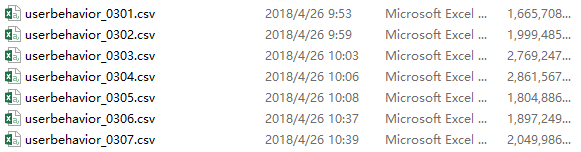
**20180301-20180307数据清洗规则**

**数据文件：**



* **原始数据导入：**

**数据表格：**20180301、20180302、……、20180307

**sql语句：**

LOAD DATA LOCAL INFILE '/home/xuyue/userbehavior/data/userbehavior\_0301.csv'-- 数据位置

INTO TABLE `20180301`-- 要导入的表

FIELDS TERMINATED BY '^'-- 数据以什么分隔

LINES TERMINATED BY '\n'-- 行间用什么分隔

(keystr,userlocation,provinceid,cityid,createTime,codestr,videoName,videoSource,firstLevel,firstLevelCount,secondLevel,secondLevelCount,threeLevel,threeLevelCount,openPostCount,videoStartTime,videoEndTime,videoErrorTime,watchTime,searchValue,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength,dt,ht);-- 表中的字段

……

* **提取code为AAAA009(视频播放结束)和AAAA017(节目基本信息)的数据：**

**数据表格：**20180301\_009, 20180301\_017、……、20180307\_009, 20180307\_017

**sql语句：**

INSERT INTO `20180301\_009` SELECT \* FROM `20180301` WHERE codestr = 'AAAA009';

INSERT INTO `20180301\_017` SELECT \* FROM `20180301` WHERE codestr = 'AAAA017';

……

* **删除AAAA009记录中videoType !=2的数据：**

**sql语句：**DELETE FROM 20180301\_009 WHERE videoType <> 2;

* **处理AAAA017事件对饮的数据：**

**1）将videoName=''的数据过滤：**

DELETE FROM 20180301\_017 WHERE videoName = '';

DELETE FROM 20180302\_017 WHERE videoName = '';

……

**2）将videoTimeLength=''、videoTime>2880的数据过滤：**

DELETE FROM 20180301\_017 WHERE videoTimeLength = '';

DELETE FROM 20180301\_017 WHERE videoTimeLength+0 > 2880;

DELETE FROM 20180302\_017 WHERE videoTimeLength = '';

DELETE FROM 20180302\_017 WHERE videoTimeLength+0 > 2880;

……

* **提取节目信息字段（createTime, videoName, contentSource, directName, actorName, videoType, videoPlot, videoScore, videoRegion, videoTimeLength）：**

**数据表格：**20180301\_video、……、20180307\_video

**sql语句：**

INSERT INTO 20180301\_video( createTime, videoName, contentSource, directName, actorName, videoType, videoPlot, videoScore, videoRegion ,videoTimeLength ) SELECT createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength FROM 20180301\_017 ORDER BY videoName,createTime+0 DESC;

……

* **删除每天的重复节目信息（保留createTime最新的）：**

DELETE FROM 20180301\_video WHERE id NOT IN (SELECT minid FROM (SELECT MIN(id) AS minid FROM 20180301\_video GROUP BY videoName) AS temp);

DELETE FROM 20180302\_video WHERE id NOT IN (SELECT minid FROM (SELECT MIN(id) AS minid FROM 20180302\_video GROUP BY videoName) AS temp);

……

* **合并7天的节目信息并删除重复的节目信息（保留createTime最新的）：**

-- 将7天节目信息数据合并到一张表里

INSERT INTO 0107\_video\_temp(createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength) SELECT createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength FROM 20180301\_video;

INSERT INTO 0107\_video\_temp(createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength) SELECT createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength FROM 20180302\_video;

……

-- 将7天中的节目-- 按videoName、createTime desc排序，然后删除重复数据

INSERT INTO 0107\_video\_temp1(createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength) SELECT createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength FROM 0107\_video\_temp ORDER BY videoName,createTime+0 DESC;

DELETE FROM 0107\_video\_temp1 WHERE id NOT IN (SELECT minid FROM (SELECT MIN(id) AS minid FROM 0107\_video\_temp1 GROUP BY videoName) AS temp);

INSERT INTO 0107\_video(createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength) SELECT createTime,videoName,contentSource,directName,actorName,videoType,videoPlot,videoScore,videoRegion,videoTimeLength FROM 0107\_video\_temp1 ORDER BY videoName;

……

* **将AAAA009记录中没有节目信息的数据删除：**

-- 给0107\_video的videoName添加索引

ALTER TABLE 0107\_video ADD INDEX (videoName);

DELETE FROM 20180301\_009 WHERE videoName NOT IN (SELECT videoName FROM 0107\_video);

DELETE FROM 20180302\_009 WHERE videoName NOT IN (SELECT videoName FROM 0107\_video);

……

* **根据videoName拼接009和017的数据：**

-- 先将009数据拷贝到新表中，防止数据拼接过程出现问题，在新表中进行操作（挑选出watchTime>0的数据）

INSERT INTO 20180301\_merge SELECT \* FROM 20180301\_009 WHERE watchTime>0;

INSERT INTO 20180302\_merge SELECT \* FROM 20180302\_009 WHERE watchTime>0;

……

-- 合并009事件与017事件的数据

UPDATE 20180301\_merge,0107\_video SET

20180301\_merge.contentSource=0107\_video.contentSource,20180301\_merge.directName=0107\_video.directName,

20180301\_merge.actorName=0107\_video.actorName,20180301\_merge.videoPlot=0107\_video.videoPlot,

20180301\_merge.videoScore=0107\_video.videoScore,20180301\_merge.videoRegion=0107\_video.videoRegion,

20180301\_merge.videoTimeLength=0107\_video.videoTimeLength WHERE 20180301\_merge.videoName = 0107\_video.videoName;

UPDATE 20180302\_merge,0107\_video SET

20180302\_merge.contentSource=0107\_video.contentSource,20180302\_merge.directName=0107\_video.directName,

20180302\_merge.actorName=0107\_video.actorName,20180302\_merge.videoPlot=0107\_video.videoPlot,

20180302\_merge.videoScore=0107\_video.videoScore,20180302\_merge.videoRegion=0107\_video.videoRegion,

20180302\_merge.videoTimeLength=0107\_video.videoTimeLength WHERE 20180302\_merge.videoName = 0107\_video.videoName;

……

* **二进制评分模型数据处理**

**数据表说明：**

1）观看记录：t3\_watch

2）节目表：t3\_video

3）用户表：t3\_user

4）评分表：t3\_score\_1

**清洗操作：**

**1、观看记录t3\_watch**

**1）合并所有3.1-3.7merge表，存为新表t3\_watch**

CREATE TABLE t3\_watch LIKE 20180301\_merge;

INSERT INTO t3\_watch SELECT \* FROM 20180301\_merge;

INSERT INTO t3\_watch SELECT \* FROM 20180302\_merge;

……

INSERT INTO t3\_watch SELECT \* FROM 20180307\_merge;

**2）keystr、videoName添加索引**

ALTER table t3\_watch ADD INDEX idxkeystr(keystr);

ALTER table t3\_watch ADD INDEX idxvideoName(videoName);

**3）从t3\_watch表中，剔除videoName或keystr为空的记录**

DELETE FROM t3\_watch WHERE keystr is NULL OR videoName is NULL;

**4）从t3\_watch表中，消除videoName字段的前后空格**

UPDATE t3\_watch SET videoName=TRIM(videoName)；

**5）给t3\_watch表添加主键eventid，标识一个观看事件**

**2、用户表：t3\_user：**

**1）从t3\_watch中提取所有不重复的keystr，存为用户表t3\_user**

create table t3\_user as (SELECT DISTINCT keystr FROM t3\_watch)；

**2）给t3\_user的keystr列添加索引**

ALTER table t3\_user ADD INDEX idxkeystr(keystr);

**3、节目表：t3\_video**

**1）t3\_video拷贝自0107\_video，索引videoName**

**2）从t3\_video表中，消除videoName字段的前后空格**

UPDATE t3\_video SET videoName=TRIM(videoName)；

**3）新添栏位videoid，int自增，从107086+1开始自增，作为主键，删除id列**

alter table t3\_video AUTO\_INCREMENT=107087;

**4、评分表t3\_score\_1**

**1）抽取t3\_watch中的不重复的keystr, videoName字段。与eventid字段一同添加到t3\_score\_1**

INSERT INTO t3\_score\_1(eventid,keystr,videoName) select eventid, keystr, videoName from t3\_watch group by keystr,videoName ORDER BY eventid；

**2）获取t3\_score\_1的videoid,userid,dt,ht数据**

UPDATE t3\_score\_1,t3\_video SET t3\_score\_1.videoid = t3\_video.videoid WHERE t3\_score\_1.videoName = t3\_video.videoName;

UPDATE t3\_score\_1,t3\_user SET t3\_score\_1.userid = t3\_user.userid WHERE t3\_score\_1.keystr = t3\_user.keystr;

UPDATE t3\_score\_1,t3\_watch SET t3\_score\_1.ht = t3\_watch.ht WHERE t3\_score\_1.eventid = t3\_watch.eventid;

UPDATE t3\_score\_1,t3\_watch SET t3\_score\_1.dt = t3\_watch.dt WHERE t3\_score\_1.eventid = t3\_watch.eventid;

**3）t3\_score\_1的score字段设置为1**

UPDATE t3\_score\_1 SET t3\_score\_1.score=1;

**4、导出数据**

SELECT userid,videoid,score FROM t3\_score\_1 ORDER BY userid,videoid INTO OUTFILE '/home/xuyue/dataInit.txt'

**模型数据说明：**

目前采用0、1打分制，认为用户对某个节目发生过行为打分即为1，否则为0。处理好的数据是所有打分为1的记录，包含3列：userID、videoID、score(均为1)，userID的范围：1~107086，videoID的范围：107087~ 125258。实验数据涵盖107086个用户设备，18172个电视节目。

训练集和测试集的划分：按用户分组，在每个用户的观看记录里按照7:3的比例随机划分，参考程序：DataDivideByUserGroup.java

