Useful SQL and Google Cloud for Facebook Data Wrangling

Keng-Chi Chang

National Taiwan University

2017-10-17

Motivation

- Calculate number of shared users between pages

	Trump	FoxNews	Clinton	NYTimes
Trump	2243216	1078513	32731	25842
FoxNews	1078513	2449174	87084	63401
Clinton	32731	87084	1768980	367021
NYTimes	25842	63401	367021	986613

- For large matrix, can run out of memory even with sparse matrix
- 0.3 billion users $\times$ 2000 pages $\times$ 5% non-zero elements $\times$ 4 Byte $\div$ 1024 (KB) $\div$ 1024 (MB) $\div$ 1024 (GB) $\approx$ 112 GB use of memory
- Strategy: User SQL to group by users first

SQL (Structured Query Language)

- Industry standard for manipulating relational data
- Useful for columnwise calculation
- Many derivatives: MySQL, NoSQL, MongoDB, Google BigQuery
- More functions: See Google BigQuery Reference

Basic Structure (SELECT, FROM, WHERE)

```
SELECT
  data.id AS user_id,
  NTH(2, SPLIT(src, "/")) AS post_id,
FROM
  [ntue-data-sci:US_Election_Dataset_Local.old_reactions_201501_to_201611]
WHERE
  data.type = "LIKE"
```

→ [ntufbdata:USdata.old_1000_user_post_like]

- In Google BigQuery: [ProjectID:DatasetID.TableID]

GROUP BY & Nested Subquery

```
SELECT
  user_id,
  GROUP_CONCAT(page_id) AS like_pages,
  GROUP_CONCAT(STRING(like_time)) AS like_times,
FROM (
  SELECT
    user_id,
    page_id,
    COUNT(*) AS like_time,
  FROM (
    SELECT
      user_id,
      NTH(1, SPLIT(post_id, "_")) AS page_id,
    FROM
      [ntufbdata:USdata.old_1000_user_post_like])
  GROUP BY
    user_id,
    page_id)
GROUP BY
  user_id
```

Time Selection & Merging Data (JOIN)

```
SELECT
  F1.user_id AS user_id,
  F1.post_id AS post_id,
FROM [ntufbdata:USdata.old_1000_user_post_like] AS F1
INNER JOIN [ntufbdata:1000_page_post.201501_to_201611_all] AS F2
ON
  F1.post_id = F2.post_id
WHERE
  DATE(post_created_date_CT) >= DATE("2016-10-01") AND
  DATE(post_created_date_CT) <= DATE("2016-11-07")
```

Workflow for Google BigQuery

1. Use SQL to extract data you want
   - Web UI, R, command line `bq`
2. Export to Google Cloud Storage
   - Provide a URI

```
gs://ntuusfb/us_user_like/us_user_like_1000_page_and_politician_times_\
 201501_to_201611_all/*.csv
```

   - For tables > 1GB, use a folder since it will be split
3. Download to server
   - Web UI, command line `gsutil`

Use `gsutil` to Download Data

- Follow the Quickstart

1. Install Google Cloud SDK
2. First time: Run `gcloud init` to configure and select project
3. Run `gsutil cp gs://[source] [destination]` to download

```
gsutil cp gs://ntuusfb/us_user_like/us_user_like_1000_page_and_politician_\
          times_201501_to_201611_all/*.csv
          ~/usfb/analysis-ideology/temp/us_user_like_1000_page_and_politician_\
          times_201501_to_201611_all/
```

General Advice for BigQuery

- Google BigQuery is pricing by *columns*
- Use subqueries for intermediate tables
  - Will save *a lot* if intermediate tables are large
- Keep a record of everything you run
  - Not only for others
  - But also for your future self
- Come up with a naming scheme *at the start* of your project
  - Project, Dataset, Table, Variable names
  - Data types (use strings for every IDs)