NA07a PageRank

Ivan Slapničar

29. listopada 2018.

1 PageRank

1.1 Doba pretraživanje

google (and others)

- 50 milijardi stranica, 3.5 milijardi pretraga dnevno
- **PageRank**
- povijest, kontekst - kolačići, spremanja podataka (o Vama), 200+ parametara Mol11: https://www.mathworks.com/moler/exm/chapters/pagerank.pdf "C, Moler, 'Google PageRank', mathWorks, 2011."

1.2 PageRank

- Teorija grafova i linearna algebra
- C. Moler, Google PageRank

Neki programi:

- https://github.com/purzelrakete/Pagerank.jl
- https://gist.github.com/domluna/2b9358ccc89fee7d5e26

Probat ćemo primjer iz Molerovog članka.

```
In [1]: i = vec([ 2 6 3 4 4 5 6 1 1])
        j = vec([ 1 1 2 2 3 3 3 4 6])
```

```
Out[1]: 9-element Array{Int64,1}:
         1
         1
         2
         2
         3
```

```
 3
 3
 4
 6
```

```
In [2]: G=sparse(i,j,1.0)
```

```
Out[2]: 6×6 SparseMatrixCSC{Float64,Int64} with 9 stored entries:
          [2, 1]  =  1.0
          [6, 1]  =  1.0
          [3, 2]  =  1.0
          [4, 2]  =  1.0
          [4, 3]  =  1.0
          [5, 3]  =  1.0
          [6, 3]  =  1.0
          [1, 4]  =  1.0
          [1, 6]  =  1.0
```

```
In [3]: typeof(G)
```

```
Out[3]: SparseMatrixCSC{Float64,Int64}
```

```
In [4]: full(G)
```

```
Out[4]: 6×6 Array{Float64,2}:
         0.0  0.0  0.0  1.0  0.0  1.0
         1.0  0.0  0.0  0.0  0.0  0.0
         0.0  1.0  0.0  0.0  0.0  0.0
         0.0  1.0  1.0  0.0  0.0  0.0
         0.0  0.0  1.0  0.0  0.0  0.0
         1.0  0.0  1.0  0.0  0.0  0.0
```

```
In [5]: c=sum(G,1)
        n=size(G,1)
        for j=1:n
            if c[j]>0
                G[:,j]=G[:,j]/c[j]
            end
        end
```

```
In [6]: full(G)
```

```
Out[6]: 6×6 Array{Float64,2}:
         0.0  0.0  0.0       1.0  0.0  1.0
         0.5  0.0  0.0       0.0  0.0  0.0
         0.0  0.5  0.0       0.0  0.0  0.0
         0.0  0.5  0.333333  0.0  0.0  0.0
         0.0  0.0  0.333333  0.0  0.0  0.0
         0.5  0.0  0.333333  0.0  0.0  0.0
```

- p je vjerojatnost da pratimo neki link
- $1-p$ je vjerojatnost da posjetimo slučajnu stranicu
- google koristi $p = 0.85$?

```
In [7]: p=0.85
        δ=(1-p)/n
```

```
Out[7]: 0.025000000000000005
```

```
In [8]: z = ((1-p)*(c.!=0) + (c.==0))/n
```

```
Out[8]: 1×6 Array{Float64,2}:
         0.025  0.025  0.025  0.025  0.166667  0.025
```

```
In [9]: A=p*G+ones(n)*z
```

```
Out[9]: 6×6 Array{Float64,2}:
         0.025  0.025  0.025     0.875  0.166667  0.875
         0.45   0.025  0.025     0.025  0.166667  0.025
         0.025  0.45   0.025     0.025  0.166667  0.025
         0.025  0.45   0.308333  0.025  0.166667  0.025
         0.025  0.025  0.308333  0.025  0.166667  0.025
         0.45   0.025  0.308333  0.025  0.166667  0.025
```

```
In [10]: sum(A,1)
```

```
Out[10]: 1×6 Array{Float64,2}:
          1.0  1.0  1.0  1.0  1.0  1.0
```

1.3 Ideja

Započnimo slučajnu šetnju iz vektora $x_0 = \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ 1/n \end{bmatrix}$.

Sljedeći vektori su

$$x_1 = A \cdot x_0 x_2 = A \cdot x_1 x_3 = A \cdot x_2 \vdots$$

Kada se vektor *stabilizira*:

$$A \cdot x \approx x,$$

tada je $x[i]$ *rang stranice* i.

```
In [12]: function myPageRank(G::SparseMatrixCSC{Float64,Int64},steps::Int)
             p=0.85
             c=sum(G,1)/p
             n=size(G,1)
             for i=1:n
                 G.nzval[G.colptr[i]:G.colptr[i+1]-1]./=c[i]
             end
             e=ones(n)
             x=e/n
             z = vec(((1-p)*(c.!=0) + (c.==0))/n)
             for j=1:steps
                 x=G*x+(z⋅x)
             end
             x/norm(x,1)
         end

Out[12]: myPageRank (generic function with 1 method)

In [13]: fieldnames(G)

Out[13]: 5-element Array{Symbol,1}:
          :m
          :n
          :colptr
          :rowval
          :nzval

In [14]: G

Out[14]: 6×6 SparseMatrixCSC{Float64,Int64} with 9 stored entries:
           [2, 1]  =  0.5
           [6, 1]  =  0.5
           [3, 2]  =  0.5
           [4, 2]  =  0.5
           [4, 3]  =  0.333333
           [5, 3]  =  0.333333
           [6, 3]  =  0.333333
           [1, 4]  =  1.0
           [1, 6]  =  1.0

In [15]: G.colptr

Out[15]: 7-element Array{Int64,1}:
           1
           3
           5
           8
```

```
 9
 9
10
```

```
In [16]: G.nzval
```

```
Out[16]: 9-element Array{Float64,1}:
          0.5
          0.5
          0.5
          0.5
          0.333333
          0.333333
          0.333333
          1.0
          1.0
```

```
In [17]: # Početni vektor
         x=ones(n)/n
```

```
Out[17]: 6-element Array{Float64,1}:
          0.166667
          0.166667
          0.166667
          0.166667
          0.166667
          0.166667
```

```
In [18]: myPageRank(G,15)
```

```
Out[18]: 6-element Array{Float64,1}:
          0.321024
          0.170538
          0.106596
          0.136795
          0.0643103
          0.200737
```

1.4 Stanford web graph

Nešto veći testni problem.

```
In [19]: W=readdlm("web-Stanford.txt",Int)
```

```
Out[19]: 2312497×2 Array{Int64,2}:
               1    6548
```

```
     1   15409
  6548   57031
 15409   13102
     2   17794
     2   25202
     2   53625
     2   54582
     2   64930
     2   73764
     2   84477
     2   98628
     2  100193
     ⋮
281849  165189
281849  177014
281849  226290
281849  243180
281849  244195
281849  247252
281849  281568
281865  186750
281865  225872
281888  114388
281888  192969
281888  233184
```

```
In [20]: ?sparse;
```

```
search: sparse sparsevec SparseVector SparseArrays SparseMatrixCSC issparse
```

```
In [21]: S=sparse(W[:,2],W[:,1],1.0)
```

```
Out[21]: 281903×281903 SparseMatrixCSC{Float64,Int64} with 2312497 stored entries:
           [6548  ,      1]  =  1.0
           [15409 ,      1]  =  1.0
           [17794 ,      2]  =  1.0
           [25202 ,      2]  =  1.0
           [53625 ,      2]  =  1.0
           [54582 ,      2]  =  1.0
           [64930 ,      2]  =  1.0
           [73764 ,      2]  =  1.0
           [84477 ,      2]  =  1.0
           [98628 ,      2]  =  1.0
           ⋮
           [168703, 281902]  =  1.0
```

```
 [180771, 281902]  =  1.0
 [266504, 281902]  =  1.0
 [275189, 281902]  =  1.0
 [44103 , 281903]  =  1.0
 [56088 , 281903]  =  1.0
 [90591 , 281903]  =  1.0
 [94440 , 281903]  =  1.0
 [216688, 281903]  =  1.0
 [256539, 281903]  =  1.0
 [260899, 281903]  =  1.0
```

```
In [22]: @time x100=myPageRank(S,100);
```

```
  5.516720 seconds (282.42 k allocations: 488.428 MiB, 3.83% gc time)
```

```
In [23]: x101=myPageRank(S,101);
```

```
In [24]: maximum(abs,(x101-x100)./x101)
```

```
Out[24]: 2.3491375260608525e-7
```

```
In [25]: # Ranks
         sort(x100,rev=true)
```

```
Out[25]: 281903-element Array{Float64,1}:
          0.0113029
          0.00926783
          0.00829727
          0.00302312
          0.00300128
          0.00257173
          0.00245371
          0.00243079
          0.00239105
          0.00236401
          0.002301
          0.00226742
          0.00223245
          ⋮
          5.33369e-7
          5.33369e-7
          5.33369e-7
          5.33369e-7
          5.33369e-7
          5.33369e-7
```

```
 5.33369e-7
 5.33369e-7
 5.33369e-7
 5.33369e-7
 5.33369e-7
 5.33369e-7
```

```
In [26]: # Pages
         sortperm(x100,rev=true)
```

```
Out[26]: 281903-element Array{Int64,1}:
  89073
 226411
 241454
 262860
 134832
 234704
 136821
  68889
 105607
  69358
  67756
 225872
 186750
      ⋮
 281627
 281646
 281647
 281700
 281715
 281728
 281778
 281785
 281813
 281849
 281865
 281888
```