浅谈微视推荐系统中的特征工程

原创 hannahguo 云加社区 2019-12-03

导语 | 在推荐系统中，特征工程扮演着重要的角色。俗话说数据和特征决定了机器学习算法的上限，而模型、算法的选择和优化只是在不断逼近这个上限。特征工程的前提是收集足够多的数据，使用数据学习知识，从大量的原始数据中提取关键信息并表示为模型所需要的形式。本文主要说明微视，这种富媒体形态的短视频平台，是如何通过视频内容特征以及用户属性和行为数据，来精准预测用户对短视频的喜好的。（本文作者：hannahguo，编辑：尾尾）

引言

本文主要是跟各位读者分享特征工程领域的一些通用方法和技巧，以及微视在特征工程上的相关实践经验。微视作为一个短视频平台，存在其独有的业务特点，这也给特征构造带来了一定的难度。比如热目类目在播放时长、互动率等指标上表现出天然的优势，长视频相比于时长较短的视频在播放完成度、完播率等指标上存在明显的劣势，如何消除这些bias的影响都是特征构造时需要特别注意的地方，而对于我们线上的多目标排序模型来说，不同单目标对应的最优特征组合也不尽相同，这些不仅需要较强的专业领域知识，更重要的是对自身业务场景的深刻认知以及大量线上实验的探索尝试与验证。

一、特征提取

微视作为一个短视频平台，存在其独有的业务特点，这也给特征构造带来了一定的难度。比如热目类目在播放时长、互动率等指标上表现出天然的优势，长视频相比于时长较短的视频在播放完成度、完播率等指标上存在明显的劣势，如何消除这些bias的影响都是特征构造时需要特别注意的地方，而对于我们线上的多目标排序模型来说，不同单目标对应的最优特征组合也不尽相同，这些不仅需要较强的专业领域知识，更重要的是对自身业务场景的深刻认知以及大量线上实验的探索尝试与验证。

特征工程就是将原始数据空间映射到新的特征向量空间，使得在新的特征空间中，模型能够更好地学习数据中的规律。因此，特征提取就是对原始数据进行处理与变换的过程。常见的原始数据类型有数值型、离散型，还有文本、图像、视频等。如果将这些数据作为一个整体来看待的话，把用户、视频、作者看作节点，用户与视频、作者的交互看作边构建出的复杂网络也是我们的原始数据。

事实上，如果特征工程做的足够好，即使是简单的模型，也能表现出非常好的效果。而复杂的模型可以在一定程度上减少特征工程的工作量。例如，对于线性模型，我们需要将类别变量进行独热编码等处理，但对于复杂一些的模型如树模型，则可以直接处理类别变量。像推荐系统中常用的LR模型，需要手工构造组合特征，而FM模型可以解决特征组合的问题，直接输入原始特征。而更复杂的DNN模型，可以自动学习特征的表示。

在微视场景下，视频的播放时长、播放完整度、点赞、转发、分享、评论等多种互动行为都是推荐模型的训练目标，根据模型所要学习的目标和业务逻辑，我们需要考虑数据中有哪些可能相关的信息，从现有数据中挖掘出对模型预测有用的特征。比如在微视排序中，用户的兴趣，在App上的播放、互动等行为以及视频的类别、标签、热度等都是强相关的因素。在确定了哪些因素可能与预测目标相关后，我们需要将此信息抽取成特征，下面会对不同特征的处理方式做具体介绍。

1.数值特征

数值类型的数据具有实际统计意义，例如用户对不同类目下视频的兴趣分，或者是计数，例如一个视频被播放了多少次、被点赞、转发以及评论了多少次等。虽然一些机器学习模型可以直接输入数值类型的数据，但是通常情况下，对数值型数据进行适当的变换和处理能带来更优的效果。对于数值特征，我们主要考虑的因素是它的大小和分布。下面介绍几种我们用到的数值特征的处理方法。

1）分桶。比如视频一周内被播放次数应该是一个有用的特征，因为播放次数跟视频的热度有很强的相关性，但是如果不同视频的播放次数跨越不同的数量级，则很难发挥想要的作用。例如LR模型，模型往往只对比较大的特征值敏感。对于这种情况，通常的解决方法是进行分桶。分桶操作可以看作是对数值变量的离散化，之后通过二值化进行 one-hot 编码。

分桶的数量和宽度可以根据业务领域的经验来指定，也有一些常规做法。(1)等距分桶，每个桶的值域是固定的，这种方式适用于样本分布较为均匀的情况；(2)等频分桶，即使得每个桶里数据一样多，这种方式可以保证每个桶有相同的样本数，但也会出现特征值差异非常大的样本被放在一个桶中的情况；(3)模型分桶，使用模型找到最佳分桶，例如利用聚类的方式将特征分成多个类别，或者树模型，这种非线性模型天生具有对连续型特征切分的能力，利用特征分割点进行离散化。

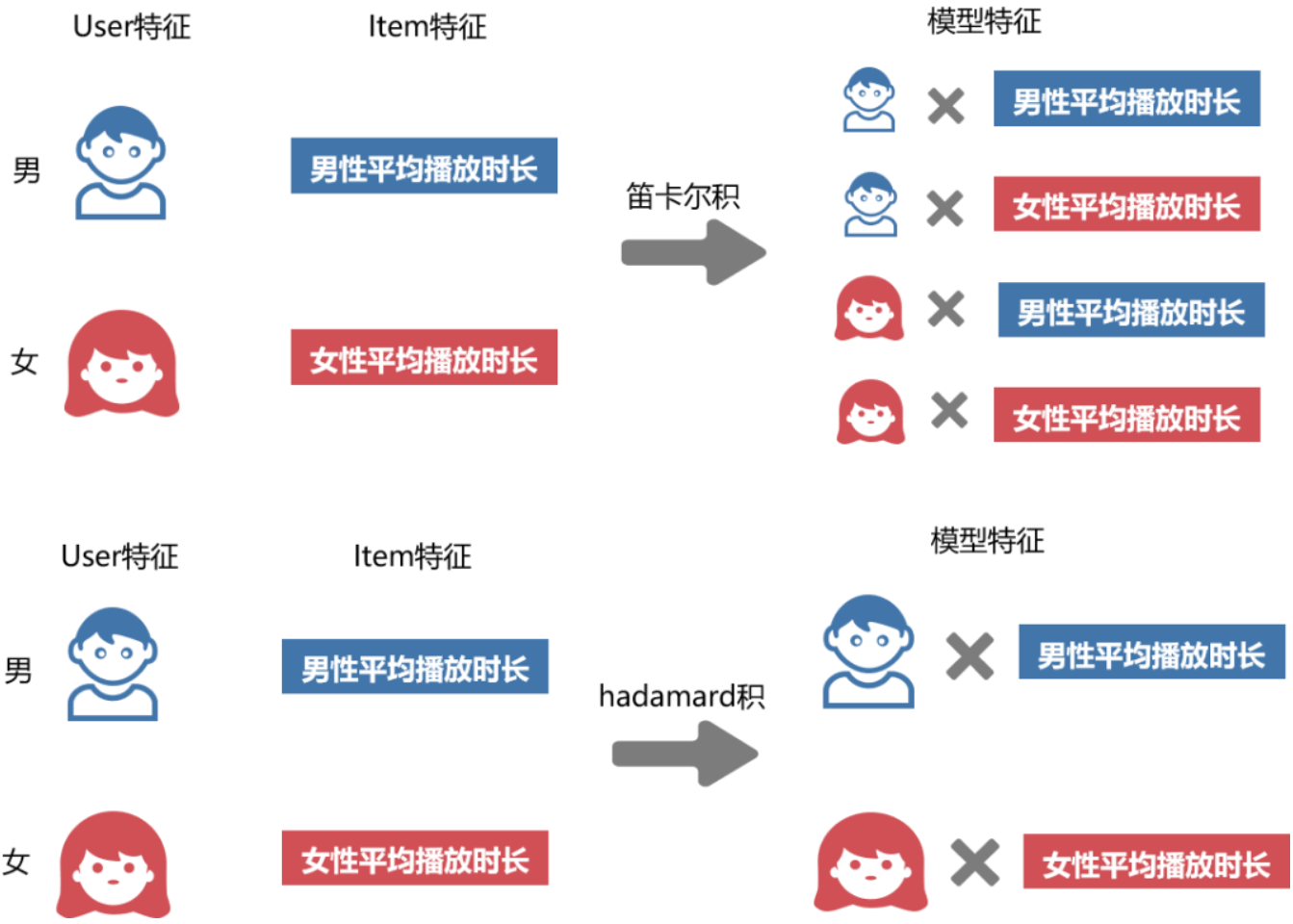
分桶是离散化的常用方法，将连续特征离散化为一系列 0/1 的离散特征，离散化之后得到的稀疏向量，内积乘法运算速度更快，计算结果方便存储。离散化之后的特征对于异常数据也具有很强的鲁棒性。需要注意的是：

- 要使得桶内的属性取值变化对样本标签的影响基本在一个不大的范围，即不能出现单个分桶的内部，样本标签输出变化很大的情况；
- 使每个桶内都有足够的样本，如果桶内样本太少，则随机性太大，不具有统计意义上的说服力；
- 每个桶内的样本尽量分布均匀。

2）截断。对于连续型数值特征，有时精度太高可能只是噪声，并不具备太多的信息，也使得特征维度急剧上升。因此可以保留一定的精度，之后当作类别特征进行处理。对于长尾的数据，可以先进行对数缩放，再进行精度截断，之后可以当做类别变量做二值化处理，这也是我们实际应用中的做法。

3）缺失值处理。实际问题中经常会遇到特征缺失的情形，对于特征缺失，可以选择补一个值，例如使用均值，中位数，众数等进行替代，需要视具体情况进行选择；也可直接忽略，即将缺失作为一种信息进行编码输入模型让其进行学习，比如用户性别缺失，可以直接将未知作为一种类别进行处理；还可以使用模型预测缺失值，当然也有一些模型可以直接处理缺失值特征，比如XGBoost。

4） 特征交叉。特征交叉可以表示特征之间的相互作用，有助于表示非线性关系，增强对问题的刻画，缺点是维度快速增长，需要更多的训练样本。提升模型表达能力常见的关联方式有内积、笛卡尔积、哈达玛积等。内积是将两个或多个特征相乘，若向量a=(a1,b1)，向量b=(a2,b2)，则向量a·向量b=a1a2+b1b2；笛卡积是将所有元素两两相乘，例如，A={a,b}，B={0,1,2}，则 A×B={(a, 0), (a, 1), (a, 2), (b, 0), (b, 1), (b, 2)} ；Hadamard积是将对应位置的元素相乘，例如，A={a,b}，B={0,1}，则 A×B={(a, 0) (b, 1) } 。下图旨在说明笛卡尔积与Hadamard积在具体特征构造上的差异。



特征交叉可以通过一些特征选择方法来选择有效的组合特征，比如卡方检验、特征相关性分析等；也可以根据经验进行组合， 有些cross特征，虽然可能没有直观的解释，但也常常会给模型带来很大的效果提升。除了手工构造交叉特征外，有些模型可以自动进行特征的交叉组合，比如常用的FM和FFM模型等。

5）标准化与缩放。数据的标准化或者归一化是将数据按比例缩放，将其转化为无量纲的纯数值，使得不同单位或量级的特征之间具有可比性，对于利用梯度下降来训练模型参数的算法，有助于提升模型的收敛速度。需要强调的是，不同业务的数据，其数据分布是不同的，缩放的方法一定要符合其特定的数据分布。一般会根据实际数据的情况，对常规做法或者公式进行调整，但大体思路上还是一致的，下面介绍3类相对通用的做法。

- 0-1标准化，是对原始数据的线性变换，使结果落到[0,1]区间，其中max为样本数据的最大值，min为样本数据的最小值。
  这种方法有一个缺陷就是当有新数据加入时，可能会导致max值和min值的变化，需要重新定义。
  如果max值和min值波动较大，容易使得标准化结果不稳定，因此在实际使用中常用经验常量值来替代最大最小值。

$$x^* = \frac{x - min}{max - min}$$

- z-score 标准化，经过处理的数据符合标准正态分布，即均值为0，标准差为1，其中μ为所有样本数据的均值，σ为所有样本数据的标准差。
  这种标准化方式要求原始数据的分布可以近似为高斯分布，否则效果会变得很糟糕。

$$x^* = \frac{x - \mu}{\sigma}$$

- 非线性标准化，这种方法一般使用在数值差异较大的场景，通过一些数学函数，比如对数、指数、正切等，将原始值做映射变换。
  实际使用中，需要根据不同业务的数据分布来选择，比如对数缩放，对数缩放对于处理长尾分布且取值为正数的数值变量非常有效，可以压缩数据范围，将长尾变为短尾，像log2和log10转换在微视中也都有使用，是一种方差稳定的变换。

6）数据平滑。常用的行为次数与曝光次数比值类的特征，由于数据的稀疏性，这种计算方式得到的统计量通常具有较大的偏差，需要做平滑处理，比如广告点击率常用的贝叶斯平滑技术。而在我们推荐场景中，也会用到很多统计类特征、比率特征。如果直接使用，比如由于不同item的下发量是不同的，这会让推荐偏向热门的类目，使得越推越窄，无法发现用户的个体差异，也不利于多样性的探索。我们可以把曝光量进行分段，同一个曝光量级的指标进行比较，也可以用该item所属类目统计量的平均值进行平滑处理。对于离群值较多的数据，我们会使用更加健壮的处理方法，比如使用中位数而不是均值，基于分位数而不是方差。而在短视频业务上较短或较长的视频在播放完成度上存在天然的差距，我们按视频本身长度离散，观看时长做分位数处理，同时做威尔逊置信区间平滑，使得各视频时长段播放完成度相对可比，避免出现打分因视频长度严重倾斜的情况。以及短视频app的投稿数量大，对于长尾的视频和类目都是需要做平滑处理的。下面介绍两种较为常用的平滑技术。

- **贝叶斯平滑**

电商领域中经常需要计算或预测一些转化率指标，比如CTR。这些转化率可以是模型的预测值，也可以作为模型的特征使用。以商品点击率预测为例，CTR的值等于点击量除以曝光量。理想情况下，例如某个广告点击量是10000次，转化量是100次，那转化率就是1%。但有时，例如某个广告点击量是2次，转化量是1次，这样算来转化率为50%。但此时这个指标在数学上是无效的。因为大数定律告诉我们，在试验不变的条件下，重复试验多次，随机事件的频率近似于它的概率。后者点击量只有2次，不满足“重复试验多次”的条件。如果对于一个新上线的商品，其曝光为0，点击量也为0，此时这件商品的CTR应该设为0还是赋一个初始值？初始值设0是可以的，但不太合理。当CTR作为特征使用时，表示这个商品完全没有点击，不太符合日常推断，通常是赋一个大于0的初始值。

以上两个问题可以使用平滑技术来解决。贝叶斯平滑的思想是给CTR预设一个经验初始值，再通过当前的点击量和曝光量来修正这个初始值。如果某商品的点击量和曝光量都是0，那么该商品的CTR就是这个经验初始值；如果商品A和商品B的曝光量差别很大，那么可以通过这个经验初始值来修正。贝叶斯平滑就是确定这个经验值的过程。贝叶斯平滑是基于贝叶斯统计推断的，因此经验值计算的过程依赖于数据的分布情况。对于一件商品或一条广告，对于某次曝光，用户要么点击，要么没点击，这符合二项分布。因此对于点击率类的贝叶斯平滑，都可以基于以下假设：对于某件商品或广告，其是否被点击是一个伯努利分布。伯努利分布的共轭分布就是Beta分布，也就是说，点击率服从Beta分布。而所有的数据有一个自身的点击率分布，这个分布可以用不同的beta分布来拟合。beta分布可以看做是对点击率的一个先验知识，我们可以根据观测来修改我们的先验，所以贝叶斯平滑就是估计Beta分布中的参数α和β，其中C和I是点击次数和曝光量。实际应用时根据历史数据得到的α和β可以帮助确定平滑参数的大致范围，防止设置参数时偏离过大。

$$r = \frac{C + \alpha}{I + \alpha + \beta}$$

- **威尔逊区间平滑**

在现实生活中我们会接触到很多评分系统，如豆瓣书评、YouTube 影评，在这些评分中有1个共同问题是每个 item 的评分人数是不同的，比如10000 个人打了 90 分似乎比只有 10 个人打了 90分更能被相信该 item 是90分的。威尔逊区间法常用来解决此类问题，是一种基于二项分布的计算方法，综合考虑评论数与好评率，平滑样本量对评价的影响，我们画像兴趣分上也用到了威尔逊区间平滑。

假设u表示正例数（好评），n表示实例总数（评论总数），那么好评率p就等于u/n。p越大，表示这个item的好评比例越高，越应该排在前面。但是，p的可信性，取决于有多少人，如果样本太小，p就不可信。我们已知p是二项分布中某个事件的发生概率，因此我们可以计算出p的置信区间。置信区间实际就是进行可信度的修正，弥补样本量过小的影响。如果样本多，就说明比较可信，不需要很大的修正，所以置信区间会比较窄，下限值会比较大；如果样本少，就说明不一定可信，必须进行较大的修正，所以置信区间会比较宽，下限值会比较小。威尔逊区间就是一个很好的修正公式，在小样本上也具有很强的鲁棒性。

在下面的公式中， p表示样本的好评率，n表示样本的大小，z表示对应某个置信水平的z统计量，是一个常数。一般情况下，在95%的置信水平下，z统计量的值为1.96。可以看到，当n的值足够大时，这个下限值会趋向 p 。如果n非常小，这个下限值会远小于 p，起到了降低好评率的作用，使得该item的打分变低、排名下降。

$$\frac{p + \frac{1}{2n}z^2_{1-\frac{\alpha}{2}} \pm z_{1-\frac{\alpha}{2}}\sqrt{\frac{p(1-p)}{n} + \frac{z^2_{1-\frac{\alpha}{2}}}{4n^2}}}{1 + \frac{1}{n}z^2_{1-\frac{\alpha}{2}}}$$

7）bias消除。微视会用到一些不同时间窗口以及实时的统计特征，比如不同类目或者不同时长区间下的完播率、平均播放时长等，考虑到冷热门类目以及长短视频在统计量上本身存在明显的差异，平滑之后我们会用统计量均值进行消偏，这也相当于有一个对热门视频降权，对长视频提权的作用。

8）多维度定义。有些特征可以结合多个属性或者统计量来定义，比如用户所属用户群特征，用户群可以从画像、操作系统等维度来定义；比如用户的活跃度特征可以从周/月登录次数、日播放时长、日播放个数、平均完播率等维度联合定义。

9）根据目标定制。有效的特征应该是与模型训练目标、样本定义紧密相关的，需要从数据中发掘与模型目标强相关的因素。比如视频的播放时长、播放完整度是我们的直接学习目标，同时考虑到用户在短视频app上的互动类型比较多，包括点赞、评论、分享、关注等，而不同的行为对于用户有着不同的价值，点赞表示兴趣偏好，评论反映用户的真实感受，分享利于传播分发，关注建立了一种好友兴趣关系，我们会同时利用这些用户互动行为进行建模。与广告点击率预估中的ctr特征类似，我们可以根据不同目标正负样本的定义，分别为每个单目标模型构造正样本率特征。以及比如从评论文本中分析用户对视频的"内容、bgm、速度、清晰度"等属性的情感倾向，对评论文本进行情感分析，让文本评论好评率来指导模型根据真实评价推荐视频就显得很有意义。

2.类别特征

类别特征可以是标签、属性、类型 ，比如在微视应用中，视频的id、作者、类别、标签、清晰度、质量、topic、bgm曲风与速度等视频属性特征。同时也可以将数值特征离散化，从定量数据中获得定性数据。下面介绍几种我们对类别变量的处理方法。

1）独热编码。独热编码通常用于处理类别间不具有大小关系的特征，每个特征取值对应一维特征，能够处理缺失值，在一定程度上也起到了扩充特征的作用。但是当类别的数量很多时，特征空间会变得非常大。在这种情况下，一般可以用PCA等方法进行降维。

2）散列编码。对于有些取值特别多的类别特征，使用独热编码得到的特征矩阵非常稀疏，再加上如果还有笛卡尔积等构造的组合特征，会使得特征维度爆炸式增长。特征数量多的问题自古有之，目前也已经有很多用于降维的方法。比如聚类、PCA等都是常用的降维方法。但这类方法在特征量和样本量很多的时候本身就计算量很大，所以对大问题也基本无能为力。特征哈希就是一种简单的降维方法，在微视使用也较多，特征哈希法的目标就是是把原始的高维特征向量压缩成较低维特征向量，且尽量不损失原始特征的表达能力，其优势在于实现简单，所需额外计算量小；降低特征维度，从而加速算法训练与预测的时间，以及降低内存消耗；但代价是通过哈希转换后学习到的模型变得很难检验，我们很难对训练出的模型参数做出合理解释。特征哈希法的另一个问题是它会把多个原始特征哈希到相同的位置上，出现哈希collision现象，但实际实验表明这种collision对算法的精度影响很小。

3）打分排名编码。比如在对用户画像的使用上，我们既直接使用了连续值画像特征，同时考虑到画像兴趣分之间天然的序列关系，分别对用户一、二级类目的top1~top10兴趣进行one-hot，利用打分的排名对类别特征进行编码，强化兴趣打分排名的信号，且这种方式对于异常点较为不敏感；还有比如用户活跃度较高的前topn类目和不活跃度较高的前topn类目也可按次序排名作为离散特征。

4）异常值处理。实际应用中，我们常常关心的一个点是异常值对整个统计以及离散化的影响，对于数值变量我们可以把绝对值处理为一个相对值，作为一个次序变量进行one-hot编码，或者做分箱离散化，都可以较好的减轻离群点的影响。对于分类变量，可以考虑神经网络中常见的做法，将分类变量做特征嵌入。比如说这里有1万个不同的标签，把它投影到64维或者128维的vector上面，相当于原来一个1万维的one-hot特征，现在只需要64维或128维就可以进行表示，将原本十分稀疏的向量空间转换到语义更为丰富且低维的特征embedding空间，且内存需求更少，精度也会更高。

5）类别特征之间交叉组合。比如用户性别、操作系统cross得到用户分群特征，视频类别与标签之间的组合特征。在实际应用中，类别特征之间的组合方式千变万化，这类特征一般从业务逻辑的角度出发进行构造。相比类别特征之间的笛卡尔积操作，基于分组统计的特征组合方式计算更加复杂，需要对业务数据有较好的理解。

6）类别特征和数值特征之间交叉组合。这类特征通常是在类别特征某个具体类别中计算一些统计量。例如用户对不同类目视频的完播率、平均播放时长，不同用户群体的互动指标等利用数值特征对类别特征进行处理。

例如我们构造了视频一二级类目的曝光次数cross快划次数的组合特征，这个可以有很直观的解释，比如分析某个用户的样本发现类似王者荣耀_31_31的组合，即我们的推荐系统给这个用户曝光了31个王者荣耀的视频，但是每个都快速划过了，如果还是继续推，这个用户体验将会是极度糟糕的，而这个特征会很大程度的解决这个问题，使得尽可能的给这个用户曝光感兴趣的类目。这个特征可以进一步细化为类目曝光次数cross一些统计量，比如完播次数、互动次数等，因为没有快划可能是用户愿意尝试去看，完播可能是喜欢这个视频内容，而互动比如说转发或者点赞，可以看做是用户表现出了更为强烈的兴趣。

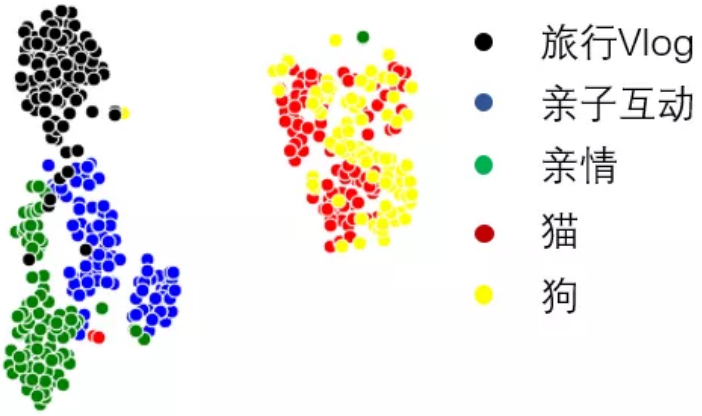
比如为了提升视频的清晰度，我们给视频清晰等级打标签，但是发现直接加入清晰度属性特征后，清晰视频分布和线上指标并没有改善。通过对终端屏幕匹配情况，视频相比于屏幕过度拉伸比例的分析，我们发现约有十分之一的播放体验表现出过度拉伸，对于部分大屏幕用户，过度拉伸比例会更高，于是我们进一步尝试了视频清晰度分别与用户手机屏幕拉伸度和手机型号的笛卡尔积特征，清晰视频播放占比有了较为明显的提升，这对用户体验起到了一定的优化作用。

3.Embedding特征

1）视频embedding

视频embedding分为基于内容的embedding和基于行为的embedding，前者使用视频的标题、封面、图像，音频等视频自身属性信息，通过nlp、图像视觉等技术获得embedding，后者是基于用户与视频的交互行为数据获得，比如我们发现用户在一个session中，前后点击的视频存在一定的相似性，通常会表现出对某类型视频的兴趣偏好，可能是同个风格类别，或者是相似的话题人物等。因此我们将一段时间内用户点击的视频id序列作为训练数据，使用skip-gram模型学习视频的embedding特征。由于用户点击行为具有上下文承接关系，因此得到的embedding特征有很好的聚类效果，使得在特征空间中，同类目的视频聚集在一起，相似类目的视频在空间中距离相近。在微视推荐系统里，视频embedding不仅可以作为排序特征，利用用户最近点击过视频的平均embedding进行召回，也是带来了效果提升。

我们使用TSNE对视频embedding进行降维，从可视化结果来看，同一个类目下的视频是聚在一起的；相似的类目在特征空间中离得较近，如“猫”和“狗”，“亲子互动”和“亲情”；差异较大的类目离得较远，如“旅行Vlog”和“猫”。这还是很符合预期的，一个用户的主要兴趣可能就集中在某几类，比如有的用户喜欢“猫”，那这个用户很大可能对“猫”的视频有大量的播放以及互动行为，那我们学习出来关于猫这个类目的视频embedding就会很相似，表现出在嵌入空间中聚集在一起的情形。



但是如果只是简单的对视频id序列学习 embedding特征，我们是无法获得新视频embedding的。针对这个问题，我们使用了side information来解决视频冷启动问题，side information指的是视频的一、二级类目、视频标签、视频时长、清晰度、距离发布的时间等视频属性信息，像距离发布的时间属性对于新闻资讯类视频，提升视频时新性有一定的帮助，尤其我们发现用户比较偏爱新发布的视频。我们将视频embedding特征转化为视频属性的embedding特征，取一段时间内同属性视频的平均embedding作为这个属性的embedding特征。这样当有新的视频进入到推荐库时，可以计算出新视频的视频属性embedding。这样做的好处是在同一个语义空间做运算，排序模型不需要再重新学习embedding的空间分布。

基于side information获得的视频embedding区分能力还是受到一定限制的，只要视频属性相同，不同视频embedding是完全一样的，而且如果增加了新的类目标签或者其他属性分类也是没有办法处理的。针对以上情况，实际使用中我们采用增量式skip-gram模型学习视频的embedding，使用推荐库最新资源线上实时训练，将新入库的视频加入到模型中做增量式学习。Incremental skip-gram模型与传统skip-gram模型的不同之处在于embedding空间分布是动态更新的。

2）user embedding

想让embedding表达什么，主要在于选择哪一种方式构建语料，不同的用户行为序列，表达的兴趣也不同，比如快速划过行为、完播行为，点赞转发行为等表达的兴趣以及程度也都是不同的。因此视频embedding向量最终的作用，是不同item在用户兴趣空间中的位置表达。目前使用较多的主要是基于word2vec以及相应衍生的embedding技术和基于图神经网络的embedding技术，像我们在隐式画像上就使用了基于异构图的user embedding。

我们也可以把推荐问题建模成一个大规模的多分类问题，使用softmax loss学习一个DNN模型，即在某一时刻某一上下文信息下为用户在视频推荐库中精准地预测出下一次播放视频的类别，最后把训练好的DNN模型最后一层隐层输出作为user embedding。深度学习模型虽然能够减少一部分特征工程的工作，但有些原始数据是不能直接输入到DNN中，与CV、NLP 不同的是，推荐系统对特征工程格外依赖，好的特征能够起到非常关键的作用。我们的输入融合多种信息，主要包括人口统计学信息，播放历史，搜索历史，上下文信息，兴趣画像等，同时使用全场景数据而不是只使用用户播放数据。同时考虑了一些泛化能力比较强的数值和时间特征，比如完播该类目的视频数，最近一次播放该类目视频距离现在的时间等刻画用户与视频类别关系的特征。除了这些偏正向的特征，用户对于一些类目曝光但不点击快速划过等负反馈的信号同样非常重要。

简单一点的做法也可以将一段时间内用户点击过的视频的平均embedding作为该用户的embedding特征，当然这里的“平均”可以是简单的算术平均，也可以是根据视频的热度和时间属性等进行加权平均或者尝试用 RNN 替换掉平均操作。同时将时间跨度取长一点，可以表达用户的长期兴趣；取短一点，可以用于刻画用户的短期兴趣，当然用户覆盖率也会随之降低。比如用户最近一周内主要点击观看的都是关于“猫”的视频，那该用户embedding特征，就会跟“猫”的向量很相近。我们也尝试将用户点赞或者分享转发过的视频序列作为训练数据，为用户互动过的视频提权。这里需要注意的是，有时单个用户行为序列太稀疏了，无法直接训练，一般可以先对用户做聚类再训练。

3）作者embedding

可以取作者近一个月内发布视频的平均embedding，作为该作者的embedding特征。这样做的出发点是，如果两个作者发布的视频相似，那么这两个作者的embedding向量也应该是相近的。假设此时某个用户喜欢作者A，那么我们可以试着把与作者A相似的作者B发布的视频推荐给该用户。

4.context特征

context特征通常是客户端带的信息，在用户授权的前提下可以直接获取，比如请求时间、用户手机品牌、手机型号、操作系统、当前网络状态（3g/4g/wifi）、用户渠道等实时属性特征以及之间的cross特征。

5.session特征

session特征一般是基于用户最近的行为流水，常见的session划分方法有以下几种：

- 固定行为数窗口，例如最近100条行为中分视频类别的完播个数、快速划过个数；
- 固定时间窗口，例如最近3天里有过正向行为的item id或者一些统计量；
- 连续行为窗口，例如用户1次打开app到关闭app期间的播放互动行为。

如上几种session定义的方法没有优劣之分，一般会结合具体业务场景做混合定义。在获取到用户的session数据后，可以直接将session里对应的item id序列作为特征，或者是session内的类别统计数据，也可以将预训练好的item embedding构造成session特征。

二、特征选择

特征选择是指选择相关特征子集的过程，好的特征选择能够提升模型的性能，更能帮助我们理解数据的特点、底层结构，这对进一步改善模型、算法都有着重要作用。特征选择主要有以下两个目的：

- 简化模型，节省存储和计算开销；
- 减少特征数量、降维，改善通用性、降低过拟合的风险。

下面介绍几种特征选择的常用方法。

1.过滤式（Filtering）

过滤式特征选择独立于学习算法，不需要依赖任何模型，直接由数据集求得，评估依赖于数据集本身。一般主要考虑特征变量和目标变量之间的相关性以及特征变量之间的相互关系，一般认为相关

度大的特征或者特征子集会对后续学习算法带来较高的准确率。这类方法在预处理时也使用较多，优点是计算效率高、复杂度低，独立于算法，但也可能选出冗余的特征。

- 覆盖率。
  首先计算每个特征的覆盖率，覆盖率很小的特征对模型的预测效果作用不大，可以剔除。
- 方差分析。
  分析特征的数据分布，比如说某个特征方差接近于0，说明不同样本在这个特征上基本没有什么差异，可以说这个特征是对于样本区分基本没有太大作用的无关变量。
  因此通常可以选择方差大于某个阈值的特征，去掉取值变化小的特征。
- Pearson相关系数。
  皮尔森相关系数是一种简单的，能帮助理解特征和目标变量之间关系的方法，用于衡量变量之间的线性相关性，取值区间为[-1，1]，-1表示完全的负相关，+1表示完全的正相关，0表示没有线性相关。
  通过分析特征与目标之间的相关性，优先选择与目标相关性高的特征。

- 假设检验。
  假设特征变量和目标变量之间相互独立，选择适当检验方法计算统计量，然后根据统计量做出统计推断。
  例如对于特征变量为类别变量而目标变量为连续数值变量的情况，可以使用方差分析，对于特征变量和目标变量都为连续数值变量的情况，可以使用皮尔森卡方检验。
  卡方统计量取值越大，特征相关性越高。

- 互信息。
  在概率论和信息论中，互信息用来度量两个变量之间的相关性。
  互信息越大则表明两个变量相关性越高，互信息为0时，两个变量相互独立。

2.封装式（Wrapping）

与过滤方法不同，封装式特征选择直接使用机器学习算法评估特征子集的效果，直接面向算法优化，效果好，缺点是需要对每一组特征子集训练一个模型，计算复杂度高。常用的特征子集搜索算法有：完全搜索；基于贪心的启发式搜索（前向/后向搜索等）；随机搜索（模拟退火、遗传算法等）。

3.嵌入式（Embedding）

过滤式方法与模型算法相互独立，不需要交叉验证，计算效率比较高，但是没有考虑具体模型算法的特点。封装式方法使用模型来评估特征子集的质量，需要多次训练模型，计算效率很低。嵌入式

方法将特征选择本身作为组成部分嵌入到学习算法里，速度快，效果好，不足是与算法绑定，需要知识调整结构和参数配置。

- 基于正则化

使用带正则惩罚项的模型，比如L1正则化，通过对回归系数添加L1惩罚项来防止过拟合，因产生稀疏解，天然具有特征选择的作用。

- 基于树模型

基于决策树的算法，如随机森林、GBDT，xgboost，在每次选择分类节点时，都会选择最佳分类特征来进行切分，重要的特征更有可能出现在分裂较早的节点，作为分裂节点的次数也越多。因此，可以基于树模型中特征出现次数等指标对特征重要性进行排序。

三、特征重要性分析

特征重要性分析是用来判断哪些变量对模型预测的影响力最大，可以帮助我们理解数据，指导模型参数的设置以及特征的选择，使模型具有良好的可解释性。

- 单特征auc。

对每个单特征训练模型，计算每个特征的auc或gauc，并对其进行排名，精度越高表示该特征重要程度越高。这个方法需要训练多个模型，效率较低。实际应用中，可以有选择的对某些特征子集进行实验。

- 特征值置换。

（1） 特征值置为0。在预测时可以依次将某个特征取值置为0，即不考虑该特征对模型的影响，计算模型auc，精度降低越多，表示这个特征对于模型预测越重要；

（2）特征取随机值。将某个特征取随机值，可以使用均匀或者高斯分布随机抽取值，然后计算模型的准确率。对于某个特征，如果用一个随机值替代后表现比之前差很多，说明该特征很重要；

（3）特征值随机打乱。随机打乱验证集中某一特征变量的值，使用训好的模型进行预测，精度损失越多，说明该特征对于预测结果的影响越大，可以按照精度损失的多少对特征重要性进行排序。这个方法比随机取值更科学，保证了与原始特征分布是一致的。举个例子说明一下，我们有一个已经训练好的模型以及评估该模型表现的评价指标（如RMSE）。假设这是一个短视频播放时长的预测模型，本来在验证集上的RMSE是200，然后我们把某一个变量的值（如性别）全部打乱重新做

预测，比如说此时RMSE变成了500，那么性别这个变量的重要性就可以记为300，即让loss增加了300。

年龄	性别	…	top1类别
21	男	…	游戏
32	女	…	萌宠
…	…	…	…
19	女	…	搞笑
41	男	…	时尚

四、结语

在实际的工程应用中，具体采用什么特征处理方式不仅依赖于业务和数据本身，还依赖于所选取的模型，因此首先要理解数据和业务逻辑以及模型的特点，才能更好地进行特征工程。通常可以考虑对样本做归一化、离散化、平滑处理以及特征变换与特征交叉，我们也会使用一些跨场景的特征迁移方法，复用现有知识域数据，比如手Q场景下的特征数据，具体方法的选择依赖于数据、资源等实际因素，复杂的模型虽然可以减轻我们在特征工程上的工作，但这并不代表我们不需要特征工程了。我们只是把更多的精力放在了模型难以直接从原始数据中学习到，需要借助对业务的理解与外部知识的特征构造上。

特征工程不仅与模型算法相关，与实际问题更是强相关的。针对不同场景，特征工程所用的方法可能相差较大，所以很难总结出适用于不同业务的一套比较通用的方法。尽管如此，但仍然有很多特征工程的方法和技巧在不同问题中都适用。本文把微视排序中一些特征工程的实践经验跟各位读者分享，希望对大家能有所帮助。

---------下方更多精彩----------