Embedding 模型在推荐系统的应用

原创 可怜小熊熊 搜狐技术产品 2019-03-07

简介

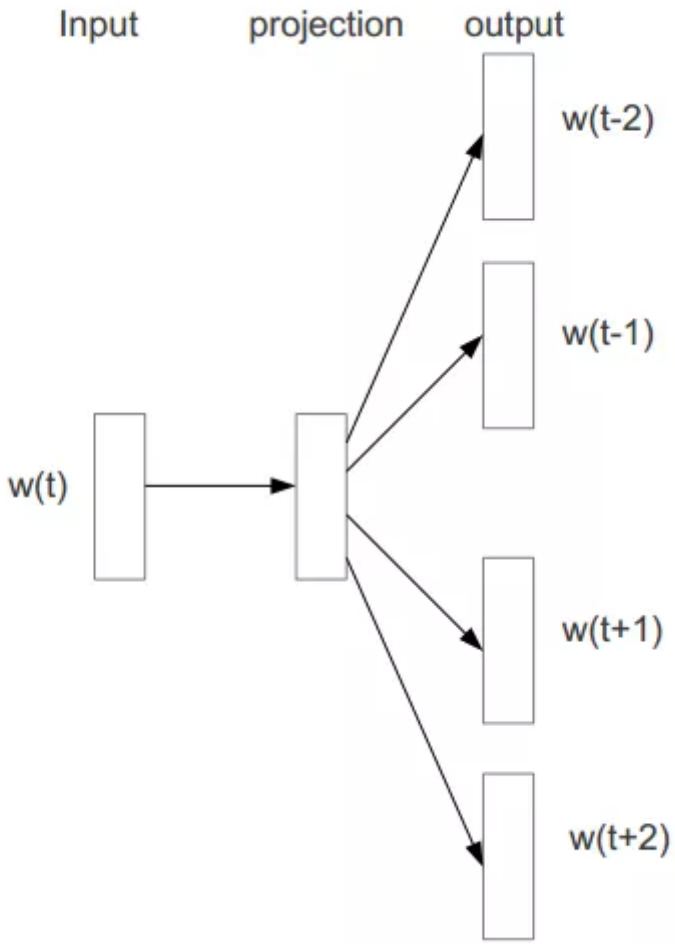
从word2vec问世以来，人们已经改变了对词语建模的思路。把词语的表示从高维稀疏转变为低维稠密，embedding不仅在nlp任务中应用越来越广泛，在搜索，推荐，广告等领域也逐渐被引起重视。把用户的行为序列看成是词语序列，训练出行为的向量表示，将向量应用在召回和排序场景已成为不错的baseline。但是，如何在基线上进行优化，需要对模型和业务有深刻的理解。

《Real-time Personalization using Embeddings for Search Ranking at Airbnb》获得了2018 KDD最佳论文，这篇论文结合业务场景对Word2vec算法做了改进，针对性的修改目标函数，构造数据集，解决了业务难点，达到了不错的效果。借鉴论文的优化思路，并结合搜狐新闻客户端的业务特点，我们在新闻推荐的召回和排序阶段也进行了改进。本文首先介绍word2vec模型，模型参数调节的思路，接着讲解论文中在房源推荐场景的优化方向，最后介绍在新闻推荐场景中的应用。

SGNS模型

模型结构

Word2Vec模型，主要有Skip-Gram和CBOW两种模型，从网络结构上理解，Skip-Gram是根据中心词来预测上下文，CBOW是根据上下文，来预测中心词。两种模型分别有两种高效优化算法来进行求解，分别是Hierarchical Softmax和负采样。其中，skip-gram和负采样相结合是较为常用的手段，简称为SGNS（Skip-gram with negative sampling）。



首先，我们需要设定一个目标，基于训练数据构建一个神经网络，当这个模型训练好以后，并不会用这个训练好的模型处理新的任务，我们真正需要的是模型权重矩阵。在skip-gram模型中，会把输入层到隐藏层的权重矩阵当做词向量。实际上隐藏层到输出层的权重也可以用来作为词向量，在实验中发现，输入层到隐藏层和隐藏层到输出层共用一套权重矩阵也是可以的，这样还可以节省模型占用内存。

模型超参数

模型主要有以下几个超参数，分别是迭代次数N，滑动窗口大小L，负采样个数，负采样概率分布，下采样概率分布和阈值t，学习率a。

负采样概率分布

$$P(c) = \frac{f(c)^a}{\sum_c f(c')^a}$$

f（c）表示词语c频率，a调节因子。通过公式不难得出，词频越高的词语，越容易被当作负样本。一般词频符合长尾分布，大量低频词不会被采样到，低频词的向量很难被充分训练。通过对a的调节，来降低高频词的采样概率，提高低频词的采样概率。当取1时，完全按照词频来采样；当a取0时，等概率采样。a通常取0.75。

下采样

$$P(c) = \frac{f(c)-t}{f(c)} - \sqrt{\frac{t}{f(c)}}$$

下采样在窗口滑动过程中生成正样本对时进行，主要是为了防止高频词形成的正样本对过多导致模型效果变差。下采样概率如上所示，其中f(c)表示词语出现的频率，t是下采样阈值，词频越高的词语，被丢弃的概率就越高。

滑动窗口

滑动窗口通常取固定值为5，滑动窗口的大小决定了模型的计算量。滑动窗口设置太小，每一轮迭代速度虽然快，但是会损失掉部分词语之间联系，导致向量表示学习不充分；滑动窗口设置太大，模型计算量增加，由于窗口增大 ，会给不相关的词语建立联系，误导模型训练。

不同的场景，不同的行为，滑动窗口大小也需要随之而改变。滑动窗口的大小在一个模型中甚至可以是可变的值，由中心词的“影响力”决定。例如推荐场景，用户不同的行为，有着不同的影响，普通的点击，窗口可能选择5，对于点赞，分享等较强的信号，会适当的扩大滑动窗口为10。

负采样个数

负采样主要是为了降低模型计算量。如果没有负采样，模型需要把词汇表中没有出现在滑动窗口的词语当作负样本。然而在实际训练过程中，并不需要这么多的负样本，过多的负样本会导致模型学偏。

负采样的个数和滑动窗口的比例尽量控制在0.1-10之间，滑动窗口决定了正样本的数量，负采样的个数决定了负样本的个数，正负样本尽量不要差距太大，建议负采样的个数和滑动窗口的比例控制为1：1。

迭代次数和学习率

迭代次数和学习率的设定，需要综合考虑计算资源，模型规模，通过观察训练过程中loss的变化来设定，源码中学习率是0.025，每处理10000个样本会按照比例减小学习率；同时设定了学习率的最小值，保证学习率不会无限制的减小。

调参

在调节参数时，可以先设置一个较大的学习率，调节负采样个数和滑动窗口大小，之后再逐渐改变下采样阈值和负采样系数，最后调节学习率和迭代次数，直至模型收敛。得到向量表示后，进行聚类，做可视化，便于诊断问题。

房源推荐场景应用

在租房场景中，使用SGNS获得房源的向量表示。

样本构造

将用户的历史行为序列切分为session，切分原则：对于session中相邻的行为，时间间隔不会超过30分钟。构造好训练样本后，接下来获取房源的向量表示。把用户点击的房源看作是词语，一个session中的点击序列看作是句子，训练模型，得到房源的embedding。

优化目标

$$\arg\max_\theta \sum_{(n,c)\in D_p} \log \frac{1}{1+e^{-v_c' v_n}} + \sum_{(n,c)\in D_n} \log \frac{1}{1+e^{v_c' v_n}}$$

$$+ \log \frac{1}{1+e^{-v_{l_b}' v_l}} + \sum_{(l,m_n)\in D_{mn}} \log \frac{1}{1+e^{v_{m_n}' v_l}}$$

以上便是目标函数，l表示当前房源，c是当前房源的context， 表示输入层到隐藏层的权重，V_c是隐藏层到输出层的权重，D_p是正样本集合，D_n是随机负采样样本的集合，D_{mn}是同城负采样的集合，V_{l_b}是预定房源的权重。目标函数的前两项分别表示滑动窗口移动过程和负采样过程。

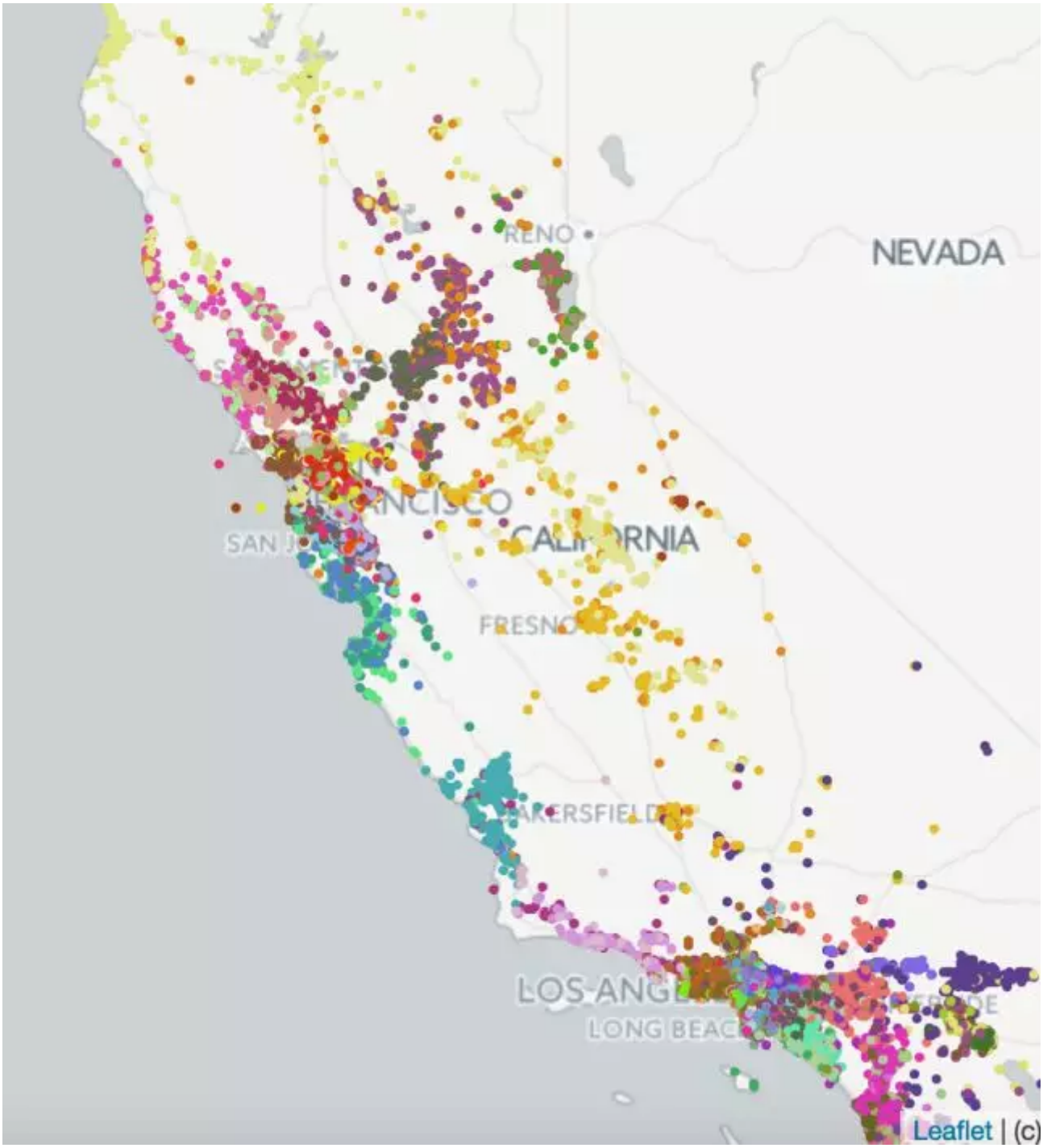
在租房场景下，最终目标是提升用户的租房成交率。所以对于用户的预定行为和点击行为，需要有不同的处理方法。根据常识，预定行为发生的频率会远远小于点击行为，但是预定行为最能反映出用户的兴趣，也和业务目标保持一致，所以这种样本很宝贵，需要增加预定行为对模型的影响。把预定的房源作为全局的上下文，用户预定一个房源，通常是经过深思熟虑后的决定，在预定之前一定会观望很久，反而点击行为发生的更加随意，作用范围会稍小。本质上是对不同的行为设定了不同的滑动窗口，相当于对预定行为做了加权。对应于目标函数的第三项。

目标函数第二项进行了全局负采样，但是根据租房的特殊性，同一session内的房源属于同一个城市，倘若进行全局负采样，极大概率会采样到其他城市的房源，那么在这种样本分布下学习到的embedding，会只学习到了城市信息，然而房源类型，价格等信息会被忽略掉。于是，论文在房源的同城房源市场中进行负采样，降低了城市因素的影响。

向量评估

训练好向量后，通过可视化的方法，验证向量是否学习到了想要的信息。

向量能否表达地理位置的相似性。对某城市的房源embedding做聚类，同类的房源用相同颜色标注在地图上，相同颜色聚集到一起。这说明学到了地理位置信息。



除了在地理位置角度的可视化，在房价，租金，建筑风格，户型等也可以进行同样的验证，便于诊断问题。embedding的评估是个难点，直接把embedding放到实际生产中，会拉长迭代周期，版本的迭代也会减缓，令人疲惫不堪。多开发一些可视化工具，大大提高了迭代速度，对于研究和诊断问题都提供了极大的便利，也是值得借鉴的地方。

用户属性和房源属性编码

session数据作为训练样本，构造的点击房源序列往往都属于同一个城市，同城房源适合短期兴趣场景。倘若使用用户的预定信息表示用户长期兴趣，存在以下几个难点：
预定样本少，数据稀疏，很多用户只有一个预定记录。再者，有的用户前后两次预定记录间隔时间太长，用户的身份可能发生变化，预定房源的偏好也会随之改变，会导致出现差异较大的样本序列。

为了避免直接使用用户id和房源id所带来的稀疏性，论文改为对用户属性id和房源属性id做embedding。房源有多重属性，位置，房源类型，价格等。对每种属性都做编码。

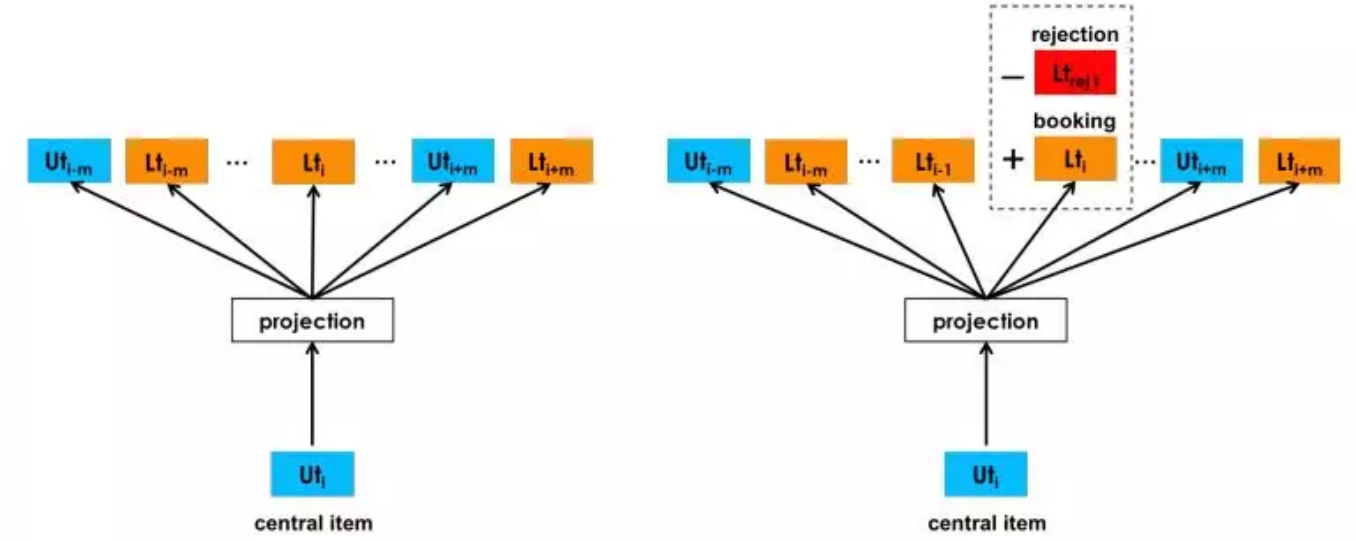
Buckets	1	2	3	4	5	6	7	8
Country	US	CA	GB	FR	MX	AU	ES	...
Listing Type	Ent	Priv	Share					
$ per Night	<40	40-55	56-69	70-83	84-100	101-129	130-189	190+
$ per Guest	<21	21-27	28-34	35-42	43-52	53-75	76+	
Num Reviews	0	1	2-5	6-10	11-35	35+		
Listing 5 Star %	0-40	41-60	61-90	90+				
Capacity	1	2	3	4	5	6+		
Num Beds	1	2	3	4+				
Num Bedrooms	0	1	2	3	4+			
Num Bathroom	0	1	2	3+				
New Guest Acc %	<60	61-90	>91					

用户也有多重属性，语言，画像，市场，预订次数等。

Buckets	1	2	3	4	5	6	7	8
Market	SF	NYC	LA	HK	PHL	AUS	LV	...
Language	en	es	fr	jp	ru	ko	de	...
Device Type	Mac	Msft	Andr	Ipad	Tablet	Iphone	...	
Full Profile	Yes	No						
Profile Photo	Yes	No						
Num Bookings	0	1	2-7	8+				
$ per Night	<40	40-55	56-69	70-83	84-100	101-129	130-189	190+
$ per Guest	<21	21-27	28-34	35-42	43-52	53-75	76+	
Capacity	<2	2-2.6	2.7-3	3.1-4	4.1-6	6.1+		
Num Reviews	<1	1-3.5	3.6-10	> 10				
Listing 5 Star %	0-40	41-60	61-90	90+				
Guest 5 Star %	0-40	41-60	61-90	90+				

对用户（房源）所有属性做笛卡尔积，得到用户的属性编码。例如某用户的编码为“旧金山-英语-有画像-照片-预定2晚-偏好价格60$”，从表中可以看出，有一些用户的属性是动态的，所以同一个用户id，会有不同用户类型。例如用户偏好价格，用户每进行一次预定，偏好价格都会有波动；例如用户预定次数也会逐渐增多。

优化目标



把同一个用户的预定房源看作是一个句子，每一个预定行为中的usertype和listtype组合当做词语，同样采用一个滑动窗口来进行更新向量，但是更新的目标有所变化。

$$\arg\max_\theta \sum_{(u_t,c)\in D_{book}} \log \frac{1}{1+e^{-v_c' v_{u_t}}} + \sum_{(u_t,c)\in D_{neg}} \log \frac{1}{1+e^{v_c' v_{u_t}}}$$

当中心词为用户类型时，选择房源类型作为上下文，目标函数如上所示。

$$\arg\max_\theta \sum_{(l_t,c)\in D_{book}} \log \frac{1}{1+e^{-v_c' v_{l_t}}} + \sum_{(l_t,c)\in D_{neg}} \log \frac{1}{1+e^{v_c' v_{l_t}}}$$

当中心词为用户类型时，选择房源类型作为上下文，目标函数如上所示。 当中心词为房源类型时，选择用户类型作为上下文，目标函数如上所示。其中D_{book}和D_{neg}分别表示用户的预定行为和拒绝行为。

这样构造模型可以把用户类型和房源类型在同一个向量空间中表示，使用用户向量和房源向量的相似度来表示用户对房源的感兴趣程度。

新闻推荐场景应用

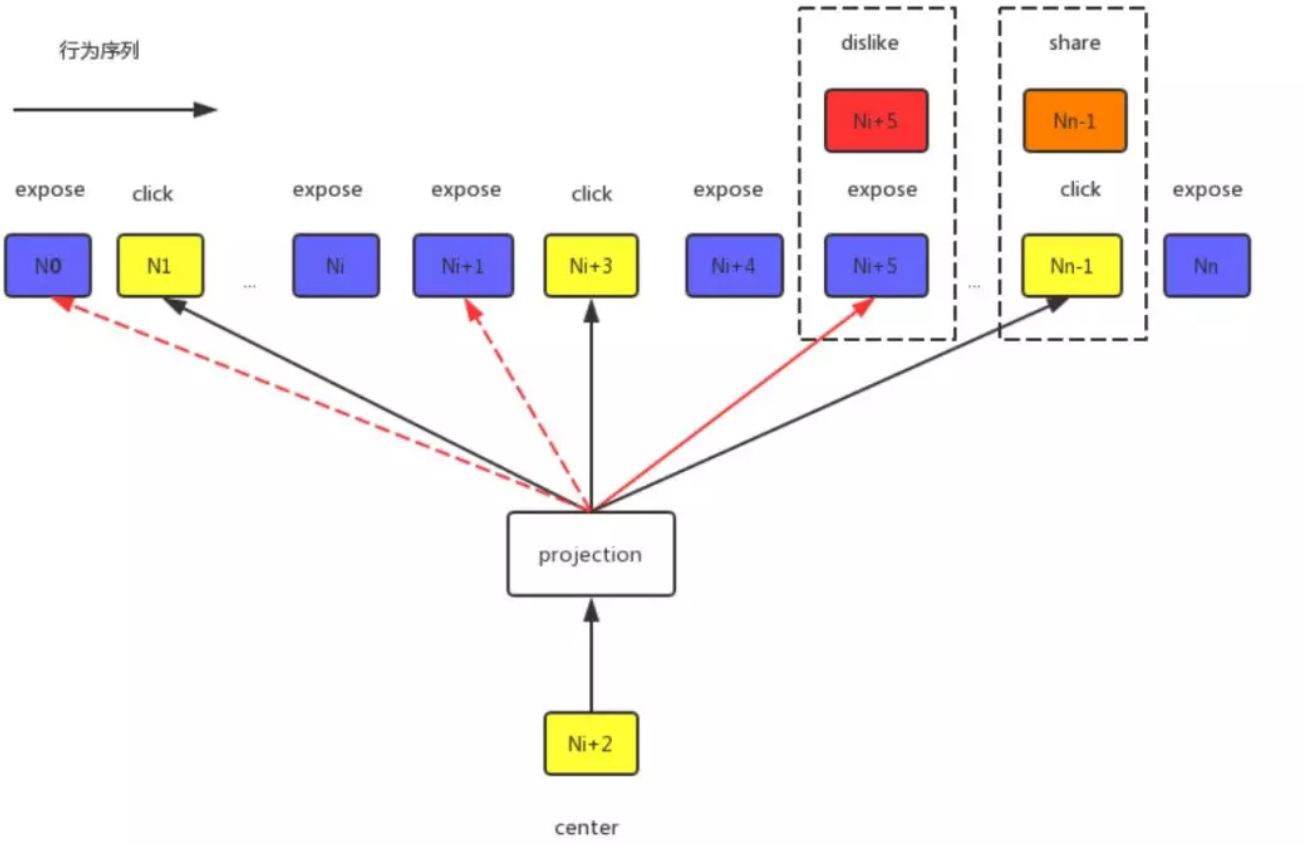
搜狐新闻推荐系统面向千万级用户，推荐系统需要迅速从十万级新闻池中检索出用户可能感兴趣的文章，召回阶段完成了新闻的初步筛选，这个过程直接影响到用户的体验。如何表示新闻，对推荐系统是一个考验。传统的表示方法，分类，主题，关键词等，比较适合精准的匹配，具有稀疏性，对于部分新闻无法奏效。通过向量表示新闻，很好的解决了这个问题。

数据处理

清洗异常数据，形成用户的行为序列。用户行为包括点击，曝光，分享，收藏，评论，不感兴趣等。把行为序列按照时间间隔进行切分，相邻两次行为超过6小时则把序列切分开。过滤掉长度过小和过大的行为序列，能得到千万级别的用户行为序列。每一个用户行为序列由一组新闻id组成，每个新闻id对应一个行为标签，表示点击，曝光，分享，收藏，评论，不感兴趣。

模型训练

数据处理后，我们拥有千万级别的用户行为序列，用户行为序列包括用户的多种行为，并按照行为时间排序。如下图所示，蓝色区域表示用户的曝光新闻，黄色表示点击新闻。用户阅读新闻后，如果很感兴趣，会分享给其他人，该行为用橙色标注。倘若用户对推荐的内容不感兴趣，会点击不感兴趣按钮，此类行为用红色标注。



训练过程沿着行为序列从左到右进行，依次选择每一个行为作为中心。只有当点击行为作为中心时才会更新模型，如果中心是曝光时，则直接跳过。当中心行为是点击时，会选择相邻的k次点击，构成正样本对。同时会进行全局负采样，按照新闻的曝光量进行采样。

但是全局负采样存在一个缺陷，可能采样到的新闻是用户感兴趣的，但是没有给用户曝光。为了解决这类问题，在用户曝光序列进行负采样。根据新闻客户端的特点，每次给用户展现15条新闻，用户需要向下滑动才能阅读完15篇新闻，有些情况下，用户并没有完全向下滑动，就开始新的请求，因此排位靠后的新闻可能并未给用户真正曝光。所以在用户点击排位之前的曝光序列中进行负采样。

除了用户的点击，曝光行为外，还有分享，不感兴趣。这两类行为具有更强的倾向性，模型给予这类行为更大的作用范围，作用于整个行为序列。用户的评论，具有情感倾向，包括不同强度的正向和负向，会使模型更加复杂，暂时并未引入。

$$\arg\max_\theta \sum_{(n,c)\in D_p} \log \frac{1}{1+e^{-v_c^{'} v_n}} + \sum_{(n,c)\in D_n} \log \frac{1}{1+e^{v_c^{'} v_n}}$$

$$+ \log \frac{1}{1+e^{-v_{n_s}^{'} v_n}} + \sum_{(n,c)\in D_e} \log \frac{1}{1+e^{v_c^{'} v_n}} + \log \frac{1}{1+e^{v_{n_d}^{'} v_n}}$$

通过上面的描述，模型目标函数如上所示，n表示当前处理的中心新闻，c表示上下文，D_P表示点击样本对，D_n表示全局负采样样本对，V_{ns}表示分享的新闻，D_e表示曝光序列负采样样本，V_{nd}表示不感兴趣新闻。V表示隐藏层到输出层的权重，表示输入层到隐藏层的权重。式子的前两项表示滑动窗口移动过程中，正样本的处理和全局负采样，第三项表示用户分享行为，第四项为在曝光序列中负采样过程，第五项是不感兴趣行为的处理方法。通过随即梯度下降法，不断的迭代，使得目标函数最大，获取新闻的向量表示。

召回阶段应用

离线训练好新闻的向量后，计算新闻的相似度矩阵，将相似度矩阵保存在推荐系统中，用户请求推荐服务时，查询用户的点击历史，收藏历史等，在相似度矩阵中获取与点击历史、收藏历史相似的新闻，进入排序阶段。

排序阶段应用

训练好的新闻向量，推荐系统会把它用在实时个性化排序阶段。

接收用户的实时行为日志，用户的点击行为，分享，收藏，不感兴趣等。分别构造用户点击，分享，不感兴趣的新闻，和候选新闻embedding的余弦相似度，把相似度进行离散化，丢到排序模型进行训练。具体特征如下：

特征名字	特征含义
EmbClickSim	点击新闻和候选新闻的相似度
EmbDislikeSim	不感兴趣新闻和候选新闻的相似度
EmbShareSim	分享新闻和候选新闻的相似度
EmbWishlistSim	收藏的新闻和候选新闻的相似度
EmbLastLongClickSim	最后一次点击新闻和候选新闻的相似度

模型训练完成后，观察特征的权重是否符合预期，之后便可进行A/B test实验。

新闻冷启动

由于新闻时效性的特殊性，每分钟都有大量新闻进入新闻池，我们加快embedding的更新频率，每小时更新一次，但是提高更新频率并不能解决新闻冷启动问题。最新的新闻没有用户的行为，无法准确的学习到向量。针对上述问题，有以下两种解决方案：对新闻的标签（分类，主题，关键词

等）做embedding；选择新闻内容相似度最近的k篇新闻，用k篇新闻的向量平均值作为该篇新闻的向量。

总结

本文阐述了word2vec的基本原理，调节参数的方法，利用word2vec获取房源、用户类型和房源类型的向量表示，在新闻推荐场景的召回和排序阶段的应用。重点介绍了训练模型中可以优化的点，包括语料的构建，目标函数的改进，可调节的参数，参数对效果的影响等。对文中的技术细节感兴趣的同学，欢迎讨论和指导。