Wikipedia ログの解析

2019 年度春学期"大規模データ処理法"課題

氏名: **Shu Sakamoto**
学籍番号: 71775236
ログイン名: t17523ss

解析の目的

2019 年度春学期「大規模データ処理」の授業では, 2019 年 4 月における Wikipedia のアクセスログを解析した. このとき我々は 2019 年 4 月に人気のあったページを確認するため, 該当期間の総アクセス数の順位に着目した. しかし, この手法を用いると "トップページ" や "特別:検索" のように Wikipedia の性質上アクセスされやすいページが上位に表示されてしまい, ノイズとなってしまう. また, "白石麻衣" のように常に人気のあるページも表示されてしまうが, これは "4 月に人気のあるコンテンツ" としての側面が薄いページであるため, これもなるべく取り除きたい. つまり, 我々 j が着目したいのは任意の記事の 4 月におけるニュース性である. そこで本課題では, 慢性的に人気のあるページではなく, コンテンツのニュース性が高いページが上位に表示されるような解析を行なった.

解析の手順

ニュース性をはかる指標として, $\frac{\Delta A}{\Delta t}$ を用いる (Aは任意の時間窓におけるアクセス数, tは時間). つまり, ある時間においてアクセス数がどれだけ増えたかに着目する. この指標を用いれば "トップページ" や "特別:検索" といったページは上位に表示されず, 例えば "令和" のようなニュース性の高いページのみが上位に表示されるはずである.
そこで, `reducer.py`においてアクセス数を足し上げていくプロセスを排除し, 代わりにアクセス数の変化を記録する過程を付け加えた. 最終的にはアクセス数の変化が正の方向に大きかったのページを出力するようにした (以下参照).

```
#!/usr/bin/env python3.6
import sys

args = sys.argv

t, c = sys.stdin.readline().strip().split("\t")
title, count, maxDiff, prev_c = t, int(c), 0, int(c)
for line in sys.stdin:
    t, c = line.strip().split("\t")
    if title != t:
        if count > int(args[1]):
            print(title, "\t", maxDiff);
        title, count, maxDiff, prev_c = t, int(c), 0, int(c)
    else:
        diff = int(c) - prev_c
        if diff > maxDiff:
            maxDiff = diff
        prev_c = int(c)
```

```
    if count > int(args[1]):
        print(title, "\t", maxDiff);
```

動作確認は

```
gzip -dc ../../../public/pv201904/pageviews-2019040[1-2]-000000.gz | ./mapper.py |
sort | ./reducer.py 1000 | nkf -w --url-input | sort -k 2gr,2
```

で, 実際の処理は

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -
files reducer.py,mapper.py -input pv201904 -output pv201904_diff-3 -mapper
'mapper.py ja' -reducer 'reducer.py 10000'
```

のコマンドで行った.

わかったこと

以下が Wikipedia 各ページの 2019 年 4 月の総アクセス数を降順に 20 位まで並べたものである.

```
メインページ        6505890
特別:検索          2190085
今昔文字鏡          950514
令和        800265
特別:最近の更新             517919
渋沢栄一            455797
Special:Search              435741
特別:ウォッチリスト          339158
飯塚幸三            331295
特別:外部リンク検索          306126
-           303693
元号一覧_(日本)             285944
桜田義孝            248488
ノートルダム大聖堂_(パリ)            202457
特別:ログイン        199339
明仁        192228
万葉集      190165
元号        174828
白石麻衣            162172
皇太子徳仁親王      160523
```

対して, 以下が Wikipedia 各ページの 2019 年 4 月の "ニュース性" を降順に 20 位まで並べたものである.

```
令和        173013
メインページ        98357
```

```
今昔文字鏡          49195
元号一覧_(日本)             37304
元号        21924
宮崎緑      12590
津田梅子            10364
北里柴三郎          9702
Special:Search              8257
浦田直也            8159
特別:外部リンク検索         8148
皇太子徳仁親王      8097
MIX_(漫画)          6730
飯塚幸三            6495
石井浩郎            6141
特別:検索           5510
西部警察            4815
特別:アカウント作成         4604
ストロベリーナイト          4594
道下俊一            4500
```

上記の結果から, $\frac{\Delta A}{\Delta t}$ をニュース性の指標として用いると "-" "特別:*" などのノイズが減ることが分かった. しかし, "メインページ" "特別:アカウント作成" などのページは順位が下がったものの未だ残っている. これは, 特定のニュース (元号発表など) に釣られてどうしてもアクセス数が増えてしまうページであることが予想できる. また, "令和" "元号" "津田梅子" "飯塚幸三" など, ニュース性の高い記事もランクインしていることが見て取れる. 本解析の場合, Δt は 1 時間となっているので, 1 時間で人々の興味が急速に集まるようなニュース (元号発表・新紙幣発表・著名人の逮捕) などが上位にランクインし, じわりじわりと興味があつまるようなページ (ニュースに関連した "万葉集" や海外のニュースである "ノートルダム大聖堂") はランクインできていない. この Δt はニュースの性質 (広まりやすさ?) を反映していると考えられるため, この間隔を 1 日や 1 週間に伸ばしたような解析を追加で行うことで, さらなる検証を行っていきたい.

やってみた感想

動作確認をする際の`sort`コマンドのオプション引数や hdfs システムを実際に手で動かす良い機会になった. また, 全く同じ数値データでも解析の切り口を変えることで様々な断面が見えることが明らかになった. 今後はデータを扱う際, 一つの表面的な解析にこだわらず様々な情報を抽出することに注力したい.