Υπολογιστική Φυσική Στερεάς Κατάστασης
Networks

Σεϊτανίδου Δήμητρα

21 Ιουλίου 2020

Με τον όρο δίκτυο (network) εννοούμε μία δομή που αποτελείτε από κόμβους και τις συνδέσεις μεταξύ αυτών των κόμβων. Όλα τα δίκτυα που θα μελετήσουμε είναι undirected, που σημαίνει ότι οι συνδέσεις δεν έχουν κατεύθυνση.

Random network

Ένα τυχαίο δίκτυο (random network) είναι ένα δίκτυο στο οποίο οι συνδέσεις ανάμεσα στους κόμβους κατανέμονται με τυχαίο τρόπο. Ένα τέτοιο δίκτυο, σαν και αυτό που θα μελετήσουμε, είναι το Erdős–Rényi. Σε αυτό το τυχαίο δίκτυο η κατανομή του αριθμού των συνδέσεων γνωρίζουμε από τη θεωρία ότι ακολουθεί την κατανομή Poisson.

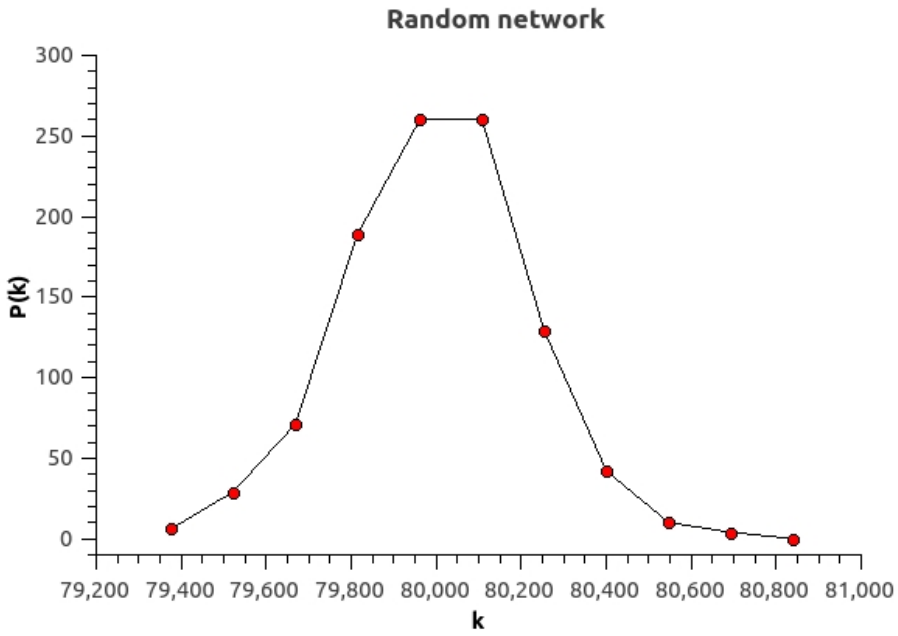
Για να καταγράψουμε τους συνδέσμους του κάθε κόμβου χρησιμοποιούμε μια λίστα γειτνίασης (adjacency list). Αποτελείται από μια πολυδιάστατη λίστα όπου η μία διάσταση είναι μεγέθους N (το πλήθος των κόμβων) και οι υπόλοιπες διαστάσεις είναι ίσες με το πλήθος των συνδέσεων του κάθε κόμβου.

Στον κώδικα που ακολουθεί έχουμε ένα δίκτυο με $N = 1000$ κόμβους και η πιθανότητα να έχει ένας από αυτούς κόμβους σύνδεση με έναν άλλον είναι $p = 0.16$. Καταμετράμε τις συνδέσεις k που έχει ο κάθε κόμβος 1000 φορές και βγάζουμε τον μέσο όρο. Τέλος κάνουμε τη γραφική παράσταση της κατανομής των k (P_k) σε συνάρτηση με το k. Ο κώδικας καθώς και η γραφική παράσταση φαίνονται παρακάτω.

```
#include <fstream>
#include <cstdlib>
#include <vector>
using namespace std;

int main(int argc, char const *argv[]) {
  int i,j,k;
  int m=1000;
  int N1=1000;
  double p=0.16;
  double x,sum;
  vector<vector<int> > list1(N1);
  vector<vector<int> > size1(N1, vector<int>(m,0));

  srand(4372);
  ofstream f1("random.txt");

```

```
  for(k=0;k<m;k++){
    for(i=0;i<N1;i++){
      list1.push_back(vector<int>());
      for(j=i+1;j<N1;j++){
        x = ((double) rand() / (RAND_MAX));
        if(x<p){
          list1[i].push_back(j);
          list1[j].push_back(i);
        }
      }
    }

    for(i=0;i<N1;i++){
      size1[i][k] = list1[i].size();
    }
  }

  for(i=0;i<N1;i++){
    sum = 0;
    for(j=0;j<m;j++){
      sum += size1[i][j];
    }
    f1 << sum/m << endl;
  }

  return 0;
}
```

Σχήμα 1: Erdős–Rényi Network



Από τη θεωρία γνωρίζουμε ότι σε ένα τυχαίο δίκτυο ο κάθε κόμβος θα πρέπει να έχει αριθμό συνδέσεων ίσο με:

$$\frac{N(N-1)}{2}p$$

που στην προκειμένη περίπτωση, για $N = 1000$ και $p = 0.16$, είναι ίσο με $79,920$. Από το διάγραμμα βρίσκουμε ότι η μέση τιμή των συνδέσεων είναι $< \mathbf{k} >= \mathbf{79,998}$. Άρα βλέπουμε ότι οι υπολογισμοί που κάναμε μας έφεραν πολύ κοντά στην θεωρητική τιμή.

Small world

Το μοντέλο small world που θα μελετήσουμε είναι το Watts–Strogatz. Σε αυτό το μοντέλο έχουμε αρχικά ένα ομαλό (regular) δίκτυο στο οποίο κάθε κόμβος είναι ένα άρτιο αριθμό συνδέσεων που μοιράζονται ισάριθμα δεξιά και αριστερά του κάθε κόμβου. Αυτό το δίκτυο θεωρούμε ότι είναι ένα small world δίκτυο με $p = 0$, όπου p είναι η πιθανότητα να ανακατανεμηθεί τυχαία μια σύνδεση. Σε αυτή την άσκηση έχουμε ένα δίκτυο με $N = 1000$ κόμβους και $k = 14$ συνδέσεις.

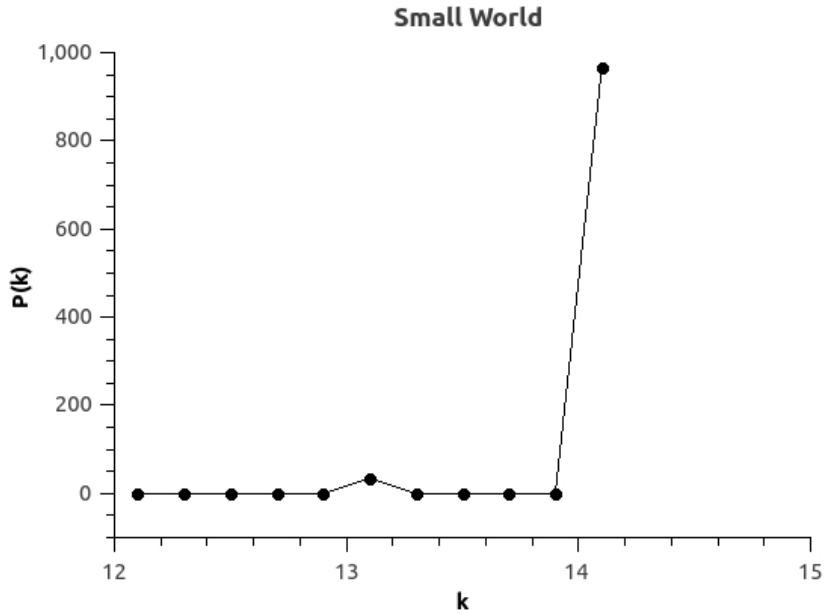
Το δίκτυο το αναπαριστούμε χρησιμοποιώντας έναν πίνακα γειτνίασης (adjacency matrix). Ο πίνακας αυτός έχει διαστάσεις $N \times N$ και οι τιμές του είναι 0 ή 1 ανάλογα αν δεν υπάρχει ή υπάρχει σύνδεση μεταξύ των κόμβων i και j.

Αρχικά ξεκινάμε με ομαλό δίκτυο και έπειτα ανακατανέμουμε τους κόμβους με πιθανότητα $p = 0.20$. Αποθηκεύουμε τις αλλαγές στον πίνακα γειτνίασης και τέλος μετράμε τις συνδέσεις που έχει ο κάθε κόμβος και κάνουμε τη γραφική παράσταση της κατανομής των k (P_k) σε συνάρτηση με το k. Παρακάτω δίνονται ο κώδικας και η γραφική παράσταση.

```cpp
#include <fstream>
#include <cstdlib>
using namespace std;

int main(int argc, char const *argv[]) {
  int i,j,y,sum;
  int N=1000;
  int k=14;
  double p=0.20;
  double x;
  int adjmatrix[N][N];

  srand(4372);
  ofstream f("small-world.txt");

  for(i=0;i<N;i++){
    for(j=0;j<N;j++){
      adjmatrix[i][j] = 0;
    }
  }

  // regular Watts-Strogatz network
  for(i=0;i<N;i++){
    for(j=0;j<N;j++){
```

```
      if(abs(i-j)<=k/2 || abs(i-j)>(N-1-k/2)){
        adjmatrix[i][j] = 1;
      }
      if(i==j){
        adjmatrix[i][j] = 0;
      }
    }
  }

  //small world with probability p
  for(i=0;i<N;i++){
    for(j=0;j<N;j++){
      if(adjmatrix[i][j]==1){
        x = ((double) rand() / (RAND_MAX));
        if(x<=p){
          adjmatrix[i][j] = 0;
          do{
            y = rand()%N;
          } while(y==i);
          adjmatrix[i][y] = 1;
        }
      }
    }
  }

  for(i=0;i<N;i++){
    sum = 0;
    for(j=0;j<N;j++){
      if(adjmatrix[i][j]==1){
        sum += 1;
      }
    }
    f << sum << endl;
  }

  return 0;
}
```

Σχήμα 2: Watts–Strogatz Network



Βλέπουμε ότι οι πιο πολλοί κόμβοι εξακολουθούν να έχουν 14 συνδέσεις και λίγοι έχουν μικρότερο αριθμό συνδέσεων, μετά την ανακατάταξη.

Power law

Τέλος δημιουργούμε ένα δίκτυο νόμου δύναμης. Χαρακτηριστικό αυτού του δικτύου είναι ότι είτε έχουμε πολλούς κόμβους με λίγες συνδέσεις είτε λίγους κόμβους με πολλές συνδέσεις. Η κατανομή των συνδέσεων δίνεται από τη σχέση του νόμου δύναμης

$$P(k) = k^{-\gamma}$$

Εφαρμόζουμε το παραπάνω και αναπαράγουμε τυχαία τον αριθμό των συνδέσεων για τρία διαφορετικά γ: $\gamma = 3$, $\gamma = 2$ και $\gamma = 2.5$. Για να πάρουμε τυχαίο αριθμό από κατανομή νόμου δύναμης χρησιμοποιούμε την παρακάτω σχέση:

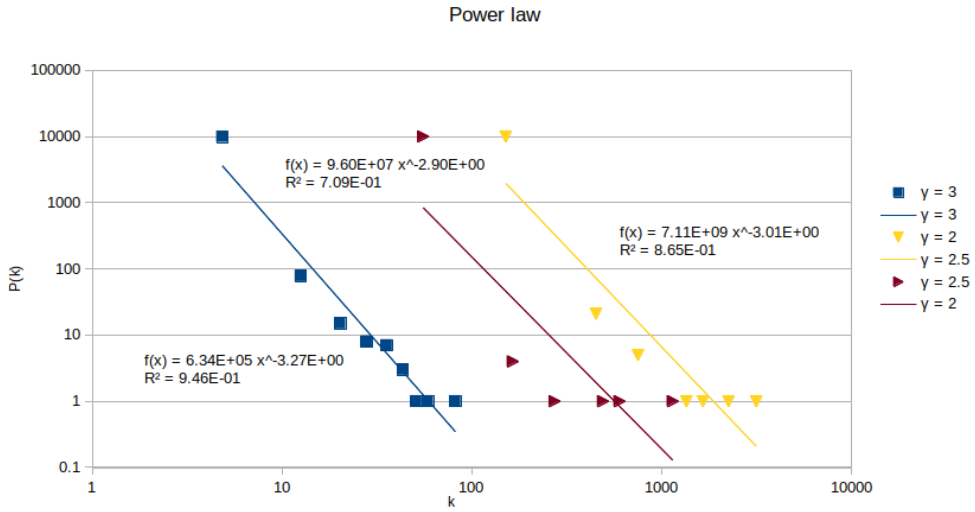
$$y = [(y_{max}^{1-\gamma} - (y_{min}^{1-\gamma})x + y_{min}^{1-\gamma}]^{\frac{1}{1-\gamma}}$$

όπου x τυχαίος αριθμός από ομοιόμορφη κατανομή, $y_{min} = 1$ και $y_{max} = N - 1$. Η παραπάνω σχέση προέρχεται από το θεώρημα μετασχηματισμού ολοκληρώματος πιθανότητας. Ο κώδικας και η γραφική παράσταση των κατανομών δίνεται παρακάτω.

```
#include <fstream>
#include <cstdlib>
#include <cmath>
using namespace std;

int main(int argc, char const *argv[]) {
  int i,j,z;
  double x,y;
```

```
  int N = 10000;
  double g1 = 3.0;
  double g2 = 2.0;
  double g3 = 2.5;
  double ymin = 1.0;
  double ymax = N-1.0;

  srand(4372);
  ofstream f1("power-law3.txt");
  ofstream f2("power-law2.txt");
  ofstream f3("power-law25.txt");

  // gamma = 3
  for(i=0;i<N;i++){
    x = ((double) rand() / (RAND_MAX));
    y = pow((pow(ymax,1.0-g1)-pow(ymin,1.0-g1))*x + pow(ymin,1.0-g1)
    ,1.0/(1.0-g1));
    z = floor(y);
    f1 << z << endl;
  }

  // gamma = 2
  for(i=0;i<N;i++){
    x = ((double) rand() / (RAND_MAX));
    y = pow((pow(ymax,1.0-g2)-pow(ymin,1.0-g2))*x + pow(ymin,1.0-g2)
    ,1.0/(1.0-g2));
    z = floor(y);
    f2 << z << endl;
  }

  // gamma = 2.5
  for(i=0;i<N;i++){
    x = ((double) rand() / (RAND_MAX));
    y = pow((pow(ymax,1.0-g3)-pow(ymin,1.0-g3))*x + pow(ymin,1.0-g3)
    ,1.0/(1.0-g3));
    z = floor(y);
    f3 << z << endl;
  }


  return 0;
}
```

Σχήμα 3: Κατανομή νόμου δύναμης



Στο σχήμα φαίνεται και η τιμή του γ που προκύπτει από κάθε προσομοίωση. Για θεωρητική τιμή 3 έχουμε $\gamma = 3.27$, για θεωρητική τιμή 2 έχουμε $\gamma = 2.90$ και τέλος θεωρητική τιμή 2.5 έχουμε $\gamma = 3.01$.