

Understanding social networks with F#

Evelina Gabasova
@evelgab

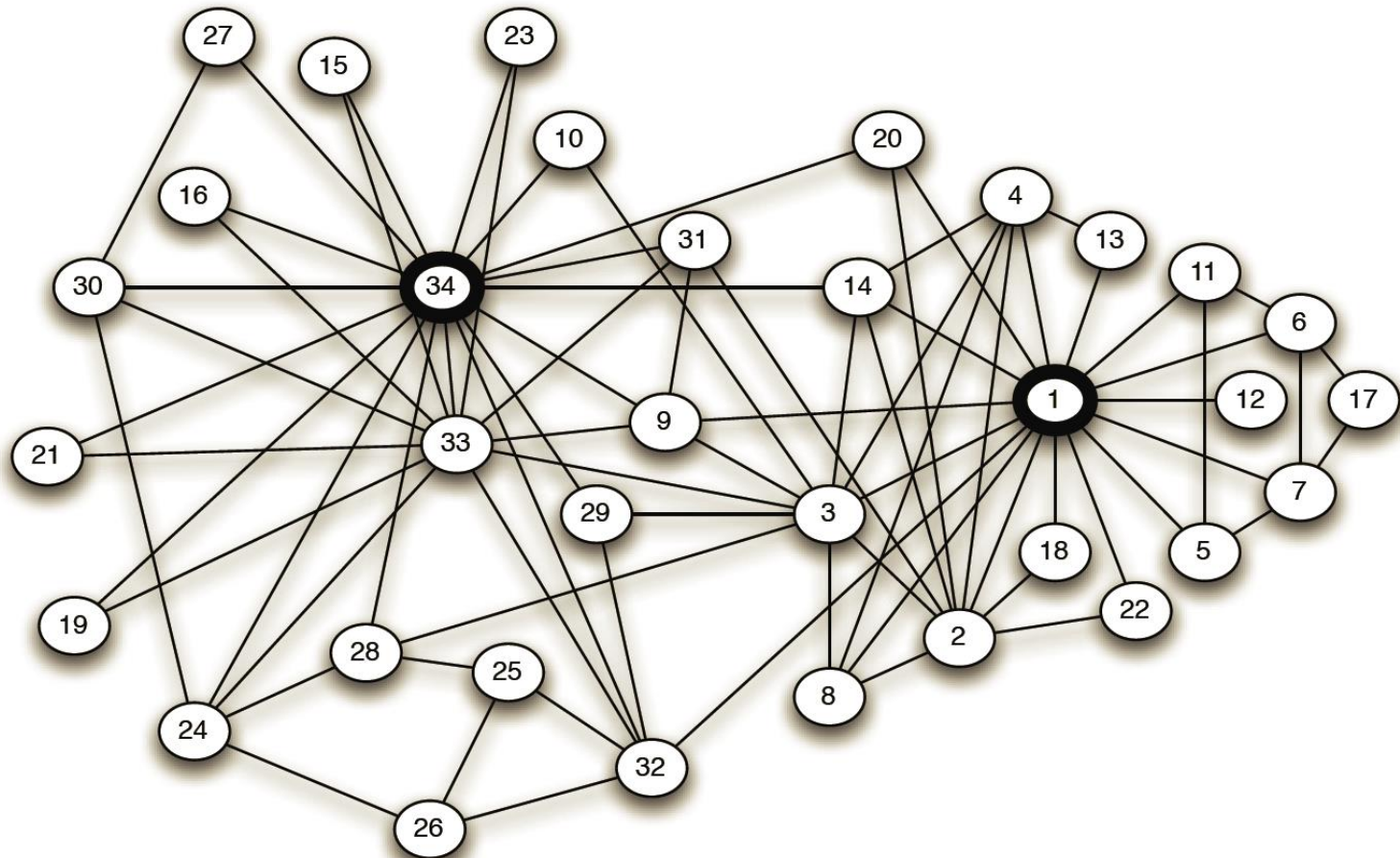
NYC F# user group
4 December 2014

Why network science

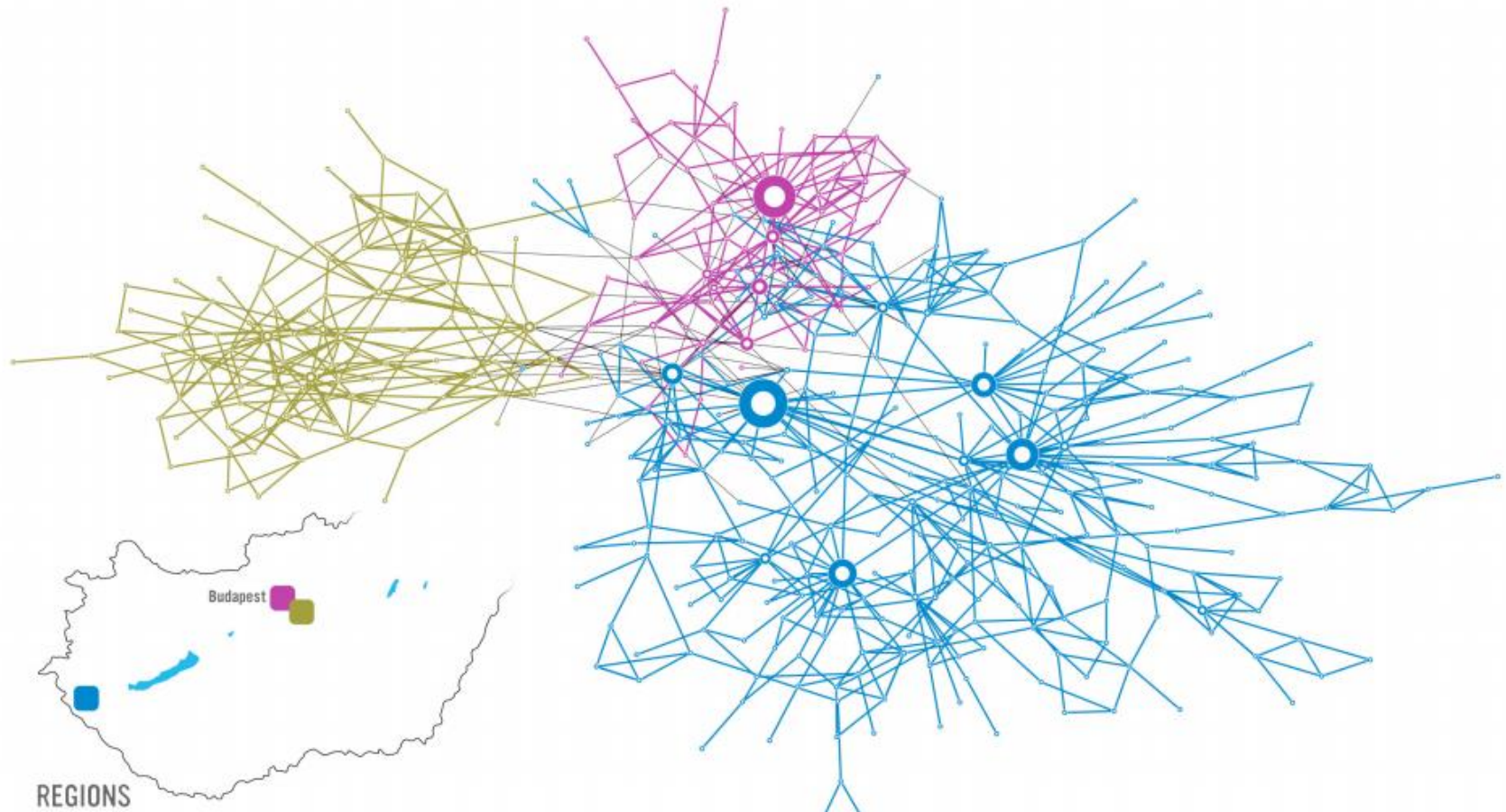
- gene interaction networks
- disease spreading
- balancing load in power grids
- brain connections
- software dependency networks
- social networks
- ...

Social network analysis

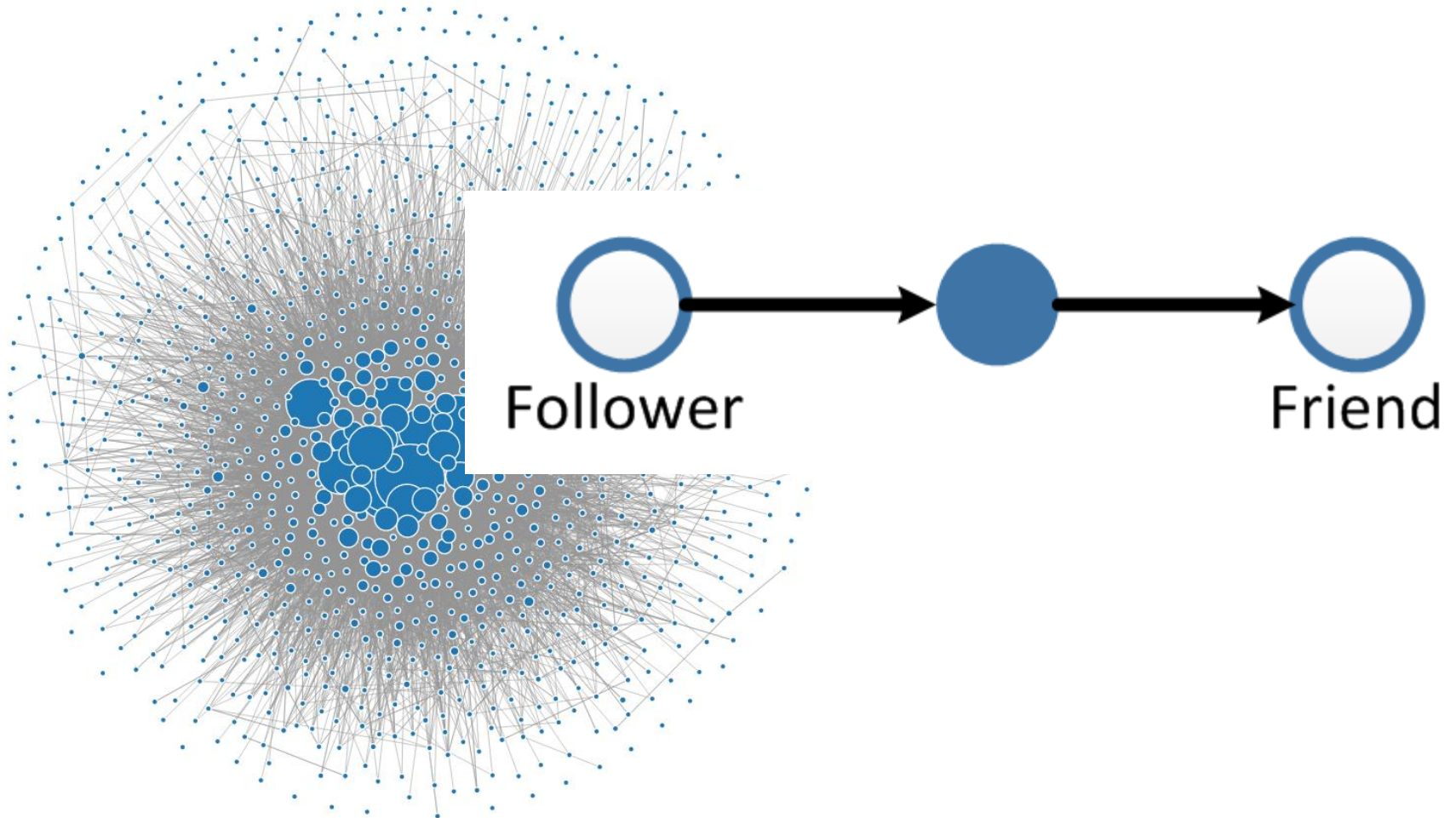
Karate club network



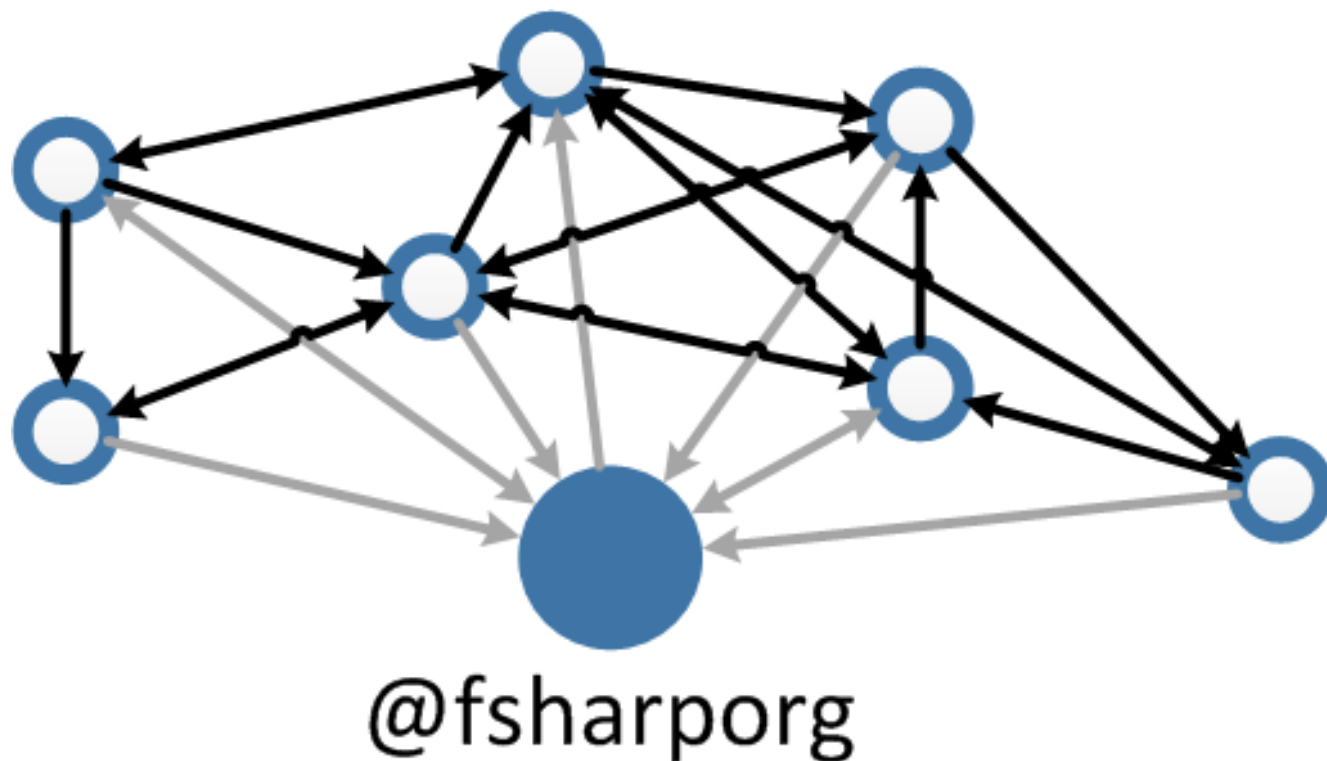
Insights from social networks



Twitter network



How large is your ego?



Connecting to twitter



A screenshot of a Twitter authorization window. The window has a title bar with standard OS controls. The header bar is black with the Twitter logo on the left and a "Sign up for Twitter" link on the right. The main content area is divided into two columns. The left column contains the heading "Authorise F# demo to use your account?", a list of permissions ("Read Tweets from your timeline." and "See who you follow."), input fields for "Username or email" and "Password", a "Remember me" checkbox with a "Forgot password?" link, and "Authorise app" and "Cancel" buttons. The right column features a blue gear icon with a white Twitter bird, the text "F# demo", a GitHub repository link, and a description of the application.

Sign up for Twitter ›

Authorise F# demo to use your account?

This application **will be able to:**

- Read Tweets from your timeline.
- See who you follow.

Username or email

Password

☐ Remember me · [Forgot password?](#)

Authorise app Cancel



F# demo
github.com/evelinag/FS...
Example application that shows how to interact with Twitter in F#

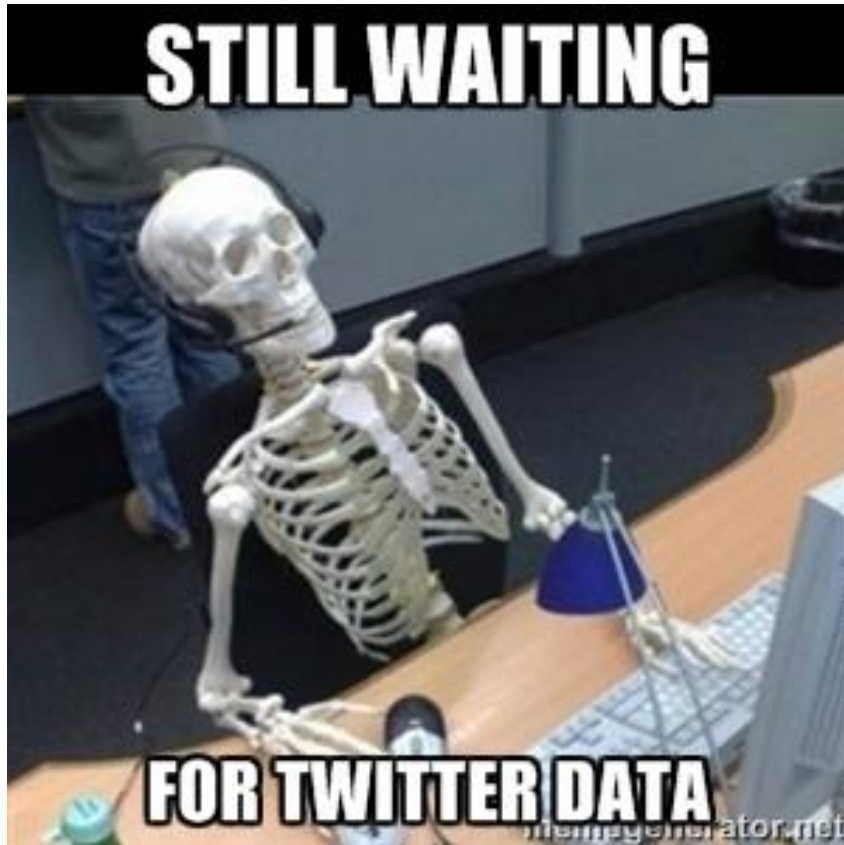


Interactions on Twitter

Retweets and Favorites



Downloading data

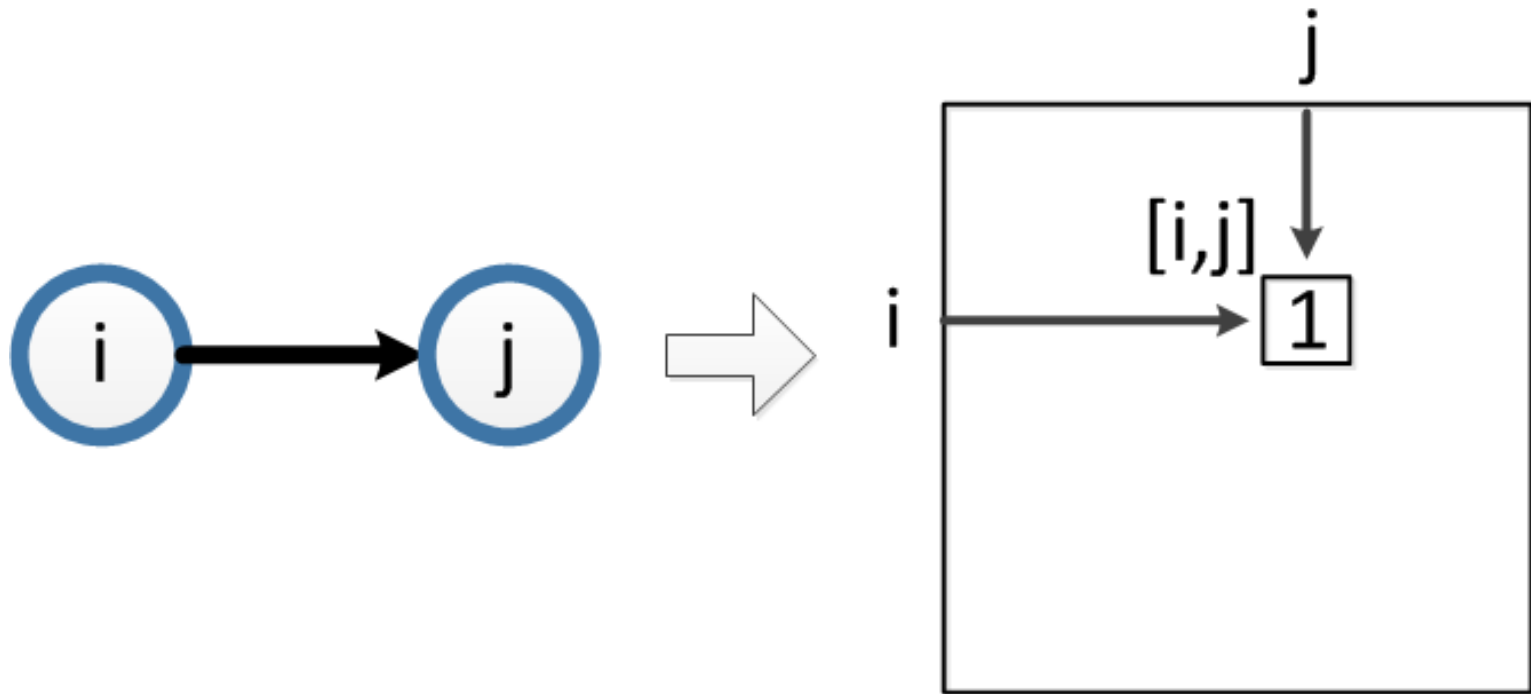


- 1) List of nodes
- 2) Connections between nodes

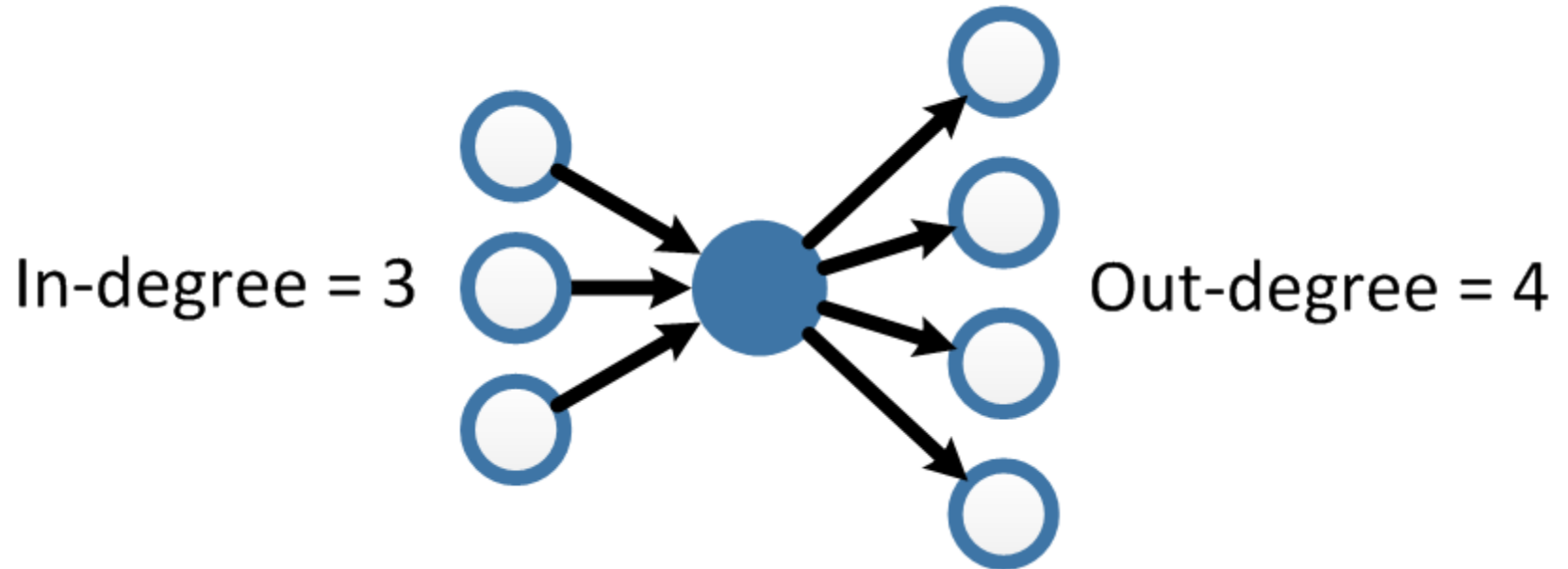
Twitter API allows only 15 requests every 15 minutes to list connections.



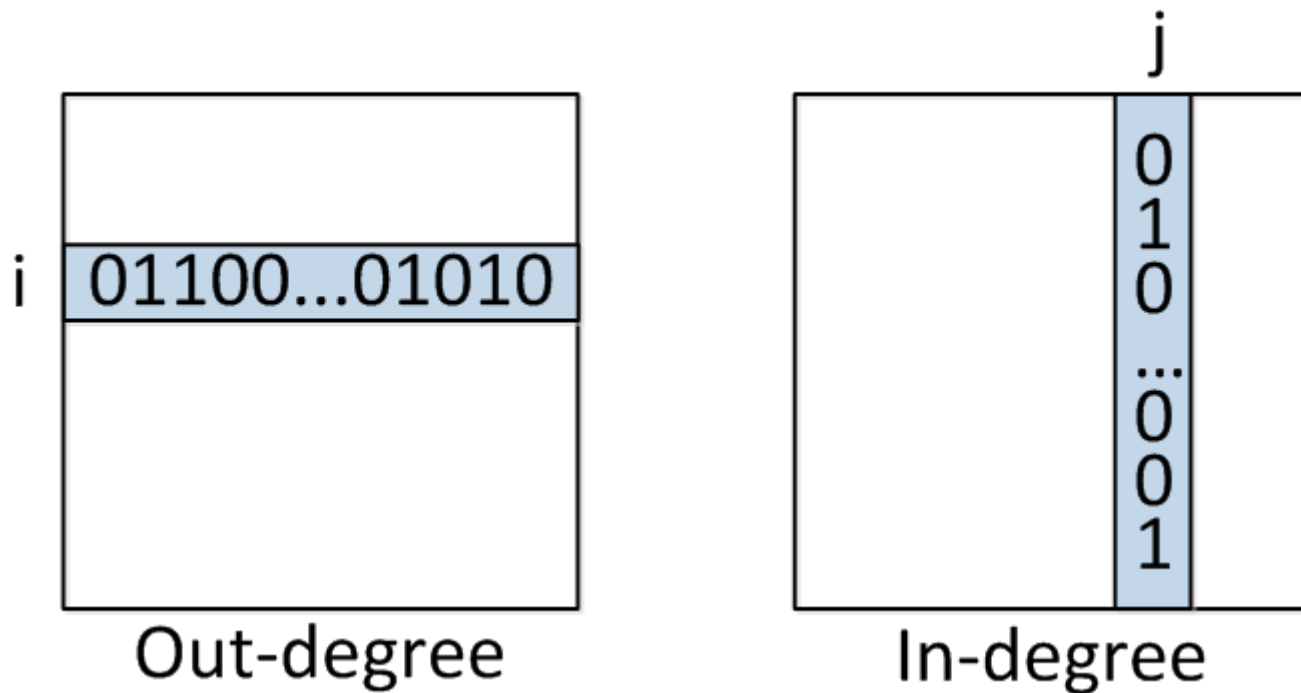
Adjacency matrix



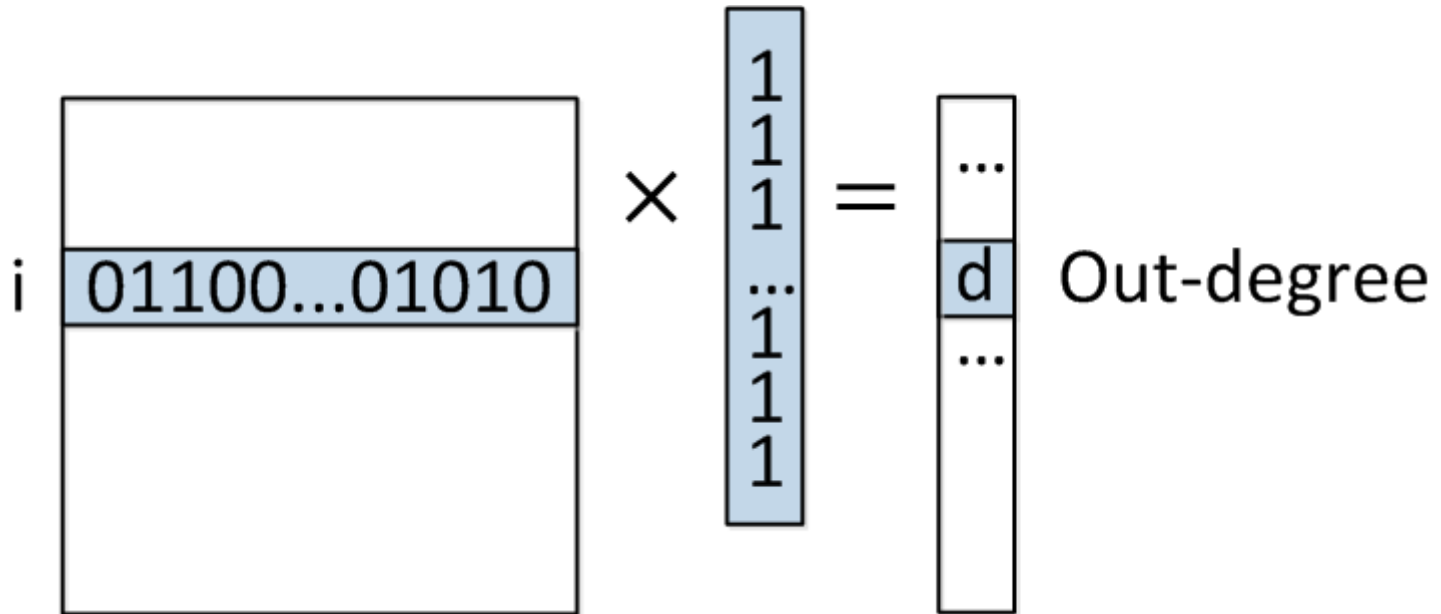
Degrees



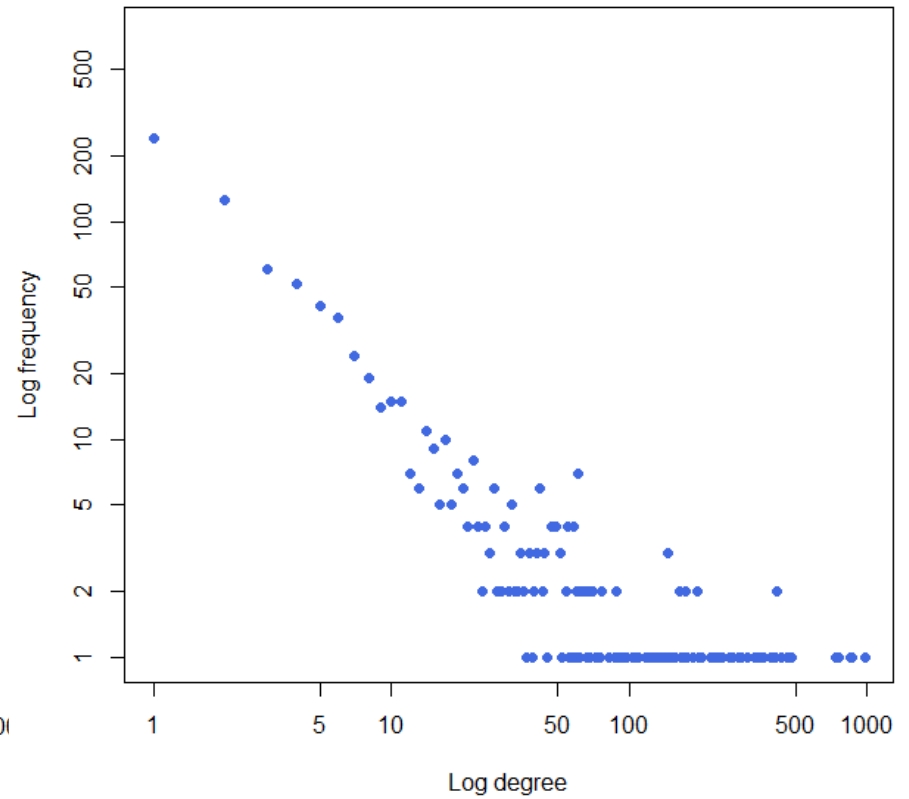
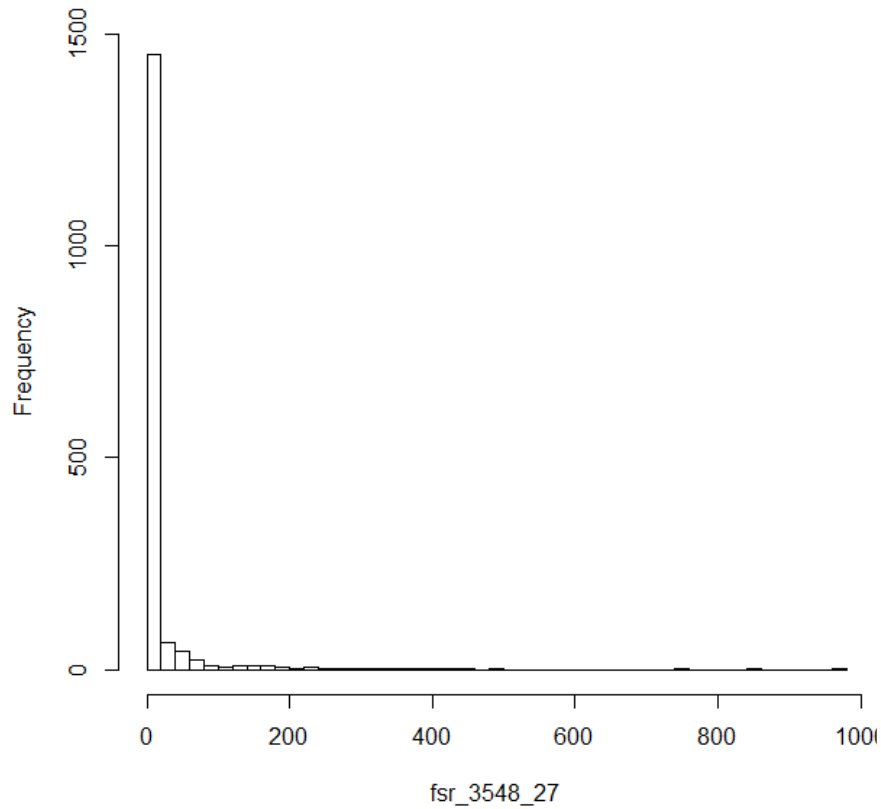
Degrees



Degrees



Degree distribution

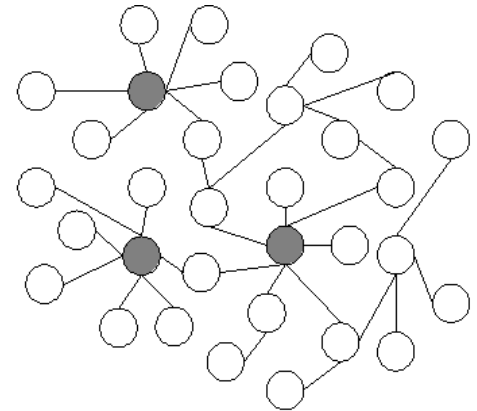
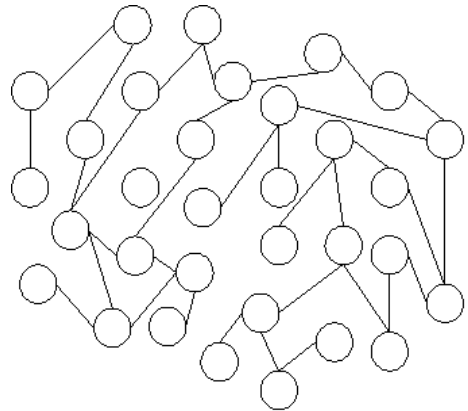


Scale-free networks

- Power law

$$P(d) \sim d^{-\gamma}$$

- Networks growing over time with preferential attachment
- Hubs
- Robustness



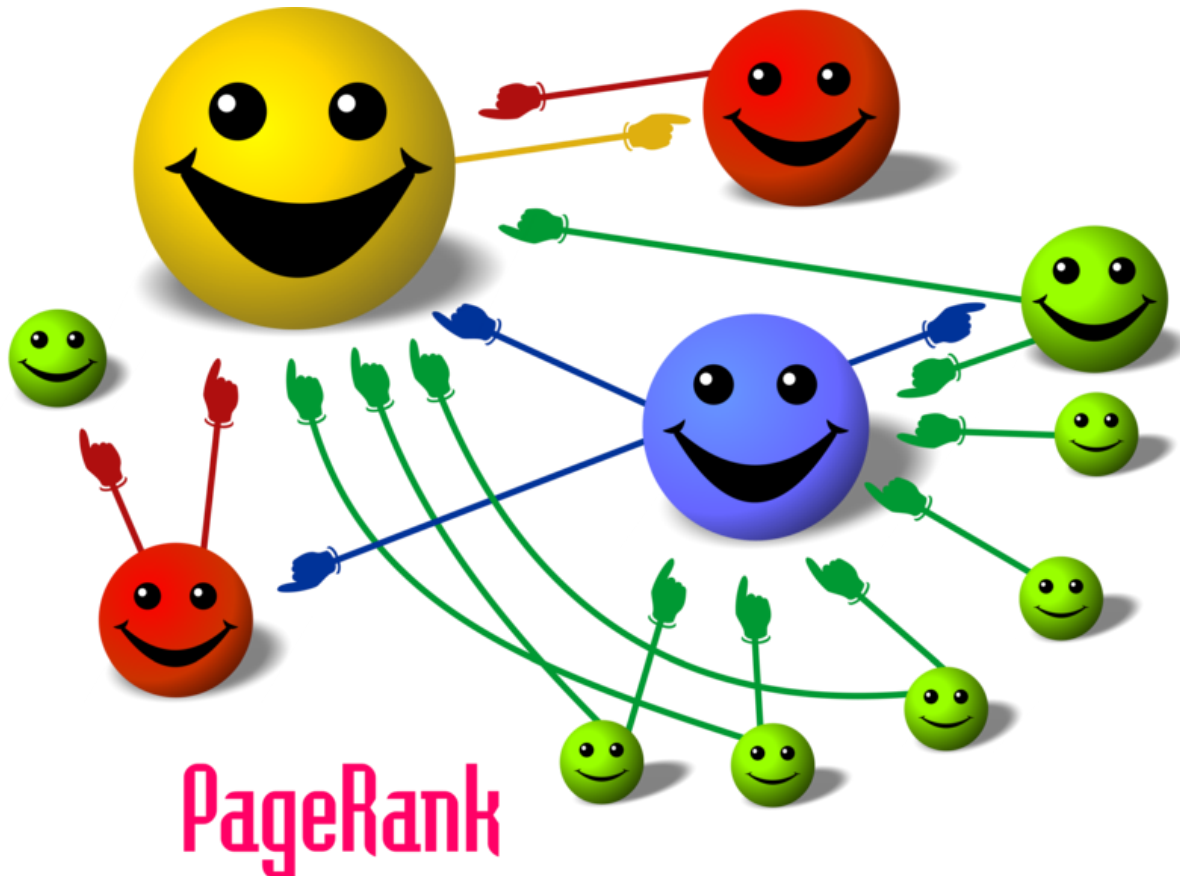
- Your friends have more friends than you have.

Hubs in networks

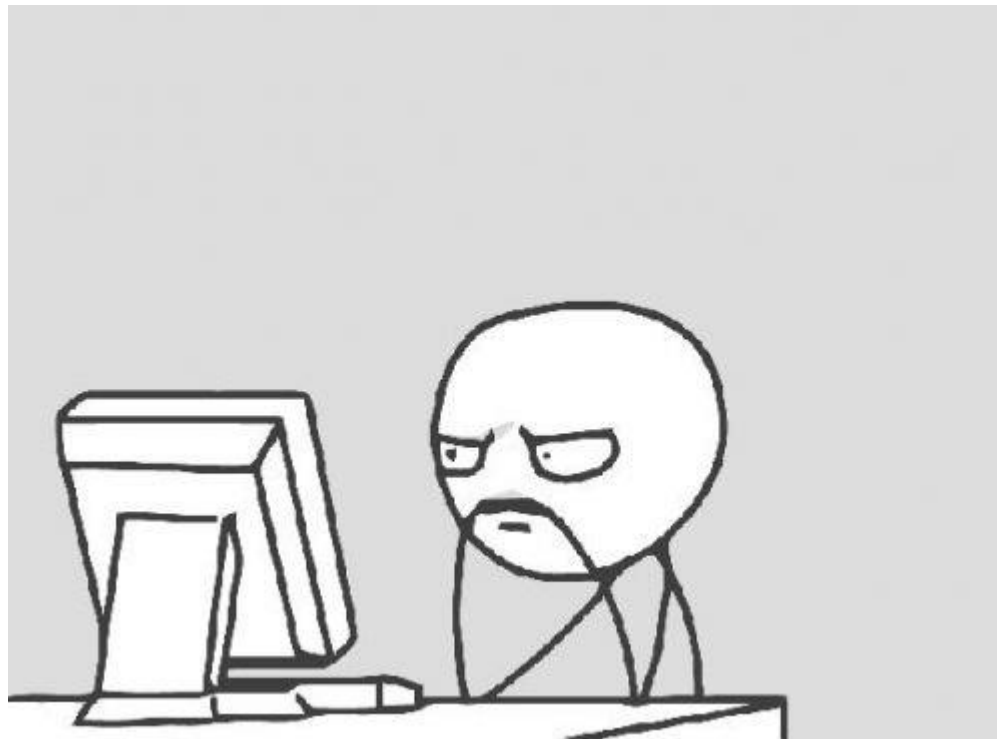
How to identify most
important nodes in a network?

Centrality with PageRank

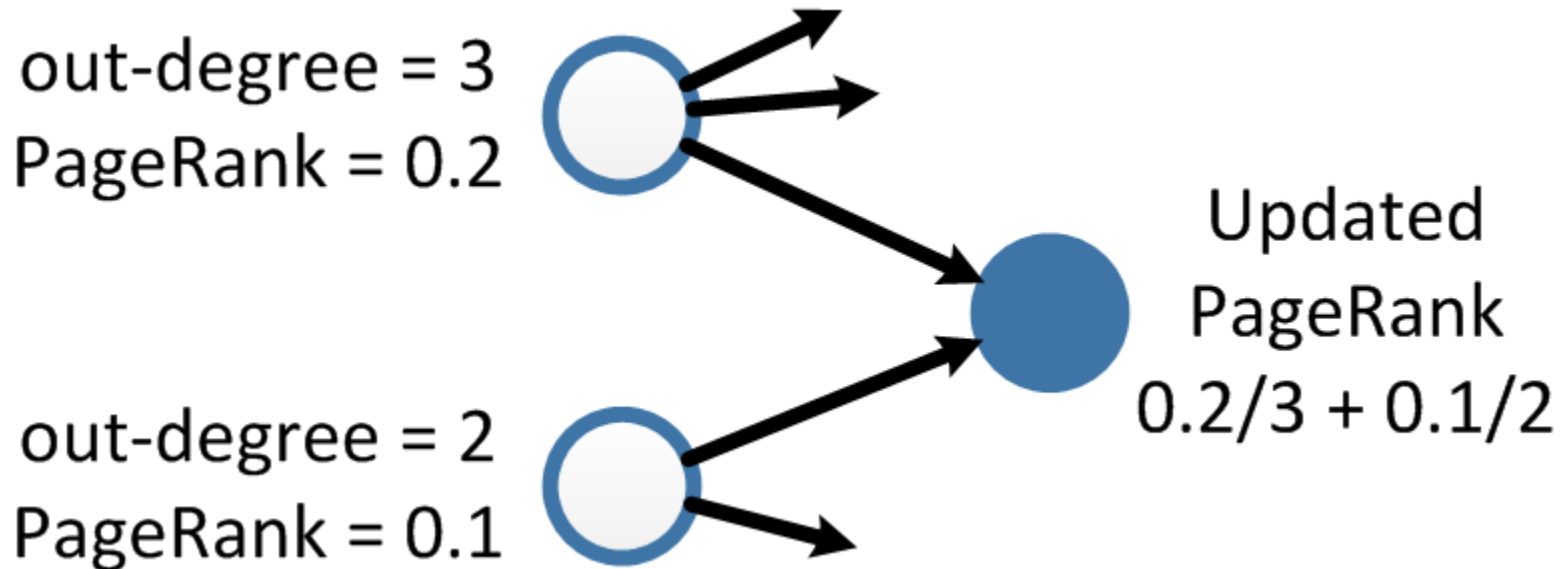
Your followers are not created equal.



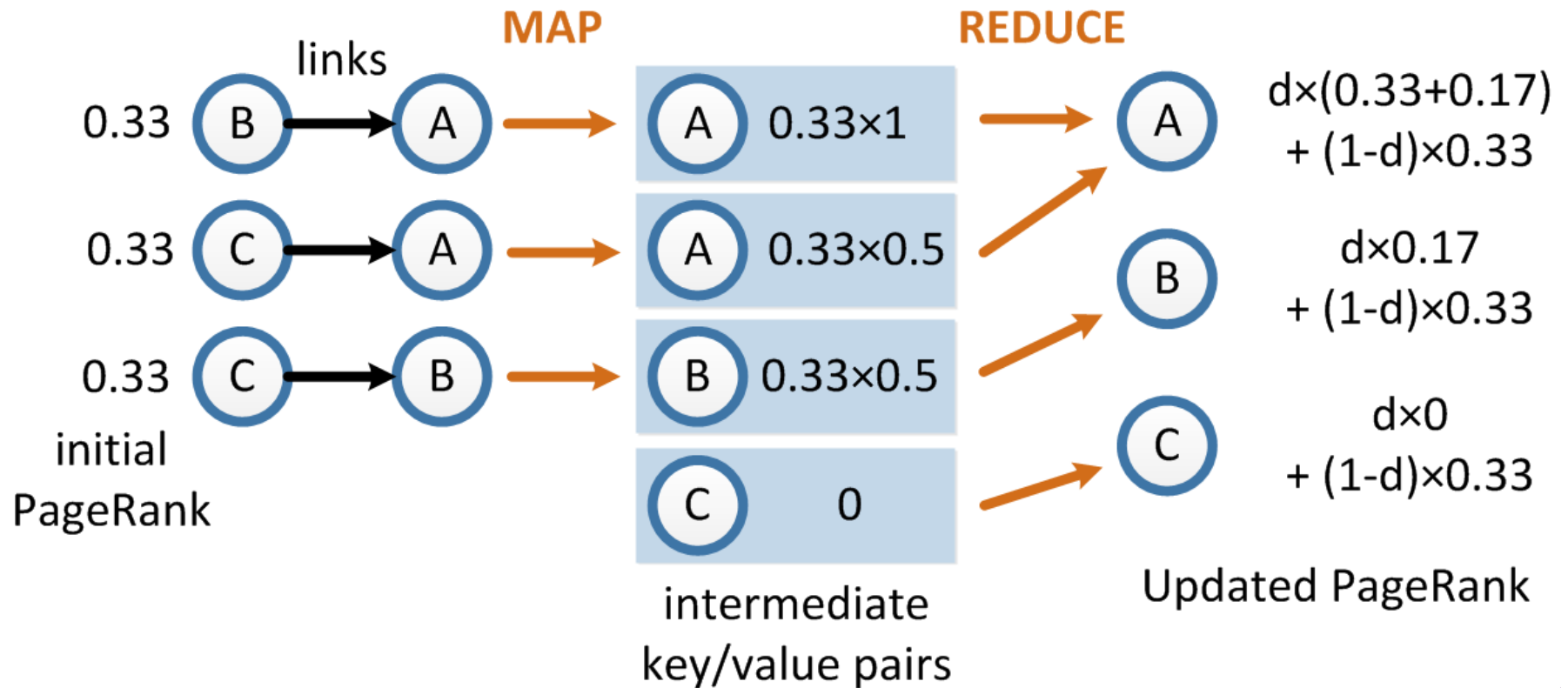
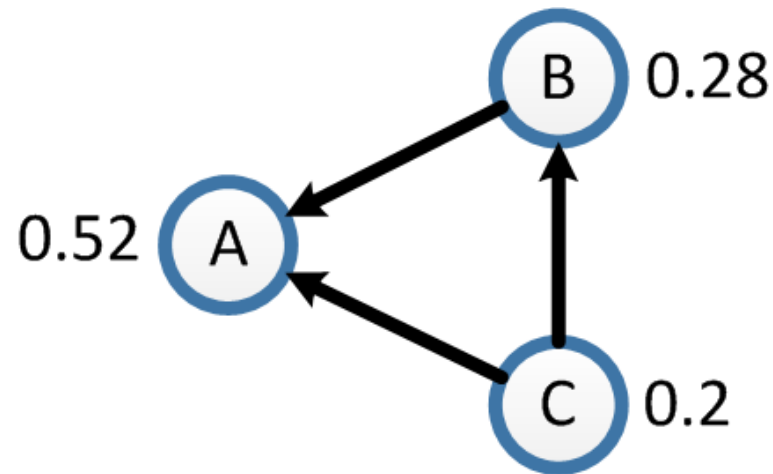
Random surfer model



Centrality with PageRank



+ random jumps



PageRank changes

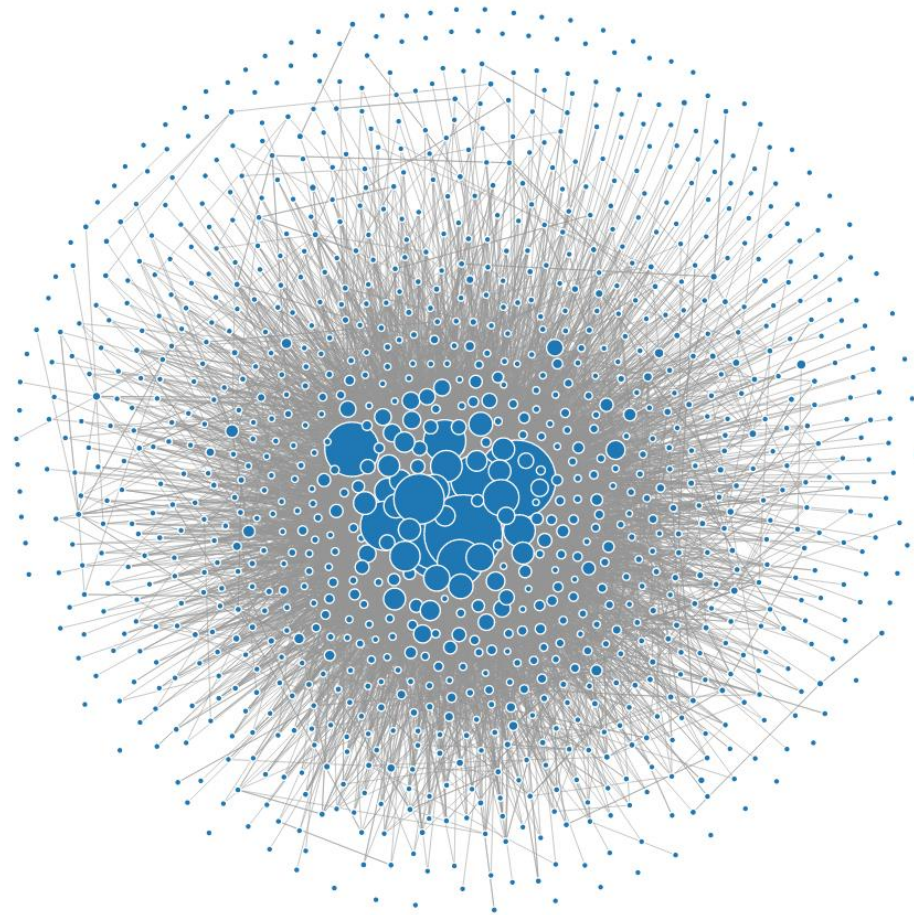
February

1. migueldeicaza (0.033130)
2. dsyme (0.032783)
3. tomaspetricek (0.027756)
4. LincolnAtkinson (0.021993)
5. VisualFSharp (0.020233)
6. c4fsharp (0.019720)
7. rickasaurus (0.019189)
8. ptrelford (0.018099)
9. 1tgr (0.016525)
10. sforkmann (0.014970)

November

1. dsyme (0.026928)
2. VisualFSharp (0.023572)
3. migueldeicaza (0.023111)
4. tomaspetricek (0.019736)
5. c4fsharp (0.018511)
6. rickasaurus (0.012992)
7. FSPowerTools (0.012077)
8. sforkmann (0.011950)
9. ptrelford (0.011787)
10. 1tgr (0.011256)

Visualization with D3.js



Spectral analysis

Eigenvalue decomposition

$$Av = \lambda v$$

Eigenvectors show centrality and
community structure

Non-backtracking matrix

Spectral analysis

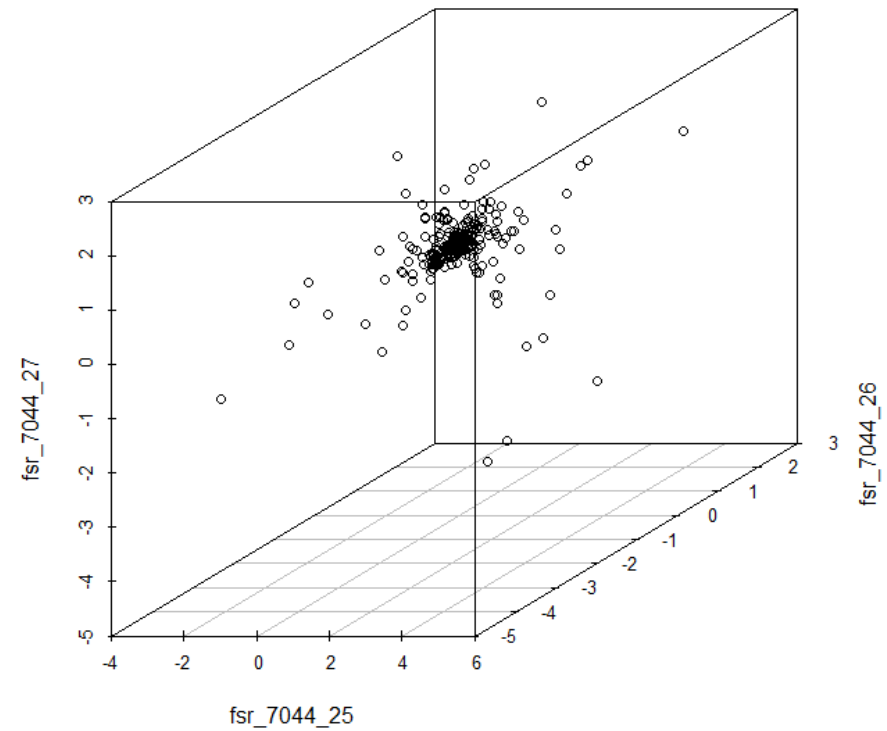
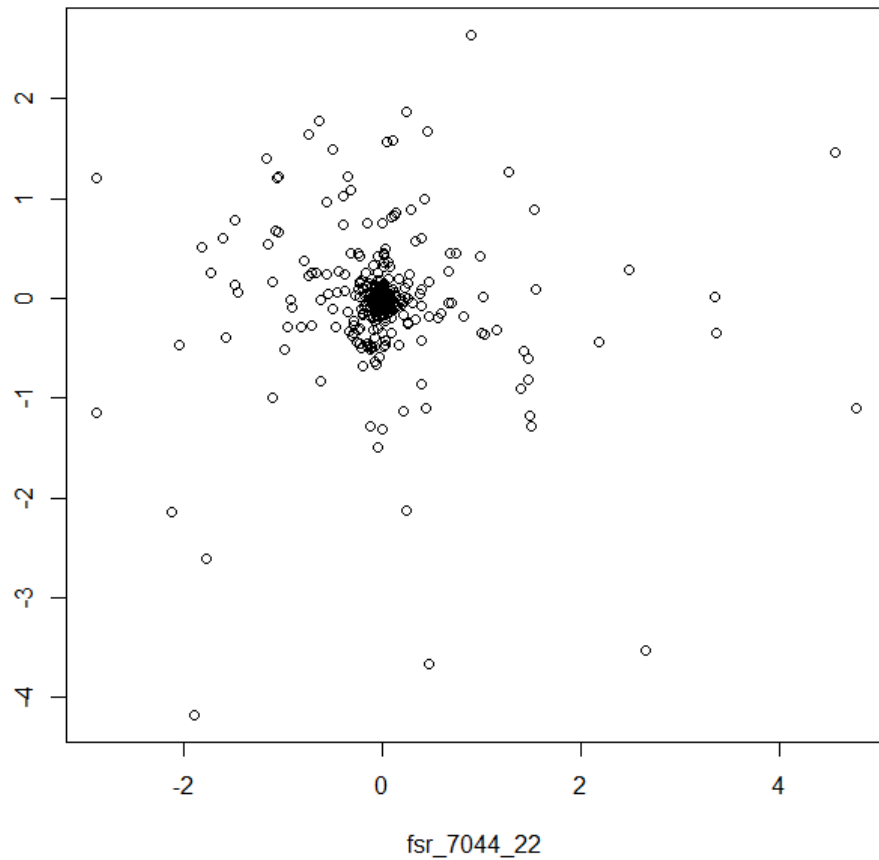
PageRank

1. dsyme
2. VisualFSharp
3. migueldeicaza
4. tomaspetricek
5. c4fsharp
6. rickasaurus
7. FSPowerTools
8. sforkmann
9. ptrelford
10. 1tgr

1st eigenvector

1. dsyme
2. c4fsharp
3. sforkmann
4. ptrelford
5. tomaspetricek
6. brandewinder
7. foxyjackfox
8. rickasaurus
9. sergey_tihon
10. panesofglass

Spectral analysis



Go play
with data

Why F#?

```
,
Lp = lapply(logL, exp)
for(m in 1:M) {
  for(i in 1:N) L[[m]][i,] = rmultinom(1,1,Lp[[m]][i,]) #Generate L from Lp
  if(w>1) L[[m]] = AlignClusters(C,L[[m]], type = 'mat') #Helps to align indices
  n[m,] = colSums(L[[m]])
  for(k in 1:K) { ###Update cluster parameters based on normal-gamma distribution
    if(d[m]==1&n[m,k]>1) {
      S[[m]][,k] = sd(X[[m]][,L[[m]][,k]==1])^2
      PostMean = sum(X[[m]][,L[[m]][,k]==1])/(n[m,k]+1)
      B[[m]][,k] = b0[[m]]+0.5*(n[m,k]*S[[m]][,k]+n[m,k]*(mean(X[[m]][,L[[m]][,k]==1])-mu0[[m]])^2.
    }
    if(d[m]>1&n[m,k]>1) {
      PostMean = (mu0[[m]]+rowSums(X[[m]][,L[[m]][,k]==1]))/(n[m,k]+1)
      S[[m]][,k] = apply(X[[m]][,L[[m]][,k]==1],MARGIN=1,FUN='sd')^2
      B[[m]][,k] = b0[[m]]+0.5*(n[m,k]*S[[m]][,k]+n[m,k]*(rowMeans(X[[m]][,L[[m]][,k]==1])-mu0[[m]])^2.
    }
    if(n[m,k]==1) {
      PostMean = (mu0[[m]]+X[[m]][,L[[m]][,k]==1])/2
      B[[m]][,k] = b0[[m]]+0.5*(X[[m]][,L[[m]][,k]==1]-mu0[[m]])^2/2}
    if(n[m,k]==0) {
      PostMean = mu0[[m]]
      B[[m]][,k] = b0[[m]]}
    Lambda = 1+n[m,k]
    A[[m]][,k] = a0[[m]]+n[m,k]/2
    Tau[[m]][,k] = rgamma(d[m],shape=A[[m]][,k],rate=B[[m]][,k])
    mu[[m]][,k] = rnorm(d[m],PostMean,sqrt(1/(Tau[[m]][,k]*Lambda)))
    Sigma[[m]][,k] = sqrt(1/Tau[[m]][,k])}
  }
```

Thank you

@evelgab

evelina@evelinag.com

github.com/evelinag

fsharp.org

F# eXchange 2015

17 April, London

