Simple PageRank Implementation in R

Ojeda Contreras Braulio Melquisedec

2023-01-11

First, we define our initial values.

```
dim <- 6
H <- matrix(c(0, 1/2, 1/2, 0, 0, 0,
              0, 0, 0, 0, 0, 0,
              1/3, 1/3, 0, 0, 1/3, 0,
              0, 0, 0, 0, 1/2, 1/2,
              0, 0, 0, 1/2, 0, 1/2,
              0, 0, 0, 1, 0, 0), ncol = dim, byrow = TRUE)

alpha <- 0.9
```

Then, we calculate the adjusted S matrix.

```
S <- H
for (i in 1:dim) {
  if (all(H[i, ] == 0))
    S[i, ] <- 1 / dim
}
print(S)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.0000000 0.5000000 0.5000000 0.0000000 0.0000000 0.0000000
## [2,] 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
## [3,] 0.3333333 0.3333333 0.0000000 0.0000000 0.3333333 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.0000000 0.5000000 0.5000000
## [5,] 0.0000000 0.0000000 0.0000000 0.5000000 0.0000000 0.5000000
## [6,] 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000
```

And also the Google matrix.

```
G <- alpha * S + sum((1 - alpha) * (1 / dim))
print(G)
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] 0.01666667 0.46666667 0.46666667 0.01666667 0.01666667 0.01666667
## [2,] 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667
## [3,] 0.31666667 0.31666667 0.01666667 0.01666667 0.31666667 0.01666667
## [4,] 0.01666667 0.01666667 0.01666667 0.01666667 0.46666667 0.46666667
## [5,] 0.01666667 0.01666667 0.01666667 0.46666667 0.01666667 0.46666667
## [6,] 0.01666667 0.01666667 0.01666667 0.91666667 0.01666667 0.01666667
```

Finally, we calculate the distribution for the n-step.

```
n <- 10000
pi_0 <- rep(1 / dim, dim)
pi_n <- pi_0
for (i in 1:n) {
  pi_n <- pi_n %*% G
}
cat(paste('pi_', as.character(i), sep = ''), '=', pi_n, '\n')
```

```
## pi_10000 = 0.03721197 0.05395735 0.04150565 0.3750808 0.2059983 0.2862459
```