抖音短视频推荐大赛小规模赛道Top8模型代码开源

专知 专知 2019-05-30

【导读】字节跳动（今日头条）公司的很多产品都火遍全球。最为典型的可能就是抖音（TikTok）了。今年年初的时候，字节跳动公司，借助ICME2019，召开了《短视频内容理解与推荐》竞赛。竞赛主要分两个赛道：大规模数据集和小规模数据集。小编在Github上发现了参加该比赛，并且获得Rank 第8名的ZooKeeper小队的代码。对短视频理解和推荐感兴趣的同学们，不要错过呦。

> 比赛名称：短视频内容理解与推荐
> 主办单位：字节跳动
> 比赛地址：
> https://biendata.com/competition/icmechallenge2019/
> Zookeeper小队解决方案：
> https://github.com/JiDong-CS/icme2019-bytedance-grand-challenge

【官方比赛介绍】

近年来，机器学习在图像识别、语音识别等领域取得了重大进步，但在视频内容理解领域仍有许多问题需要探索。字节跳动公司旗下的TikTok（抖音海外版）短视频APP在全球范围内的用户中获得非常多的好评，短视频的内容理解与推荐技术成为了我们关注的焦点。

一图胜千言，仅一张图片就包含大量信息，难以用几个词来描述，更何况是短视频这种富媒体形态。面对短视频内容理解的难题，字节跳动作为一家拥有海量短视频素材和上亿级用户行为数据的公司，通过视频内容特征和用户行为数据，可以有充足的数据来预测用户对短视频的喜好。

本次竞赛提供多模态的短视频内容特征，包括视觉特征、文本特征和音频特征，同时提供了脱敏后的用户点击、喜爱、关注等交互行为数据。参赛者需要通过一个视频及用户交互行为数据集对用户兴趣进行建模，然后预测该用户在另一视频数据集上的点击行为。

【竞赛任务】

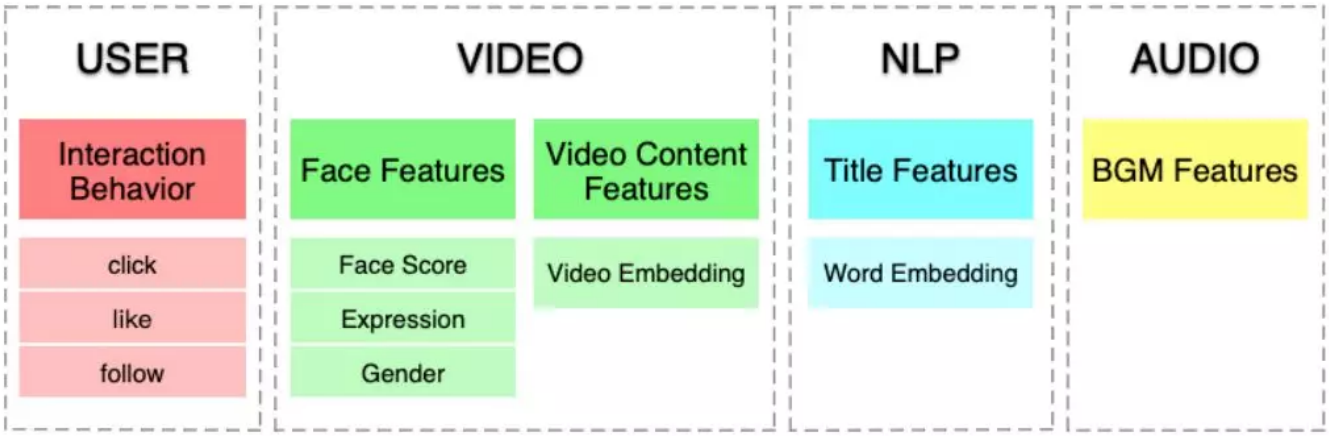
通过构建深度学习模型，预测测试数据中每个用户id在对应作品id上是否浏览完作品和是否对作品点赞的概率加权结果。 本次比赛使用 AUC（ROC曲线下面积）作为评估指标。AUC 越高，代表结果越优，排名越靠前。

- **赛道1：**大规模数据集，亿级别的数据信息。
- **赛道2：**小规模数据集，千万级别的数据信息。

【数据集】

Byte-Recommend100M交互数据集，包含数万用户的亿级别交互数据。

Byte-Recommend100M中的多模态功能包括面部特征，视频内容特征，标题特征和BGM特征，这些特征都是嵌入向量的形式。 参与者能够将它们结合起来以获得更好的推荐。



【数据字段】

	A	B	C	D
1	字段	字段描述	数据类型	备注
2	uid	用户id	int	已脱敏
3	user_city	用户所在城市	int	已脱敏
4	item_id	作品id	int	已脱敏
5	author_id	作者id	int	已脱敏
6	item_city	作品城市	int	已脱敏
7	channel	观看到该作品的来源	int	已脱敏
8	finish	是否浏览完作品	bool	
9	like	是否对作品点赞	bool	
10	music_id	音乐id	int	已脱敏
11	device	设备id	int	已脱敏
12	time	track1：作品观看的起始时间 track2：作品发布时间	int	已脱敏
13	duration_time	作品时长	int	单位：秒

【Zookeeper解决方案】

方法一：LGB-based method

该方法基于 LGB 模型，具体特征工程和模型描述如下。

特征工程

- 基础特征：原始特征
- 统计特征：我们用的都是常规操作，如 count、ratio、nunique 和 ctr 相关特征。count：一维+二 维 count 计数特征 # 对交叉特征求 count
- ratio：类别偏好的 ratio 比例特征
- nunique：类别变量的 nunique 特征

- face 相关的特征：图像的位置（width, height, x, y），beauty 的统计特征(max, avg),男性数量，女性 数量，是否有男性或者女性，face 的数量等 ['face_nums','x','y', 'width', 'height', 'size', 'male_cnt', 'female_cnt', 'avg_beauty','max_beauty','author_male_cnt','author_female_cnt', 'uid_female_ratio']
- title 相关的特征：title 中不同词的数量(unique)以及 title 的长度
- 在该条样本时间前，针对 uid，authorid，musicid 等 组合的正负样本数量统计特征

模型

- 最终使用了 基础特征，count 特征，ratio 特征，face 特征，title 特征，正负样本数量统计特征
- 针对 finish 和 like 采用上述的同一套特征，使用 lgb 模型，对两个任务分别预测

方法二：XDeepFM-based methods

该方法基于 XDeepFM 模型，基于不同的特征输入，训练了两个 XDeepFM 模型，该方法主要考虑了行 为特征和受众特征，它们起到了协同过滤作用。具体特征工程和模型如下所述。

特征工程

- 基本特征： uid，user_city，item_id，item_city，author_id，channel，device_id，music_id；
- 行为特征：（训练集+测试集中）浏览过的视频、音乐、作者、城市列表，计算 TF 值（取前 500 维)；
- 受众特征：（训练集+测试集中）视频、音乐、作者的用户 uid 列表，计算 TF-IDF 值（取前 400 维）；
- 标题特征：计算 TF-IDF 值；
- 脸部特征：{“num_face”: “人脸数目”, “female_ratio”: “女性比例”, “max_beauty”: “beauty 最大值”, “min_beauty”: “beauty 最小值”, “avg_beauty”: “beauty 平均值”, “max_area”: “最大人脸面积”, “avg_area”: “平均人脸面积”}；
- 时间特征：通过时间戳获取年、月、日、时、分，以及工作日特征，月-日交叉表示节日特征；
- video 嵌入：128 维原始特征；
- audio 嵌入：128 维原始特征； 9) count 特征: 计算单个类别特征和多个类别特征共现的次数。

模型训练过程

- 构建统计特征：用户行为特征、物品受众特征
- 构建标题特征
- 构建时间特征

- 调用 DataParser.py 生成特征文件：对 track2 数据进行分块，并行构造特征，生成 tf_record 记录
- 调用 Main.py 进行训练

具体运行命令，请参见模型目录下 build_features.sh 和 run_model.sh

方法三： 模型融合

Track2 线上最优结果是通过模型融合获得的，融合方式是根据经验启发式的设计各模型结果权重，具体计算公式如下：

- finish = (0.5 * xdeepfm1_finish + 0.5 * xdeepfm2_finish) * 0.7 + 0.3 * lgb_finsh
- like = 0.4 * xdeepfm1_like + 0.6 * lgb_like

根据上述方式融合之后，track2 线上 private 最终得分为 0.799658049326414。

> 具体代码，和模型，请访问Github呦。
> Github地址：
> https://github.com/JiDong-CS/icme2019-bytedance-grand-challenge

-END-