

▼ Table of Contents

- [1 目的 : Google page rankと線形代数](#)
- [2 解説](#)
 - [2.1 Google page rank](#)
 - [2.2 ランクの計算手順](#)
- [3 演習](#)
 - [3.1 Page Rankの手順の確認](#)
 - [3.1.1 ヒント](#)
 - [3.1.2 ヒント](#)
 - [3.1.3 ヒント](#)
 - [3.1.4 解答例](#)
 - [3.2 New heading](#)
 - [3.3 初期ベクトルがちがうと...](#)
 - [3.3.1 解答例](#)
 - [3.4 固有ベクトルから確認](#)
 - [3.4.1 解答例](#)

数式処理group work-4(Google page rank)

file:~/python/doing_math_with_python/group_works_4
cc by Shigeto R. Nishitani 2009-2018

▼ 1 目的 : Google page rankと線形代数

Googleの躍進の土台となったPageRankは線形代数の計算です。線形代数のコマンドを実際に使いながら、その中身を理解して行くことを目的としています。

▼ 2 解説

2.1 Google page rank

Googleのpage rankは非常に単純な仮定

「多くの良質なページからリンクされているページはやはり良質なページである」

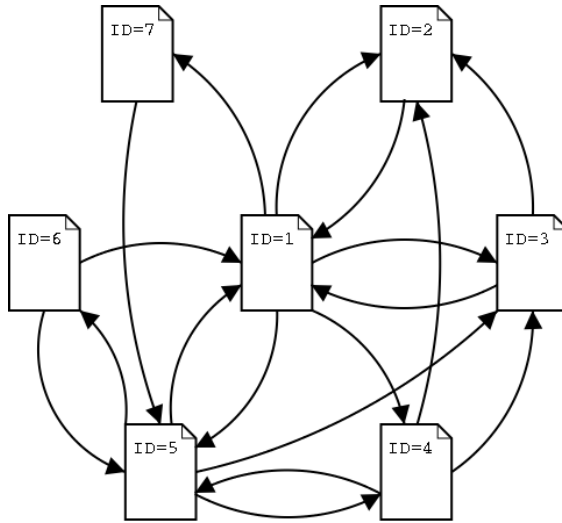
から成り立っている。博士論文のテーマを探していたスタンフォード大学院コンピュータサイエンス学部時代のラリー・ページ(Larry Page)が考案しました。

『そのアルゴリズムはページの名をとって「ページランク(Page Rank)」と呼ばれたが、特定のサイトに入るリンクの数と、リンクしたサイトのそれぞれに入るリンクの数の、その両方を考慮に入れる。これは学術論文の引用の度数計算の方法を手本にしており、予想通りに機能した』(『ザ・サーチ グーグルが世界を変えた』ジョン・バツェル著、中谷和男訳、2005年日経BP社)

詳しい解説は<http://ja.wikipedia.org/wiki/ページランク>
(<http://ja.wikipedia.org/wiki/%E3%83%9A%E3%83%BC%E3%82%B>にある。

▼ 2.2 ランクの計算手順

つぎのようなリンクが張られたページ群を考える。



まずは、リンクを再現する隣接行列を作る。ページに番号をつけて、その間が結ばれている $i - j$ 要素を1、そうでない要素を0とする。上の例では、

```

1 2 3 4 5 6 7
--
0 1 1 1 1 0 1
1 0 0 0 0 0 0
1 1 0 0 0 0 0
0 1 1 0 1 0 0
1 0 1 1 0 1 0
1 0 0 0 1 0 0
0 0 0 0 1 0 0

```

となる。

- この隣接行列を転置する。これはページランクが「どれだけリンクしているか」ではなく、「どれだけリンクされているか」を評価するためである。
- 転置した隣接行列 tM のそれぞれの列ベクトルの総和が1となるように規格化して「推移確率行列」をつくる。(ヒント参照)
- 初期ベクトルとして、すべての要素が同じ値で、足して1になるような長さ7の列ベクトルを用意する。
- 初期ベクトルに何度か(例えば10回、あるいは収束するまで)推移確率行列を掛ける。この操作は、行列の最大固有値に属する固有ベクトルを見つけることに相当する。
- 得られたベクトルの各要素が対応するページの得点とみなせ、

得点順にランクが高くなる。

▼ 3 演習

▼ 3.1 Page Rankの手順の確認

うえの手順にしたがって、「推移確率行列(transition probability matrix)」 $trans_mat$ を作り、初期ベクトル($init_v$)に5回ほど作用させて数字の変化を観察せよ。ページランクはどうなるか。

3.1.1 ヒント

手順2、3に於いて、和を取って規格化する代わりに、以下のようにして作成した対角(diagonal)行列 VA を、転置した隣接行列 tM に右からかければ推移確率行列が得られる。

```

> V = [1/5,1,1/2,1/3,1/4,1/2,1]
> diag(*V)

```

3.1.2 ヒント

pythonでMatrixでvectorを作ろうとすると行ベクトルになります。右から掛けるときには転置(transpose)操作が必要です。

3.1.3 ヒント

はじめに

```
init_printing()
```

をいれておくと、固有値ベクトルとかの表示が綺麗。

▶ 3.1.4 解答例

[...]

▼ 3.3 初期ベクトルがちがうと...

初期ベクトルを

```
> v_init=Matrix([100,0,0,0,0,0])
```

として、推移確率行列に右から掛け、それに伴うベクトルの各要素の変化を観察し、前問と比較せよ。その結果から、推移確率行列を掛けることによってどのように状態が推移していくか、漸近していく様子を記述せよ。

▶ 3.3.1 解答例 [\[...\]](#)

▶ 3.4 固有ベクトルから確認 [\[...\]](#)

推移確率行列の固有値・固有ベクトルをもとめ、最大固有値に対応する固有ベクトルを取り出せ。前問までに得られた結果と比較し、一致していることを確かめよ。ただし、固有ベクトルの大きさは任意であるため、`norm()`で規格化しておくと比較しやすい。

3.4.1 解答例 [\[...\]](#)