实验手册——

搜狗搜索日志分析系统

目 录

一、数据预处理（Linux 环境）.....3
1. 查看数据.....3
2. 数据扩展.....3
3. 数据过滤.....3
二、基于 Hive 构建日志数据的数据仓库.....4
1. 基本操作.....4
2. 创建分区表（按照年、月、天、小时分区）.....5
3. 查询结果.....6
四、实现数据分析需求一：条数统计.....6
五、实现数据分析需求二：关键词分析.....7
1. 查询关键词长度统计.....7
2. 查询频度排名（频度最高的前 50 词）.....7
六、实现数据分析需求三：UID 分析.....7
1. UID 的查询次数分布（查询 1 次的 UID 个数，...查询 N 次的 UID 个数）.....7
2.UID 平均查询次数.....7
3. 查询次数大于 2 次的用户总数.....7
4.查询次数大于 2 次的用户占比.....7
4. 查询次数大于 2 次的数据展示.....8
七、实现数据分析需求四：用户行为分析.....8
1. 点击次数与 Rank 之间的关系分析.....8
2. 直接输入 URL 作为查询词的比例.....8
3. 独立用户行为分析.....9
八、实现数据分析需求五：实时数据.....10
九、使用 Sqoop 将数据导入 MySQL.....10
十、HBase Shell 操作命令实验.....11
十一、使用 Sqoop 将数据导入 HBase.....12
十二、HBase Java API 访问统计数据.....13
1. 操作要求.....13
2. 数据准备.....13
3. 数据导入.....14
十三、Mahout 聚类操作实验.....14
1. 数据描述.....14
2. 准备数据.....15
3. 运行聚类程序.....15

一、数据预处理（Linux 环境）

搜狗数据的数据格式：
访问时间\t 用户 ID\t[查询词]\t 该 URL 在返回结果中的排名\t 用户点击的顺序号\t 用户点击的 URL
其中，用户 ID 是根据用户使用浏览器访问搜索引擎时的 Cookie 信息自动赋值，即同一次使用浏览器输入的不同查询对应同一个用户 ID。

1.查看数据

进入实验数据文件夹

```
[zkpk@master ~]$ cd /home/zkpk/resources/sogou-data/500w
```

less 查看

```
[zkpk@master 500w]$ less /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8
```

解决中文显示乱码问题
本步骤已经完成从 gbk 转化为 utf-8 格式，不需要再操作。见下面目录，该目录是乱码清洗的 Java 代码：

```
[zkpk@master ~]$ cd /home/zkpk/resources/ide-code/workspace/test
```

查看总行数

```
[zkpk@master ~]$ cd /home/zkpk/resources/sogou-data/500w
[zkpk@master 500w]$ wc -l /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8
```

截取部分数据数据

```
[zkpk@master 500w]$ head -100 ~/resources/sogou-data/500w/sogou.500w.utf8 > ~/resources/sogou-data/500w/sogou.demo
```

2.数据扩展

将时间字段拆分并拼接，添加年、月、日、小时字段

```
[zkpk@master ~]$ cd /home/zkpk/resources/ide-code
[zkpk@master ide-code]$ bash sogou-log-extend.sh /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8 /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8.ext
```

3.数据过滤

过滤第 2 个字段（UID）或者第 3 个字段（搜索关键词）为空的行（需要用第 2 步数据扩展的结果）

```
[zkpk@master ~]$ cd /home/zkpk/resources/ide-code
[zkpk@master  ide-code]$  bash  sogou-log-filter.sh  /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8.ext
/home/zkpk/resources/sogou-data/500w/sogou.500w.utf8.flt
```

数据加载到 HDFS 上

```
[zkpk@master ~]$ hdfs dfs -mkdir -p /sogou/20111230
[zkpk@master ~]$ hdfs dfs -put /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8 /sogou/20111230/
[zkpk@master ~]$ hdfs dfs -mkdir -p /sogou_ext/20111230
[zkpk@master ~]$ hdfs dfs -put /home/zkpk/resources/sogou-data/500w/sogou.500w.utf8.flt /sogou_ext/20111230
```

二、基于 Hive 构建日志数据的数据仓库

要求：
Hadoop 集群正常启动

打开 Hive 客户端

```
[zkpk@master ~]$ hdfs dfs -mkdir -p /sogou/20111230
[zkpk@master ~]$ cd /home/zkpk/apache-hive-0.13.1-bin
[zkpk@master apache-hive-0.13.1-bin]$ bin/hive
```

下面操作都是在 Hive 客户端操作。

1.基本操作

查看数据库

```
hive>show databases;
```

创建数据库

```
Hive>create database sogou;
```

使用数据库

```
hive>use sogou;
```

查看所有表名

```
hive>show tables;
```

创建外部表

```
hive>CREATE EXTERNAL TABLE sogou.sogou_20111230(
    > ts STRING,
    > uid STRING,
    > keyword STRING,
    > rank INT,
    > order INT,
    > url STRING)
    > COMMENT 'This is the sogou search data of one day'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
```

```
    > STORED AS TEXTFILE
    > LOCATION '/sogou/20111230';
```

查看新创建表结构

```
hive> show create table sogou.sogou_20111230;
hive> describe sogou.sogou_20111230;
```

删除表

```
hive> drop table sogou.sogou_20111230;
```

2.创建分区表（按照年、月、天、小时分区）

创建扩展 4 个字段（年、月、日、小时）数据的外部表：

```
hive> CREATE EXTERNAL TABLE sogou.sogou_ext_20111230(
    > ts STRING,
    > uid STRING,
    > keyword STRING,
    > rank INT,
    > order INT,
    > url STRING,
    > year INT,
    > month INT,
    > day INT,
    > hour INT
    > )
    > COMMENT 'This is the sogou search data of extend data'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE
    > LOCATION '/sogou_ext/20111230';
```

创建带分区的表：

```
hive> CREATE EXTERNAL TABLE sogou.sogou_partition(
    > ts STRING,
    > uid STRING,
    > keyword STRING,
    > rank INT,
    > order INT,
    > url STRING
    > )
    > COMMENT 'This is the sogou search data by partition'
    > partitioned by (
    > year INT,
```

```
    > month INT,
    > day INT,
    > hour INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE;
```

灌入数据

```
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> INSERT OVERWRITE TABLE sogou.sogou_partition PARTITION(year,month,day,hour) select * from sogou.sogou_ext_20111230;
```

3.查询结果

```
hive> select * from sogou_ext_20111230 limit 10;
hive> select url from sogou_ext_20111230 limit 10;
hive> select * from sogou_ext_20111230 where uid='96994a0480e7e1edcaef67b20d8816b7';
```

四、实现数据分析需求一：条数统计

数据总条数

```
hive> select count(*) from sogou.sogou_ext_20111230;
```

非空查询条数

```
hive> select count(*) from sogou.sogou_ext_20111230 where keyword is not null and keyword !='';
```

无重复总条数（根据 ts、uid、keyword、url）

```
hive> select count(*) from (select * from sogou. sogou_ext_20111230 group by ts,uid,keyword,url having count(*) =1) a;
```

独立 UID 总数

```
hive> select count(distinct(uid)) from sogou.sogou_ext_20111230;
```

五、实现数据分析需求二：关键词分析

1.查询关键词长度统计

```
hive> select avg(a.cnt) from (select size(split(keyword,'\\s+')) as cnt from sogou. sogou_ext_20111230) a;
```

2.查询频度排名（频度最高的前 50 词）

```
hive> select keyword,count(*) as cnt from sogou.sogou_ext_20111230 group by keyword order by cnt desc limit 50;
```

六、实现数据分析需求三：UID 分析

1. UID 的查询次数分布（查询 1 次的 UID 个数，...查询 N 次的 UID 个数）

```
hive> select SUM(IF(uids.cnt=1,1,0)),SUM(IF(uids.cnt=2,1,0)),SUM(IF(uids.cnt=3,1,0)),SUM(IF(uids.cnt>3,1,0))
from
    > (select uid,count(*) as cnt from sogou.sogou_ext_20111230 group by uid) uids;
```

2.UID 平均查询次数

```
hive> select sum(a.cnt)/count(a.uid) from (select uid,count(*) as cnt from sogou.sogou_ext_20111230 group by
uid) a;
```

3. 查询次数大于 2 次的用户总数

```
hive> select count(a.uid) from (
    > select uid,count(*) as cnt from sogou.sogou_ext_20111230 group by uid having cnt > 2) a;
```

4.查询次数大于 2 次的用户占比

A UID 总数：

```
hive> select count(distinct (uid)) from sogou.sogou_ext_20111230;
```

B UID2 次以上的数量：

```
select count(a.uid) from (
    > select uid,count(*) as cnt from sogou.sogou_ext_20111230 group by uid having cnt > 2) a;
```

结果 C=B/A

4. 查询次数大于 2 次的数据展示

```
hive> select b.* from
    > (select uid,count(*) as cnt from sogou.sogou_ext_20111230 group by uid having cnt > 2) a
    > join sogou.sogou_ext_20111230 b on a.uid=b.uid
    > limit 50;
```

七、实现数据分析需求四：用户行为分析

1. 点击次数与 Rank 之间的关系分析

Rank 在 10 以内的点击次数占比

A：

```
hive> select count(*) from sogou.sogou_ext_20111230 where rank < 11;
```

B：

```
hive> select count(*) from sogou.sogou_ext_20111230;
```

占比：A/B

用户只翻看搜索引擎返回结果的前 10 个结果，即返回结果页面的第一页。这个用户行为决定了尽管搜索引擎返回的结果数目十分庞大，但真正可能被绝大部分用户所浏览的，只有排在最前面的很小一部分而已。所以传统的基于整个结果集合查准率和查全率的评价方式不再适用于网络信息检索的评价，我们需要着重强调在评价指标中有关最靠前结果文档与用户查询需求的相关度的部分。

2. 直接输入 URL 作为查询词的比例

（1）直接输入 URL 查询的比例

A：

```
hive> select count(*) from sogou.sogou_ext_20111230 where keyword  like '%www%';
```

B：

```
hive> select count(*) from sogou.sogou_ext_20111230;
```

占比：A/B

（2）直接输入URL的查询中，点击数点击的结果就是用户输入的URL的网址 所占的比例

C：

```
hive> select SUM(IF(instr(url,keyword)>0,1,0)) from
    > (select * from sogou.sogou_ext_20111230 where keyword  like '%www%') a;
```

占比：C/A

从这个比例可以看出，很大一部分用户提交含有URL的查询是由于没有记全网址等原因而想借助搜索引擎来找到自己想浏览的网页。因此搜索引擎在处理这部分查询的时候，一个可能比较理想的方式是首先把相关的完整URL地址返回给用户，这样有较大可能符合用户的查询需求。

3.独立用户行为分析

（1） 查询搜索过”仙剑奇侠传“的 uid，并且次数大于 3

```
hive> select uid,count(*) as cnt from sogou.sogou_ext_20111230  where keyword='仙剑奇侠传' group by uid having
cnt > 3;
```

（2）查找 uid 是 653d48aa356d5111ac0e59f9fe736429 和 e11c6273e337c1d1032229f1b2321a75 的相关搜索记录

```
hive> select * from sogou.sogou_ext_20111230 where uid='653d48aa356d5111ac0e59f9fe736429' and keyword like '%
仙剑奇侠传%';
hive> select * from sogou.sogou_ext_20111230 where uid='e11c6273e337c1d1032229f1b2321a75' and keyword like '%
仙剑奇侠传%';
```

（3）分析打印结果

```
653d48aa356d5111ac0e59f9fe736429    www.163dyy.com    影视  4
e11c6273e337c1d1032229f1b2321a75    baike.baidu.com   信息  20
```

搜索具备多样性，因人而异，主要注意个性化需求

八、实现数据分析需求五：实时数据

每个 UID 在当天的查询点击次数

（1）创建临时表

```
hive> create table sogou.uid_cnt(uid STRING, cnt INT) COMMENT 'This is the sogou search data of one day'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE;
```

（2）查询并插入

```
hive> INSERT OVERWRITE TABLE sogou.uid_cnt select uid,count(*) as cnt from sogou.sogou_ext_20111230 group by uid;
```

九、使用 Sqoop 将数据导入 MySQL

要求：
MySQL 服务启动且运行正常，命令为：

```
[zkpk@master ~]$ /etc/init.d/mysqld status
```

Hadoop 集群启动且运行正常，命令为：

```
[zkpk@master ~]$ jps
```

将前面生成的实时数据从 HDFS 导入到 MySQL 中，步骤如下：
以下操作都是在 MySQL 交互客户端执行。

（1）登录 MySQL
mysql -uhadoop -phadoop

（2）创建数据库
查看 test 数据库是否存在：

```
mysql> show databases;
```

如果不存在就创建：

```
mysql> create database test;
```

（2）创建表

```
mysql> CREATE TABLE `test`.`uid_cnt` (
    ->  `uid` varchar(255) DEFAULT NULL,
    ->  `cnt` int(11) DEFAULT NULL
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

提示：语句中的引号是反引号`，不是单引号’。
创建成功后，退出 MySQL。

（3）导入数据

进入 sqoop 安装主目录：

```
[zkpk@master ~]$ cd /home/zkpk/sqoop-1.4.5.bin__hadoop-2.0.4-alpha
```

导入命令：

```
[zkpk@master sqoop-1.4.5.bin__hadoop-2.0.4-alpha]$ bin/sqoop export --connect jdbc:mysql://192.168.190.147:3306/test --username hadoop --password hadoop --table uid_cnt --export-dir '/user/hive/warehouse/sogou.db/uid_cnt' --fields-terminated-by '\t'
```

注意：红色 IP 部分需要使用 HadoopMaster 节点对应的 IP 地址

代码解释：

```
bin/sqoop export   ##表示数据从 hive 复制到 mysql 中\
--connect jdbc:mysql://192.168.1.113:3306/test \
--username root \
--password admin \
--table bb   ##mysql 中的表，即将被导入的表名称 \
--export-dir '/user/hive/warehouse/sogou.db/uid_cnt'   ##hive 中被导出的文件 \
--fields-terminated-by '\t'    ##hive 中被导出的文件字段的分隔符
```

十、HBase Shell 操作命令实验

要求：
HBase 集群正常启动，且可以运行正常

进入客户端

```
[zkpk@master ~]$ cd /home/zkpk/hbase-0.98.7-hadoop2

[zkpk@master hbase-0.98.7-hadoop2]$ bin/hbase shell
```

查看列表

```
hbase(main):001:0> list
```

创建表 test

```
hbase(main):002:0> create 'test', {NAME => 'f1', VERSIONS => 5}
```

再次查看列表对比

```
hbase(main):003:0> list
```

插入数据

```
hbase(main):001:0> put 'test', 'aid001', 'f1:uid', '001'
```

扫描查询数据

```
hbase(main):001:0> scan 'test'
```

单条查询数据

```
hbase(main):002:0> get 'test','aid001'
```

查看表结构

```
hbase(main):003:0> describe 'test'
```

修改表

```
hbase(main):004:0> disable 'test'
hbase(main):004:0> alter 'test', NAME => 'f1', VERSIONS => 3
hbase(main):004:0> enable 'test'
```

再次查看表结构对比

```
hbase(main):004:0> describe 'test'
```

清空表

```
hbase(main):004:0> truncate 'test'
```

扫描表

```
hbase(main):004:0> scan 'test'
```

删除表

```
hbase(main):004:0> disable 'test'
hbase(main):004:0> drop 'test'
hbase(main):004:0> list
```

创建新表 uid_cnt

```
hbase(main):004:0> create 'uid_cnt', {NAME => 'f1', VERSIONS => 5}
```

十一、使用 Sqoop 将数据导入 HBase

要求：
MySQL 服务启动且运行正常
HBase 集群启动且运行正常

进入 sqoop 安装主目录

```
[zkpk@master ~]$ cd /home/zkpk/sqoop-1.4.5.bin__hadoop-2.0.4-alpha
```

执行导入命令：

```
[zkpk@master sqoop-1.4.5.bin__hadoop-2.0.4-alpha]$ bin/sqoop import --connect jdbc:mysql://192.168.190.147:3306/test --username hadoop --password hadoop --table uid_cnt --hbase-table uid_cnt --column-family f1 --hbase-row-key uid --hbase-create-table -m 1
```

其中红色主机 ip 需要改为本机的 ip 地址
字段解释：

```
bin/sqoop import --connect jdbc:mysql://192.168.190.147:3306/test --username hadoop --password hadoop --table uid_cnt
--hbase-table uid_cnt  HBase 中表名称
--column-family f1  列簇名称
--hbase-row-key uid  HBase 行键
--hbase-create-table  是否在不存在情况下创建表
-m 1 启动 Map 数量
```

十二、HBase Java API 访问统计数据

1.操作要求

Hadoop 集群启动且运行正常
HBase 集群启动且运行正常

2.数据准备

将之前的 uid_cnt 数据从 HDFS 复制到本地

```
[zkpk@master ~]$ cd
[zkpk@master ~]$ hdfs dfs -get /user/hive/warehouse/sogou.db/uid_cnt .
[zkpk@master ~]$ uid_cnt
[zkpk@master uid_cnt]$ cat ~/uid_cnt/00000* > uid_cnt.output
[zkpk@master uid_cnt]$ head uid_cnt.output
```

3.数据导入

使用 Java 程序将数据导入 HBase 中：

```
[zkpk@master ~]$ cd ~/resources/ide-code/
[zkpk@master ide-code]$ hadoop jar hbase-example.jar com.adintellig.teach.HBaseImportTest /home/zkpk/uid_cnt/uid_cnt.output
```

十三、Mahout 聚类操作实验

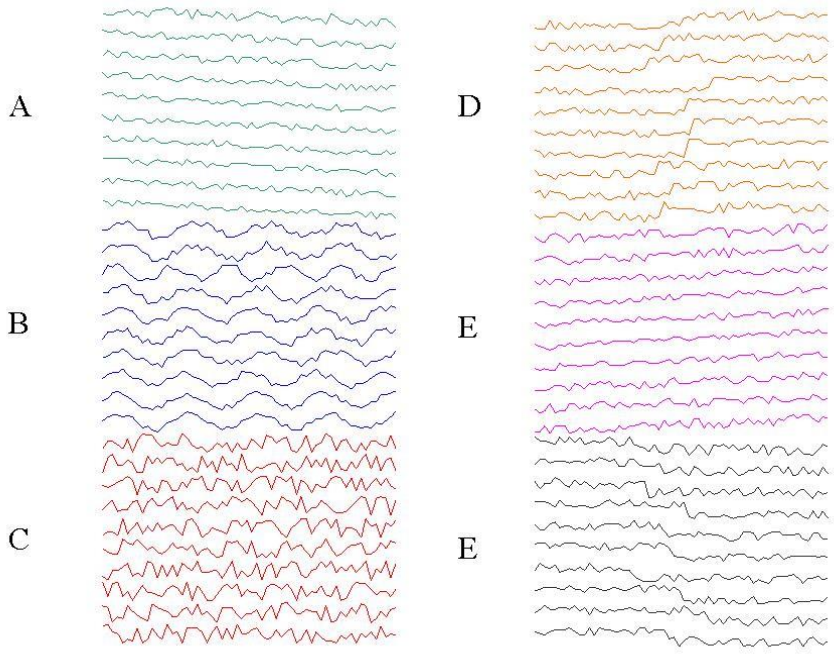
1.数据描述

控制时序数据

共分为六类：

- 趋势向下：Downward Trend (A)
- 周期：Cyclic (B)
- 正常：Normal (C)
- 向上偏移：Upward Shift (D)
- 趋势向上：Upward Trend (E)
- 向下偏移：Downward Shift (F)

下面每一副图都代表了一类数据:



数据文件描述：

- ASCII 编码
- 600 行
- 每行 60 列
- 每行表示一个图表

类别有下面的一些参数定义（600 样本的实际分类）：

- 1-100 正常
- 101-200 周期
- 201-300 趋势向上
- 301-400 趋势向下
- 401-500 向上偏移
- 501-600 向下偏移

2.准备数据

在 HadoopMaster 节点上，打开终端，然后下载控制数据：

```
[zkpk@master ~]$ wget http://archive.ics.uci.edu/ml/databases/synthetic_control/synthetic_control.data
```

如果无法联网下载，则执行下面的命令：

```
[zkpk@master ~]$ cd
[zkpk@master ~]$ cp ~/resources/mahout-data/synthetic_control.data ~/
```

实验数据上传到 HDFS

```
[zkpk@master ~]$ cd
[zkpk@master ~]$ hdfs dfs -mkdir testdata
[zkpk@master ~]$ hdfs dfs -put synthetic_control.data testdata/
[zkpk@master ~]$ hdfs dfs -ls testdata
```

3.运行聚类程序

直接运行下面的命令：

```
[zkpk@master ~]$ hadoop jar /home/zkpk/mahout-distribution-0.9/mahout-examples-0.9-job.jar
org.apache.mahout.clustering.syntheticcontrol.kmeans.Job
```

查看结果文件列表：

```
[zkpk@master ~]$ hdfs dfs -ls output
```

HDFS 上文件列表如下：

```
-rw-r--r--   1 zkpk supergroup        194 2014-07-18 14:23 /user/zkpk/output/_policy
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:23 /user/zkpk/output/clusteredPoints
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:18 /user/zkpk/output/clusters-0
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:18 /user/zkpk/output/clusters-1
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:23 /user/zkpk/output/clusters-10-final
```

```
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:19 /user/zkpk/output/clusters-2
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:19 /user/zkpk/output/clusters-3
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:20 /user/zkpk/output/clusters-4
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:20 /user/zkpk/output/clusters-5
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:21 /user/zkpk/output/clusters-6
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:21 /user/zkpk/output/clusters-7
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:22 /user/zkpk/output/clusters-8
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:22 /user/zkpk/output/clusters-9
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:18 /user/zkpk/output/data
drwxr-xr-x   - zkpk supergroup          0 2014-07-18 14:18 /user/zkpk/output/random-seeds
```

查看文件：

```
[zkpk@master ~]$ /etc/init.d/mysqld status
[zkpk@master ~]$ /home/zkpk/mahout-distribution-0.9/bin/mahout clusterdump -i /user/zkpk/output/clusters-2 -p /user/zkpk/output/clusteredPoints -o result.txt
```

可以看到聚类的所有类别和每行数据所在的类别。