

# Networks: structure, evolution & processes

## Internet Analytics - Lab 2

---

### 2.2 Network sampling

#### Exercise 2.7: Random walk on the Facebook network

```
In [1]: %config Completer.use_jedi = True
import random
```

```
In [2]: import requests
URL_TEMPLATE = 'http://iccluster028.iccluster.epfl.ch:5050/v1.0/facebook?user={user_
user_id = 'a5771bce93e200c36f7cd9dfd0e5deaa'

def getData(fid):
    url = URL_TEMPLATE.format(user_id=fid)
    response = requests.get(url)
    return response.json()
```

```
In [3]: data = getData(user_id)

age = 0
N= 1000.0
for i in range(int(N)):
    i+=1
    if(i % 100 ==0 ):
        print(i)
        print("average currently : " + str(age/i))
    age += data["age"]
    random.seed(i)
    newFriend = random.randint(0,len(data["friends"])-1)
    data= getData(data["friends"][newFriend])
average = float(age)/N
print(average)
```

```
100
average currently : 32.6
200
average currently : 29.01
300
average currently : 26.61
400
average currently : 27.045
500
average currently : 27.85
600
average currently : 27.20833333333332
700
average currently : 26.187142857142856
800
```

```

average currently : 25.89
900
average currently : 25.11888888888889
1000
average currently : 24.522
24.583

```

## Exercise 2.8

1. Even running the random walk at  $N = 1000$  we get an average around 22 to 26 years of age which is very far from the actual average. 2. This of course depends on randomness but currently it seems people seem to be friends with people of the same age.

In [4]:

```

def noreturn(N=1000.0, weighted = True, reverse = False):
    age = 0
    data = getData(user_id)
    #set of people we already have the age of
    fset = {user_id}
    #saving how many friends one has so we can use weighted sampleing later
    deg = [(user_id ,len(data["friends"]))]
    newF = user_id
    #represents the percentage of times we continue sampleing from current new frien
    #from fset
    theta = 0.95

    age100=0
    for i in range(int(N)):
        if(i%100 == 0 and i!= 0):
            print("last 100 average : " + str(age100/100.0))
            print("current average : " + str(age/i))

            age100=0
            #used to calculate overall age average
            age+= data["age"]
            #used to calculate average over last 100 people
            age100+=data["age"]
            tries=0
            #random resapmling
            random.seed(i)
            if(random.uniform(0,1) >= theta):
                random.seed(i)
                newF= random.sample(fset, k=1)[0]

            while( newF in fset):
                tries +=1
                if(tries > 10):
                    tries= 0
                    if(weighted):
                        deg.sort(key = lambda x : x[1])
                        weights = [j for i,j in deg]
                        if(reverse):
                            weights = weights.reverse()
                        random.seed(i*tries)
                        newF = random.choices([i for i, j in deg],
                                                weights,
                                                k=1)[0]
                    else:
                        newF = resample(fset,i*tries)#if can't fight a friend that hasn
                        # then go back in graph to random point and try new friend
                else:
                    data= getData(newF)
                    random.seed(i*tries)
                    randInt= random.randint(0,len(data["friends"])-1)

```

```

        newF = data["friends"][randInt]
        deg.append((newF, len(data["friends"])))
        fset.add(newF)
        data = getData(newF)
    return age/N
def resample(fset, seed):
    random.seed(seed)
    random.sample(fset, k=1)[0]

print(noreturn(weighted=True, reverse=True))

```

```

last 100 average : 19.13
current average : 19.13
last 100 average : 19.87
current average : 19.5
last 100 average : 20.4
current average : 19.8
last 100 average : 21.72
current average : 20.28
last 100 average : 19.81
current average : 20.186
last 100 average : 19.81
current average : 20.123333333333335
last 100 average : 20.39
current average : 20.161428571428573
last 100 average : 22.93
current average : 20.5075
last 100 average : 21.26
current average : 20.591111111111111
20.513

```

3. The average we got is around 21 using the no return random walk with random resampling, making sure that most of the time we don't count the same people twice and that each of our datapoints is unique hopefully giving us a better overview over the whole network. In addition, when choosing a neighboring node for the random walk proves difficult (over ten tries of choosing a friend returns a friend already visited) then the algorithm chooses another node to retry based on a weighted distribution. The weights of that distribution are based on the amount of friends a user has which hopefully helps explore more of the network. Reversing the weights order, such that people with less friends are more likely to be picked surprisingly didn't have much effect