Twitterアカウントの判別

B2 金子侑紀

動機

- フォローが多くて情報が追いにくい
- リストに入れるのを自動でやってほしい

- リストごとにトレンドを取得したい（目標）

手法

- フォロー関係があるかどうかで判別する

- 同じ傾向のツイートがあるかどうかで判別する

手法

- フォロー関係があるかどうかで判別する
  →ネットワーク分析(19章)

- 同じ傾向のツイートがあるかどうかで判別する
  →形態素解析＋クラスタ分析

手法

- フォロー関係があるかどうかで判別する
  →ネットワーク分析(19章)

- 同じ傾向のツイートがあるかどうかで判別する
  →形態素解析＋クラスタ分析

まずはツイートを集める

```python
import tweepy
import settings
import os
import time

api = settings.api

fout = open("output.csv", "w")


for status in tweepy.Cursor(api.home_timeline).items():
    try:
        time      = str(status.created_at)
        user      = str(status.user.screen_name)
        status    = str(status.text).replace("\n","")
        if "RT" in status:  # RTは書き込まない
            pass
        else:
            fout.write(time+","+user+","+status+"\n")
            print(status)   # txtファイルに書き込まれたツイー
    except: # RateLimit対策
        time.sleep(900)
fout.close()
print("Finished!")
```

- タイムラインのツイートを取得
- python + Tweepy

Rで形態素解析

RパッケージRMeCabで用意されている関数

www.ic.daito.ac.jp/~mizutani/mining/rmecab_func.html

Rのパッケージ RMeCabは、指定したファイルを**形態素解析**エンジン MeCabに渡して**形態素解析**を実施し、MeCabから返す次の ... その一つが、形態素の頻度で利用した関数 RMeCabFreq( ) で、テキスト内の形態素原形の登場頻度をデータフレームとして ...

17/09/07 にこのページにアクセスしました。

形態素の頻度分析

www.ic.daito.ac.jp/~mizutani/mining/rmecabfreq.html

関数 RMeCabFreq( ) は指定されたテキストファイルを**形態素解析**して、その活用形を原形に変換した上で、その頻度を数えて、 ... また、Windows**R**を利用する場合にはデフォルト文字符号化を Shift-JIS に設定している場合には、文字符号化変換する必要が ...

このページに 2 回アクセスしています。前回のアクセス: 17/09/07

【R】【MeCab】RMeCabのインストールと形態素解析 - Qiita

qiita.com › 投稿 › R

2016/10/06 - **R**で**形態素解析**をしてみたので、インストールから簡単なデモまで一通り説明します。 **R**を使って**形態素解析**をすると、いろんなソフトを行ったり来たりせずに一貫して、分析が進められるのでなかなか便利です。 設定した環境iMac (27-inch, Mid ...

Rで形態素解析 RMecabを使ってみる : 都筑総研 - livedoor Blog

blog.livedoor.jp/g0031067/archives/51942787.html

- RMeCabを使う

Rで形態素解析

- 類似したツイートをしている人を探したい
- 同じ名詞をツイートする人を同グループに分類する

Rで形態素解析

docMatrix, docMatrix2, docMatrixDF

関数ファミリー docMatrix, docMatrix2, docMatrixDF は、**検索語・文書行列**（term-document matrix）を生成する関数である。 検索語・文書行列とは、あるterm（形態素原形）t_j が文書ファイル d_k 内に現れる頻度を f_{ij} として、指定したカテゴリの全てのm語のterm $t_j(j = 1 \ldots m)$ と検索する全てのn個の文書ファイル $d_k(k = 1 \ldots n)$ について求めた行列 $F = (f_{jk})$ のことである。 検索語・文書行列によって、文書間の類似性（あるいは差異）を検討することができる。 ある文書集合においてあるterm群が高い頻度で登場している一方で、別の文書集合では同じターム群が低い頻度でしか現れないとすれば、それら2つの文書集合は語彙の使い方において別種の文書グループとして類別してもよいだろう。

RMeCab

- **docMatrix(ディレクトリ名, pos=c("名詞"))**

- 返り値はターム行列

- 指定したディレクトリに含まれるすべてのファイルについて単語の出現頻度をまとめたリストを作成する

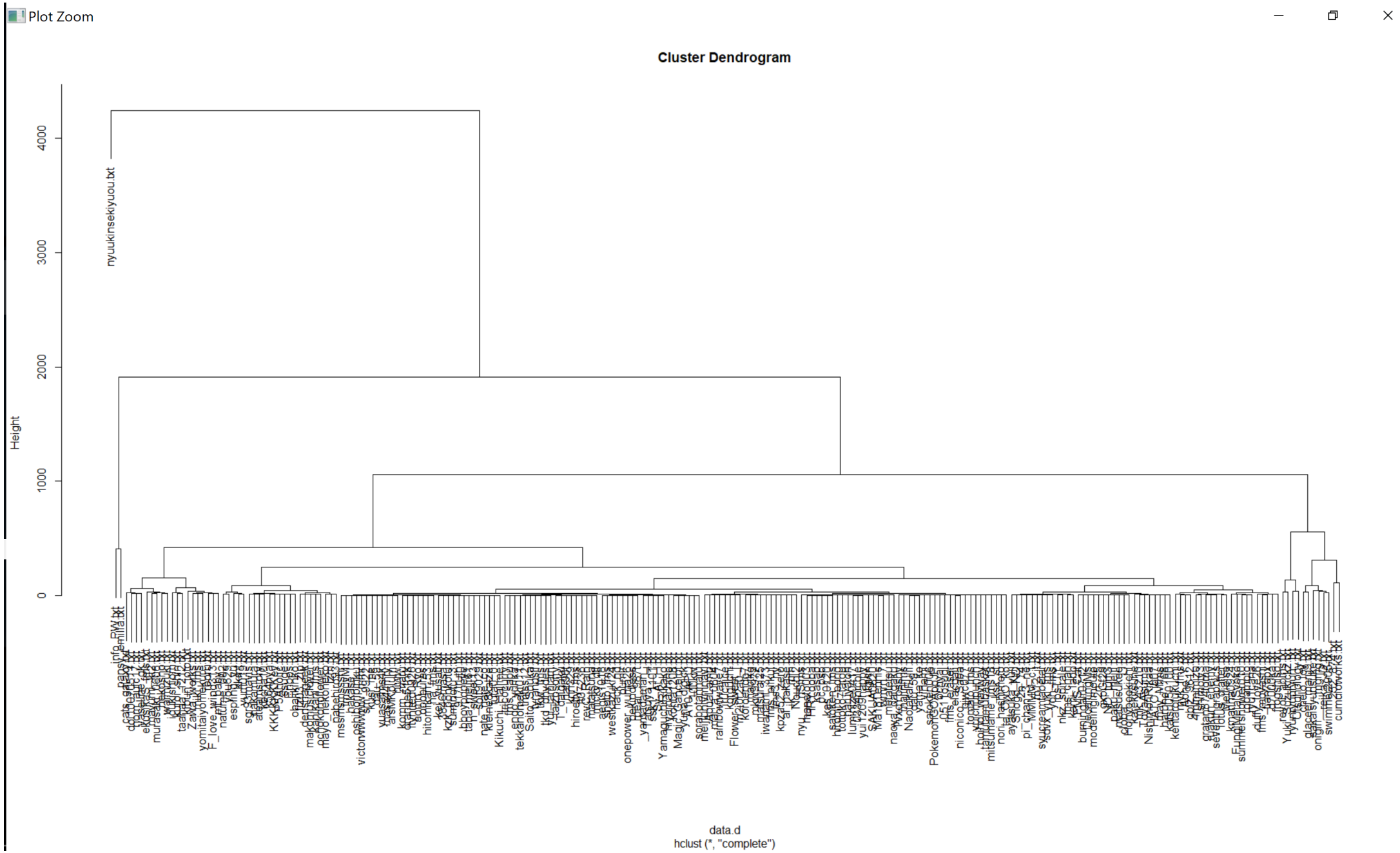
ターム行列

- 複数のドキュメント（D_i）をまとめ、行列にしたもののこと。ドキュメント内に表れるターム（T_j）の頻度と各ドキュメントを成分(D_ij)とします。タームは形態素で表現されます。

$$
\begin{array}{c|cccc}
 & T_1 & T_2 & \cdots & T_t \\
D_1 & d_{11} & d_{12} & \cdots & d_{1t} \\
D_2 & d_{21} & d_{22} & \cdots & d_{2t} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & d_{n1} & d_{n2} & \cdots & d_{nt}
\end{array}
$$

クラスタ分析

```
library(RMeCab)
data<-docMatrix("data/",pos=c("名詞"))
data.d<-dist(t(data))
clust<-hclust(d=data.d)
plot(clust)
```


Plot Zoom
Cluster Dendrogram
Height
0
1000
2000
3000
4000
nyuukinsekiyuou.txt
data.d
hclust (*, "complete")

無理

- 全然うまく分けられない（細かくなりすぎてしまう）
- 自分が別のグループだと認識している人も同じような発言をしている

手法

- フォロー関係があるかどうかで判別する
  →ネットワーク分析(19章)

- 同じ傾向のツイートがあるかどうかで判別する
  →形態素解析＋クラスタ分析

ネットワーク分析

- 何らかの関係の有無及びその度合いを調べるための分析手法

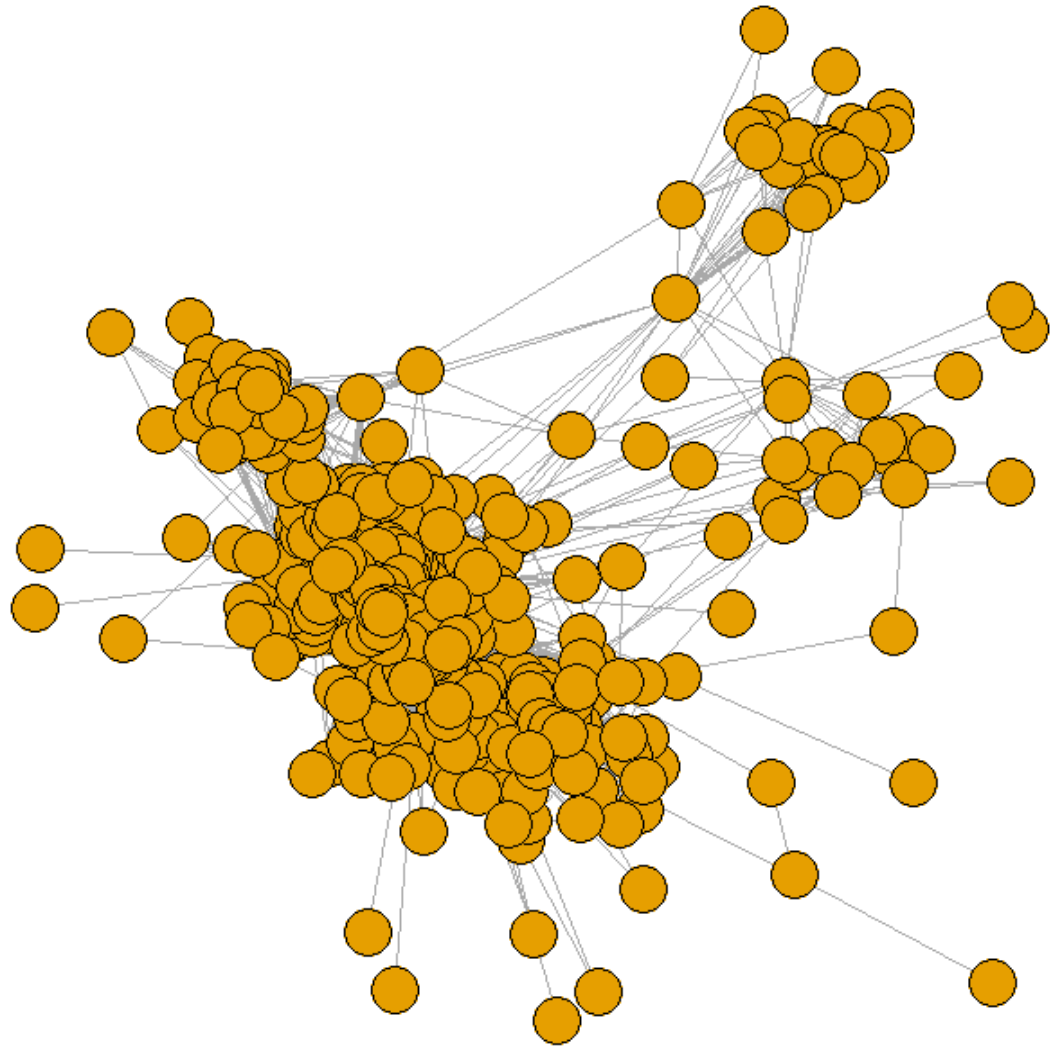
ネットワーク分析

- 自分がフォローしているユーザーについて、フォロー関係の有無を調べる
- 教科書で説明されていたigraphを用いる

```
import os

f = open('myFollow.txt')
data = f.read()  # ファイル終端まで全て読んだデータを返す
f.close()
member = data.split('\n')

os.chdir('./data')

fout=open("clustdata.txt","w")
for j in member:
    print(j)
    if not j=="":
        f2 = open(j+".txt")#それぞれのフォロワーの記述さ
        data2=f2.read()
        f2.close()
        anothermember=data2.split('\n')
        for k in anothermember:
            if k in member:
                fout.write(j+","+k+",1\n")
```

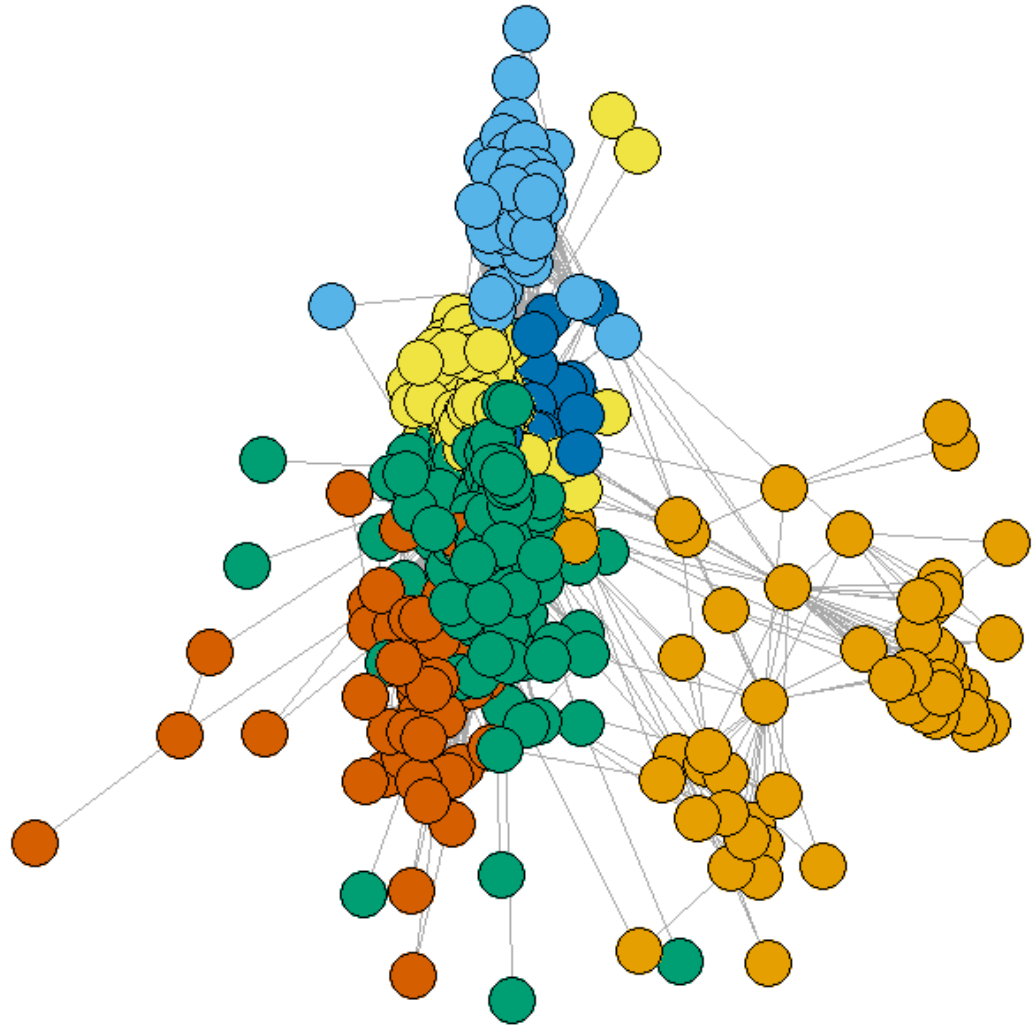
ネットワークの作成

```
library(igraph)
d <- read.table("clustdata.txt")
g <- graph.data.frame(d,directed = FALSE)
plot(g, vertex.label="",vertex.size=10)
```



コミュニティ分析

```
eb <- edge.betweenness.community(g)
V(g)$color <- eb$membership
plot(g,vertex.label="",vertex.size=10)
```



- いいバランスに分かれている（気がする）
- 当初の目標だったトレンドの取得をやってみる

頻度分析

```
eb1<-RMeCabFreq("eb1.txt")
eb1<-eb1[eb1$Info1=="名詞",]
eb1<-eb1[eb1$Info2=="一般"|eb1$Info2=="固有名詞",]
eb1<-eb1[order(eb1$Freq,decreasing = T),]
```

オタク	自分	人	口座	S	成績
自分	人	たより	残高	バイト	GPA
人	メッキ	イケ	気	先生	バイト
感じ	大人	自分	笑	GPA	基礎
あと	Twitter	グラブル	余裕	中野	自分