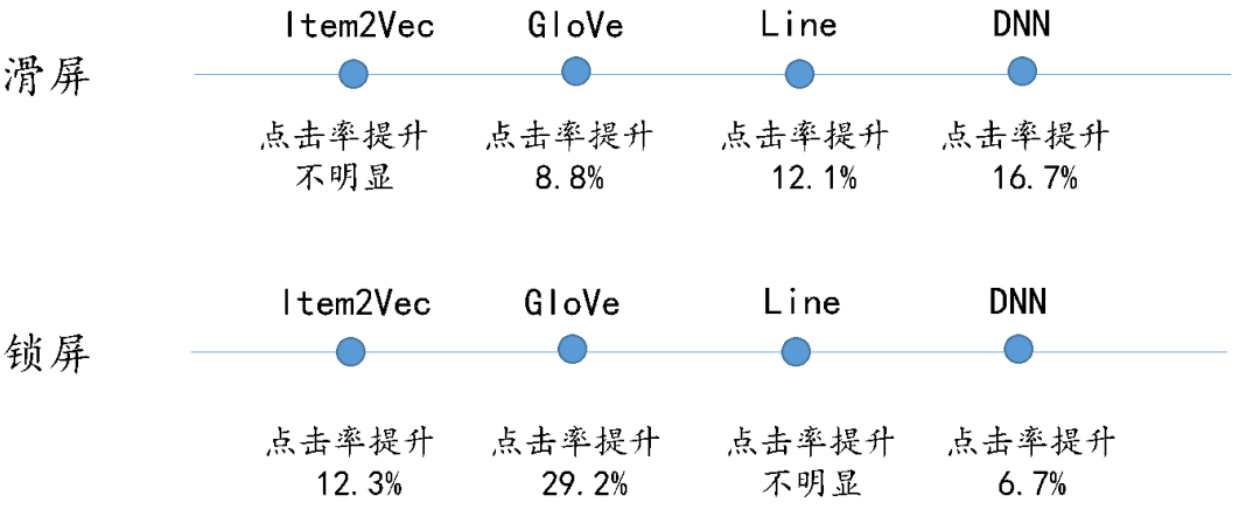


图7 Embedding召回在阅图锁屏的效果

2） i主题

i主题应用给用户推荐个性化的手机主题，提升用户的使用体验。我们在i主题推荐中上线了Item2Vec，GloVe，Line，YouTube DNN这4个Embedding召回算法的A/B测试，线上效果提升如下图所示，目前YouTube DNN没有使用很多特征，效果表现一般，但有较大提升空间。

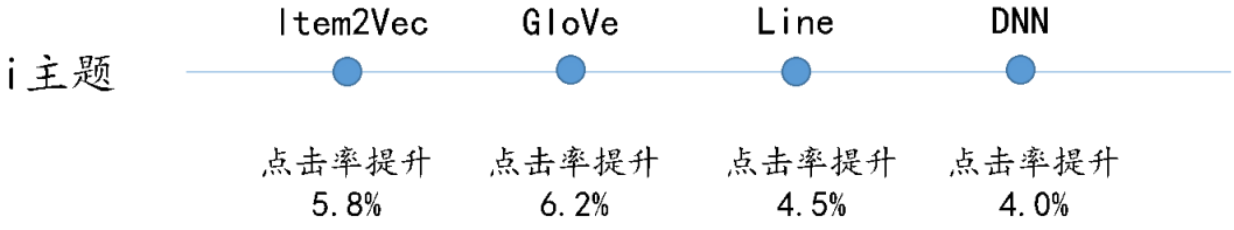


图 8 Embedding召回在i主题中的效果

我们对GloVe、Line和YouTube DNN的物品Embedding向量相似度进行了可视化展示，为

每个物品召回最相似的物品，图9展示了i主题的几个例子，左边是目标主题，右边是用对应算法产生的相似度前3的主题，可以看出通过用户行为能较好地学习到物品内容的相似度。Line和GloVe的相似度较好，YouTube DNN多样性会比较好。

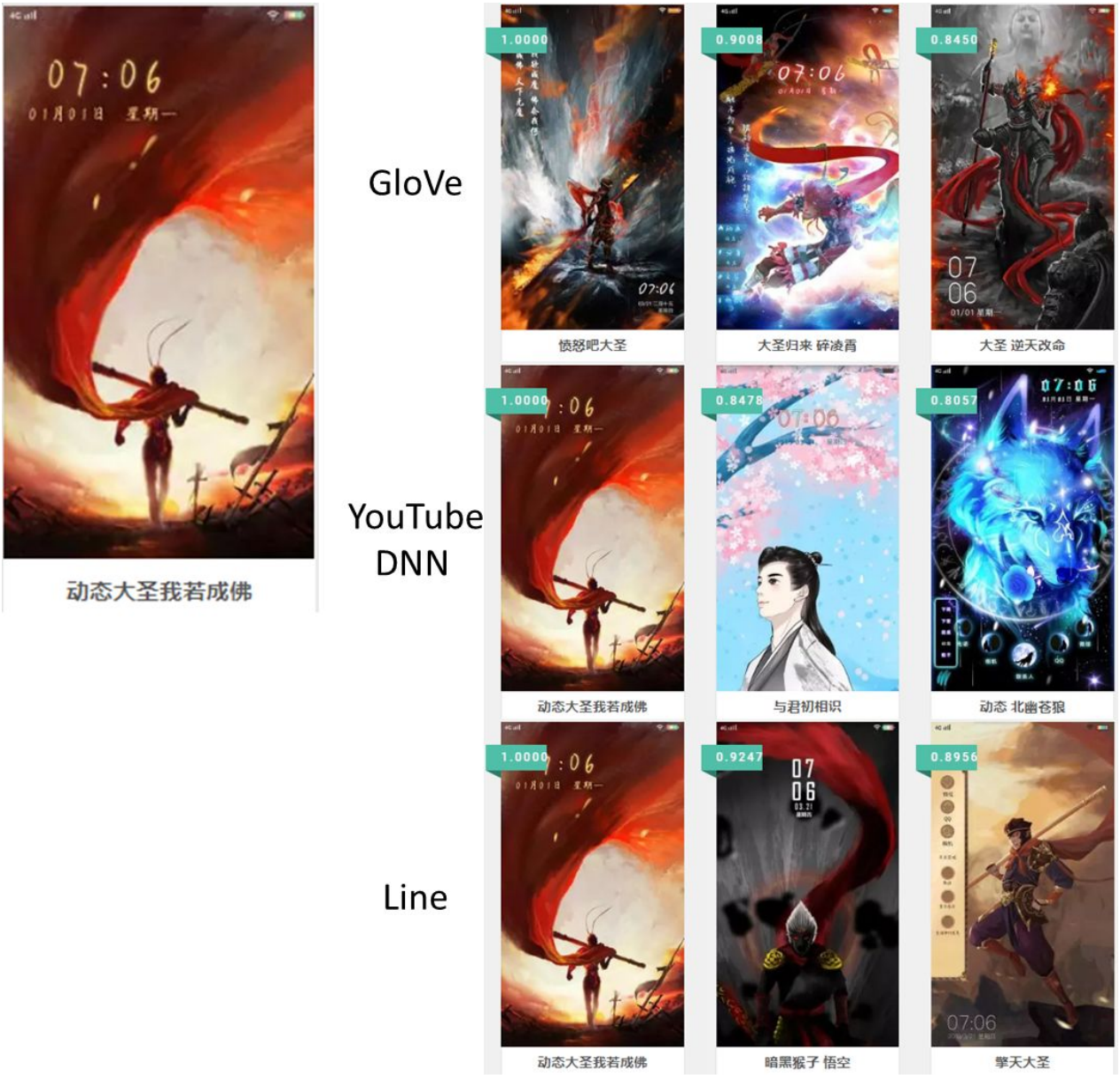


图9 i主题物品相似度

不足和展望

目前实现的框架还有很多改进之处：

1. 无监督的方法，例如Item2Vec、GloVe、Line，都是生成物品Embedding向量，而用户Embedding向量是通过Average Pooling得到的，这种方式比较简单，未来可以引入Attention机制，对用户感兴趣的Item加权重来表征用户兴趣。另外，基于图的Embedding算法有较多前沿研究，许多算法都取得了不错的效果，我们目前使用的Line算法只是比较基础

的版本，其实还是转化为序列处理，之后可以考虑尝试其他更先进的图Embedding算法。

2. 目前在阅图锁屏和i主题两个业务中只是实现了离线召回，如果采用在线召回的方式，实时获取用户的特征输入模型得到用户的Embedding向量，能够学习到用户的实时兴趣。

目前离线直接通过计算召回率、准确率等指标评估模型的效果，可以直接评估Embedding的质量。也可以分开评估用户向量和物品向量的好坏，参考Word Embedding评估的方法，使用Relatedness、Analogy、Categorization以及降维可视化等方法。

参考文献

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

[2] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[3] Tang J, Qu M, Wang M, et al. Line: Large-scale information network embedding[C]//Proceedings of the 24th international conference on world wide web. International World Wide Web Conferences Steering Committee, 2015: 1067-1077.

[4] Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations[C]//Proceedings of the 10th ACM conference on recommender systems. ACM, 2016: 191-198.

[5] Yi X, Yang J, Hong L, et al. Sampling-bias-corrected neural modeling for large corpus item recommendations[C]//Proceedings of the 13th ACM Conference on Recommender Systems. 2019: 269-277.

[6] 推荐系统的中 EMBEDDING 的应用实践https://lumingdong.cn/application-practice-of-embedding-in-recommendation-system.html#Graph_Embedding

[7] 深度学习在美图个性化推荐的应用实践 https://zhuanlan.zhihu.com/p/87466510?utm_source=wechat_session&utm_medium=social&utm_oi=573454785584304128&from=groupmessage&isappinstalled=0

[8] 深度学习中不得不学的Graph Embedding方法 https://zhuanlan.zhihu.com/p/64200072

[9] 推荐系统召回四模型之：全能的FM模型https://zhuanlan.zhihu.com/p/58160982

[10] 推荐系统召回四模型之二：沉重的FFM模型https://zhuanlan.zhihu.com/p/59528983