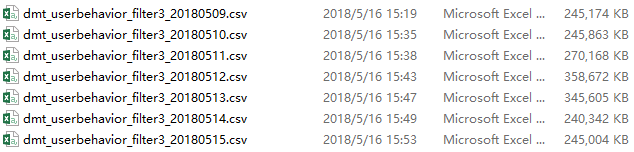
**20180509-20180515数据清洗规则**

**数据文件：**



**备注：**此次提供的原始数据是根据videoName把AAAA009事件、AAAA017事件拼接好的点播数据。

* **原始数据导入：**

**数据表格：**20180509、20180510、……、20180515

**sql语句：**

LOAD DATA LOCAL INFILE '/home/xuyue/userbehavior/data/ dmt\_userbehavior\_filter3\_20180509.csv '-- 数据位置

INTO TABLE `20180509`-- 要导入的表

FIELDS TERMINATED BY ','-- 数据以什么分隔

OPTIONALLY ENCLOSED BY '"'-- 无视以双引号包裹的字段里面的逗号

LINES TERMINATED BY '\r\n'-- 行间用什么分隔

IGNORE 1 LINES -- 忽略表头

……

LOAD DATA LOCAL INFILE '/home/xuyue/userbehavior/data/ dmt\_userbehavior\_filter3\_20180515.csv '-- 数据位置

INTO TABLE `20180515`-- 要导入的表

FIELDS TERMINATED BY ','-- 数据以什么分隔

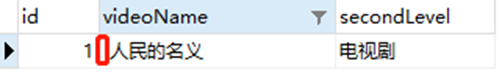
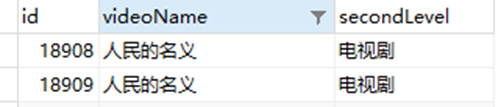
OPTIONALLY ENCLOSED BY '"'-- 无视以双引号包裹的字段里面的逗号

LINES TERMINATED BY '\r\n'-- 行间用什么分隔

IGNORE 1 LINES -- 忽略表头

* **去除videoName首尾空格：**

**节目名称字符串首尾可能存在空格，为了防止同一个节目被当做两个节目来处理，需要去除videoName首尾空格**

** **

**数据表格：**20180509、20180510、……、20180515

**sql语句：**

UPDATE `20180509` SET videoName = TRIM(videoName);

……

UPDATE `20180515` SET videoName = TRIM(videoName);

* **根据已有的节目类型信息填充videoName相同的数据的类型信息：**

**1）secondLevel字段记录了节目类型信息，首先查看一下7天的观看的节目类型都有哪些种类**

SELECT DISTINCT(secondLevel) from `20180509` ORDER BY secondLevel;

……

SELECT DISTINCT(secondLevel) from `20180515` ORDER BY secondLevel;

7天的观看类型种类数一样，如下：查询到20种结果，红框圈出的认为是无效类型



**2）在填充之前将“更多”、“最新”这两种无效分类设置为空**

UPDATE `20180509` SET secondLevel='' WHERE secondLevel='更多' OR secondLevel='最新';

……

UPDATE `20180515` SET secondLevel='' WHERE secondLevel='更多' OR secondLevel='最新';

**3）挑选出所有secondLevel字段不为空的记录，插入新表(videoName, secondLevel)**

INSERT INTO 20180509\_secondLevel(videoName,secondLevel) SELECT videoName,secondLevel FROM `20180509` WHERE secondLevel != '' ORDER BY videoName;

……

INSERT INTO 20180515\_secondLevel(videoName,secondLevel) SELECT videoName,secondLevel FROM `20180515` WHERE secondLevel != '' ORDER BY videoName;

**4）将7天的secondLevel字段信息合并**

INSERT INTO temp1(videoName, secondLevel) SELECT videoName,secondLevel FROM 20180509\_secondLevel;

……

INSERT INTO temp1(videoName, secondLevel) SELECT videoName,secondLevel FROM 20180515\_secondLevel;

**备注：**在检查secondLevel这个字段时，我们发现同一个节目会对应几种节目类型，如下所示：



对于同一个节目有不同类别信息这种情况，我们选取出现频次最大的类别作为最终的节目类别

**5）统计每个videoName,secondLevel分组出现的频次**

INSERT INTO temp2 SELECT videoName,secondLevel,count(\*) FROM temp1 GROUP BY videoName,secondLevel ORDER BY videoName;

**6）在分组内按照频次大小降序排列（temp3添加一个自动递增的id列）**

INSERT INTO temp3(videoName,secondLevel,count) SELECT \* FROM temp2 ORDER BY videoName, count DESC;

**7）选取频次最大的即id最小的，其他删除**

DELETE FROM temp3 WHERE id NOT IN (SELECT minid FROM (SELECT MIN(id) AS minid FROM temp3 GROUP BY videoName) AS temp);

**8）将处理好的节目类型信息放到一张新表里**

INSERT INTO 0915\_secondLevel SELECT videoName,secondLevel FROM temp3;

**9）更新原数据表中的secondLevel字段**

UPDATE `20180509` a, 0915\_secondLevel b SET a.secondLevel=b.secondLevel WHERE a.videoName= b.videoName;

……

UPDATE `20180515` a, 0915\_secondLevel b SET a.secondLevel=b.secondLevel WHERE a.videoName= b.videoName;

填充后的效果如下图所示：



对于一些有多种类别信息的节目，它每种类别信息出现次数相同目，无法保证处理之后节目类别信息的准确性，如下图所示，对于我们肉眼观察的同一种类型的节目可能会出现各种各样的类型：



* **数据清洗：**

**1）核查keystr、videoName是否有空值 (经检查，没有空值)**

SELECT \* FROM `20180509` WHERE keystr='';

SELECT \* FROM `20180509` WHERE videoName='';

SELECT \* FROM `20180510` WHERE keystr='';

SELECT \* FROM `20180510` WHERE videoName='';

……

SELECT \* FROM `20180515` WHERE keystr='';

SELECT \* FROM `20180515` WHERE videoName='';

**备注：**根据以下语句可以得到contentSource的取值有：''、未来电视、芒果TV、电视看点4种取值，contentSource=''时，没有节目详情

SELECT DISTINCT(contentSource) from `20180509`;

SELECT DISTINCT(contentSource) from `20180510`;

……

SELECT DISTINCT(contentSource) from `20180515`;

**2）删除没有节目信息的记录**

INSERT INTO 20180509\_filter SELECT \* FROM `20180509` WHERE contentSource<>'';

INSERT INTO 20180510\_filter SELECT \* FROM `20180510` WHERE contentSource<>'';

……

INSERT INTO 20180515\_filter SELECT \* FROM `20180515` WHERE contentSource<>'';

**3）7天数据汇总**

INSERT INTO totalrecord SELECT \* FROM 20180509\_filter;

INSERT INTO totalrecord SELECT \* FROM 20180510\_filter;

……

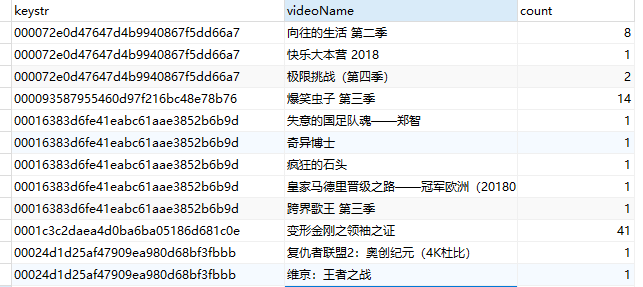
INSERT INTO totalrecord SELECT \* FROM 20180515\_filter;

**备注：**由于目前打分都是直接以二进制的形式，并没有用到watchTime和videoTimeLength，所以此次的数据并没有根据这两个字段来过滤数据

**4）统计每个用户观看同一个节目的次数（这个次数信息目前由于节目类别缺失问题，暂时未应用到算法中）**

INSERT INTO totalwatchcounts SELECT keystr,videoName,COUNT(\*) FROM totalrecord GROUP BY keystr,videoName ORDER BY keystr,videoName;

得到的结果大致如下：



* **二进制评分模型数据处理**

**数据表说明：**

1）观看记录：totalrecord

2）节目表：t1\_user

3）用户表：t1\_video

4）评分表：t1\_score\_1

**清洗操作：**

**1）构建用户表t1\_user：**

**从totalrecord表中提取所有不重复的keystr，为其设置编号（从1开始）**

CREATE TABLE `t1\_user` (

`userID` int(2) NOT NULL AUTO\_INCREMENT,

`keystr` varchar(255) NOT NULL,

PRIMARY KEY (`userID`),

KEY `index\_keystr` (`keystr`)

);

INSERT INTO t1\_user(keystr) SELECT DISTINCT(keystr) FROM totalrecord;

**2）构建节目表t1\_video：**

**从totalrecord表中提取所有不重复的videoName，为其设置编号（接着用户编号开始）**

CREATE TABLE `t1\_video` (

`videoID` int(2) NOT NULL AUTO\_INCREMENT,

`videoName` varchar(255) NOT NULL,

PRIMARY KEY (`videoID`),

KEY `index\_videoName` (`videoName`)

);

ALTER TABLE t1\_video AUTO\_INCREMENT=110805;-- 用户有110804个

INSERT INTO t1\_video(videoName) SELECT DISTINCT(videoName) FROM totalrecord;

**3）构建评分表t1\_score\_1：有观看记录的节目即认为打分为1**

CREATE TABLE `t1\_score\_1` (

`userID` int(2) DEFAULT NULL,

`keystr` varchar(255) DEFAULT NULL,

`videoID` int(2) DEFAULT NULL,

`videoName` varchar(255) DEFAULT NULL,

`score` tinyint(1) NOT NULL DEFAULT '1'

);

INSERT INTO t1\_score\_1(keystr,videoName) SELECT keystr,videoName FROM totalrecord GROUP BY keystr,videoName; -- 共900249条记录

UPDATE t1\_score\_1,t1\_user SET t1\_score\_1.userID=t1\_user.userID WHERE t1\_score\_1.keystr=t1\_user.keystr;

UPDATE t1\_score\_1,t1\_video SET t1\_score\_1.videoID=t1\_video.videoID WHERE t1\_score\_1.videoName=t1\_video.videoName;

**4）导出数据**

SELECT userID,videoID,score FROM t1\_score\_1 ORDER BY userID,videoID INTO OUTFILE '/home/xuyue/dataInit.txt''

输出文件：



**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：1~110804，videoID的范围：110805~ 123971。实验数据涵盖110804个用户设备，13167个电视节目。

目前只是第一版清洗，还没有用到该数据集，所以暂时未进行数据集的划分操作。