Main

Bo Peng

April 27, 2017

SQL Cloud Database

This chunk contains the SQL commands that cleaned the data, transformed formats and merged the tables by document_id, ad_id and user_id. The final output is the "final" table stored in the online database.

```
--merge tables: all-info of doc
select
documents_categories.document_id as doc_id,
documents_categories.category_id as category_id,
documents_categories.confidence_level as category_confidence,
documents_meta.source_id as source_id,
documents_meta.publish_time as publish_time,
documents_meta.publisher_id as publisher_id,
documents_topics.topic_id as topic_id,
documents_topics.confidence_level as topic_confidence
into doc_info
from documents_categories
join documents_meta on documents_categories.document_id=documents_meta.document_id
join documents_topics on documents_categories.document_id = documents_topics.document_id;


--create doc_category with same doc_id as in doc_info
select * into unique_doc_cat_id
from (
select distinct document_id from documents_categories
intersect
select distinct doc_id from doc_info) as a;

select
a.document_id as doc_id,
a.category_id as cat_id,
a.confidence_level as cat_con
into doc_category
from documents_categories a
inner join unique_doc_cat_id b
on a.document_id  =  b.document_id;


--create doc_topic with same doc_id as in doc_info
select * into unique_doc_topic_id
from (
select distinct document_id from documents_topics
intersect
select distinct doc_id from doc_info) as a;

select
a.document_id as doc_id,
```

```
a.topic_id as topic_id,
a.confidence_level as cat_con
into doc_topic
from documents_topics a
inner join unique_doc_topic_id b
on a.document_id  =  b.document_id;



--create doc_meta with same doc_id as in doc_info
select * into unique_doc_meta_id
from (
select distinct document_id from documents_meta
intersect
select distinct doc_id from doc_info) as a;

select
a.document_id as doc_id,
a.source_id as source_id,
a.publisher_id as publisher,
a.publish_time as time
into doc_meta
from documents_meta a
inner join unique_doc_meta_id b
on a.document_id  =  b.document_id;



--sample 200000 doc_id
select top 200000 * into doc_id_sample
from unique_doc_cat_id;


--sampled doc_category with 200000 sampled doc_id
select
a.doc_id as doc_id,
a.cat_id as cat_id,
a.cat_con as con
into cat_sample
from doc_category a
inner join doc_id_sample b
on a.doc_id  =  b.document_id;


--sampled doc_topic with 200000 sampled doc_id
select
a.doc_id as doc_id,
a.topic_id as topic_id,
a.cat_con as con
into topic_sample
from doc_topic a
inner join doc_id_sample b
on a.doc_id  =  b.document_id;
```

```
--sampled doc_meta with 200000 sampled doc_id
select
a.doc_id as doc_id,
a.source_id as source_id,
a.publisher as publisher,
a.time as publish_time
into meta_sample
from doc_meta a
inner join doc_id_sample b
on a.doc_id  =  b.document_id;


--merged doc with new clusters for categories and topics
select
a.doc_id as doc_id,
a.new_cluster as cat,
b.new_cluster as topic,
c.source_id as source_id,
c.publisher as publisher,
c.publish_time as publish_time
into doc_merge
from cat_clustered a
join topic_clustered b on a.doc_id=b.doc_id
join meta_sample c on a.doc_id=c.doc_id;


--get the final table with all data merged with sampled doc_id
SELECT
clicks_train.display_id,
clicks_train.ad_id,
clicks_train.clicked,
clicks_events.document_id,
clicks_events.geo_location,
clicks_events.platform,
clicks_events.timestamp,
clicks_events.uuid
INTO sum_clicks_events
FROM clicks_train, clicks_events
WHERE clicks_train.display_id = clicks_events.display_id
ORDER BY clicks_train.display_id ASC;


--update the final table with geo_location represents only countries
UPDATE final
SET geo_location = LEFT(geo_location, 2);
```

Fetch the "final" table from AWS RDS and store as data.frame To access the RDS from the R console, in the "ODBC Data Sources" program (pre-installed if you are using a Windows machine), create a data source called "project5243", using type "ODBC Driver for SQL Server", server "project5261.ckquajgj1vtb.us-east-1.rds.amazonaws.com,1433", verification method "SQL Server Verification", user name "bpeng", password "qqqq123456".

Data Preprocessing

Transform timestamp into a AM/PM variable

Keep only the country code component in the geo_locatin variable

Transform country codes into continent codes

Eliminate rows containing NAs and convert into factors

Randomly sample 90% of the data as training set, and the remaining 10% as testing set.

Clustering inactive advertisers with hard threshold

Clean data

Train Logistic Regression Model

```
## Warning: package 'speedglm' was built under R version 3.3.3
## Loading required package: Matrix
## Loading required package: MASS
```

Test over test set

Convert logit results into probability

Evaluation

Rearrange the data by display_id and descending probability of being clicked

Calculate test accuracy for evaluation

```
##    display_id  ad_id clicked       prob Rank     score
## 1         116 292543       0 0.38329575    1 0.0000000
## 2         116  53300       0 0.22379346    2 0.0000000
## 3         116  56754       0 0.22379346    3 0.0000000
## 4         116 332908       1 0.16622944    4 0.2500000
## 5         116 288377       0 0.05410362    5 0.0000000
## 6         116 180923       0 0.04225260    6 0.0000000
## 7         844 107451       1 0.41411731    1 1.0000000
## 8         844 133753       0 0.21055757    2 0.0000000
## 9         844 139563       0 0.19304652    3 0.0000000
## 10        844 116984       0 0.12387687    4 0.0000000
## 11        844  39279       0 0.08590013    5 0.0000000
## 12        844 288388       0 0.05215732    6 0.0000000
## 13       1630  60630       0 0.20922121    1 0.0000000
## 14       1630 103756       1 0.14704922    2 0.5000000
## 15       2128 167205       0 0.42307717    1 0.0000000
## 16       2128  31770       0 0.24969019    2 0.0000000
## 17       2128  56749       1 0.21714415    3 0.3333333
## 18       2128 227442       0 0.16702339    4 0.0000000
## 19       2128 304341       0 0.14849493    5 0.0000000
```

```
## 20       2128 310685       0 0.12167568    6 0.0000000
## 21       2452 289122       1 0.39413516    1 1.0000000
## 22       2452 289915       0 0.39413516    2 0.0000000
## 23       2452 104833       0 0.34731492    3 0.0000000
## 24       2452 132821       0 0.32625882    4 0.0000000
## 25       2452 170148       0 0.32625882    5 0.0000000
## 26       2452 224171       0 0.14576399    6 0.0000000
## 27       2541 170777       0 0.45222806    1 0.0000000
## 28       2541 220703       0 0.38136109    2 0.0000000
## 29       2541 146666       1 0.24698819    3 0.3333333
## 30       2541  43258       0 0.15453146    4 0.0000000
```

```
## 0.6232148
```