Networks Structure and Dynamics

Sorbonne Université
Master d'Informatique spécialité Réseaux

Examination

February 6^{th} 2018

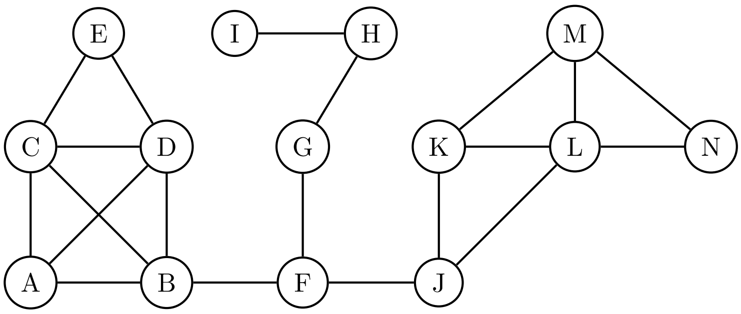
Maximilien Danisch, Fabien Tarissan and Lionel Tabourier

The exam is 2 hours long. Documents are allowed, but no electronic device is allowed.
The exam is graded on 60 points: Part 1 is graded on 40 points and Part 2 on 20 points.

Part 1: Networks Structure

Exercice 1 — *Simple measures on a small graph*

Consider the following graph.



Q1. Draw the degree distribution of the graph.

Q2. Compute the number of triangles and the transitivity ratio.

Q3. Compute the clustering coefficient of each node and the average clustering coefficient.

Q4. Compute the core value of each node.

Q5. Give a possible BFS and DFS ordering of the nodes starting from node L.

Q6. Which node has the highest betweenness centrality[1]? Justify briefly.
Compute the betweenness centrality of nodes B, F, I and L.

Q7. The closeness centrality of a node u is defined as follows:

$$closeness(u) = \sum_{v \neq u} \frac{1}{d(u,v)},$$

where $d(u,v)$ is the distance between u and v.
Compute the closeness centrality of nodes B, F, I and L.

[1] The betweenness centrality of a node is the number of shortest paths passing by the node.

Exercice 2 — *Real-world graph algorithmics*

Q1. Describe an algorithm that lists all triangles in the input graph in time $O(\delta \cdot (m+n))$ where δ is the degeneracy of the graph (i.e., the core value of the graph), m is the number of edges and n the number of nodes. Explain briefly why this algorithm is efficient on large sparse real-world graphs.

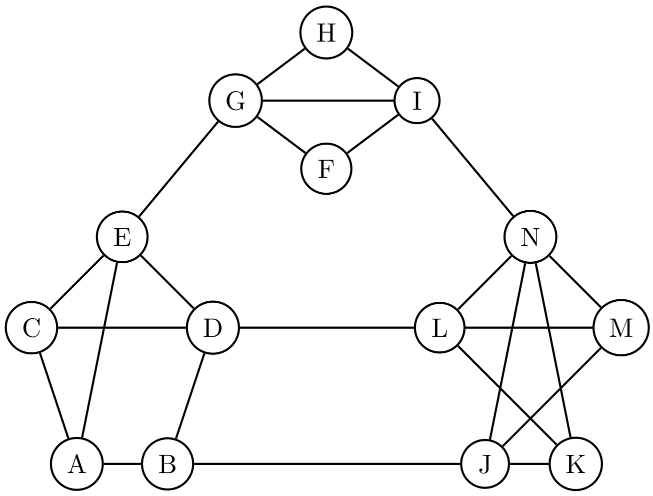
Q2. Describe an algorithm that computes a lower bound of the diameter of the input graph. The lower bound should be empirically good for large sparse real-world graphs and the algorithm should run in linear time. Give some intuition of why the lower bound is empirically good.

Q3. Describe an algorithm that computes a 2-approximation of the densest subgraph[2] in linear time. Justify briefly why the asymptotic running time consumption is linear.

For each description, make sure that you detail all important steps. You can include some pseudocodes.

Exercice 3 — *Community detection*

Consider the following graph.



Q1. Explain why the following partition into communities makes sens and compute its modularity $\{\{A, B, C, D, E\}, \{F, G, H, I\}, \{J, K, L, M, N\}\}$.

Q2. Give a high level pseudocode of a divisive community detection algorithm based on edge triangularity[3] (for this question we do not pay attention to the running time of the algorithm).

Q3. Simulate an execution of the algorithm you suggested in Question 2 on the graph, draw the associated dendrogram and give the partition returned by the algorithm.

Q4. Assume Algorithm 1 outputs the partition $\{\{A, C, D, E\}, \{F, G, H, I\}, \{B, J, K, L, M, N\}\}$ and Algorithm 2 outputs the partition $\{\{A, C, E, G, H\}, \{B, D, F, I, J, K, L, M, N\}\}$. Which output do you think is the best? Justify briefly. Suggest a relevant criterion to score the two algorithm outputs by comparing them to the ground-truth partition given in Question 1. Compute these two scores.

[2] Average degree density

[3] The triangularity of an edge is defined as the number of triangles the edge belongs to.

Part 2: Networks Dynamics

In this exercise, we consider the evolution of contact networks. For this purpose, we consider *edge-markovian evolving graphs* (denoted EMEG in the following).

We remind that in the EMEG model, the state G_{t+1} of graph G at instant $t+1$ is determined by the state G_t of the graph at instant t, the appearance probability p of non-existing links and disappearance probability d of existing links.

In the following, we represent the contact trace as a sequence of 4-tuple: `x y ts te`, where x and y are the identifiers of the nodes involved in the contact, t_s indicates the starting time of the contact between x and y and t_e its end. We assume in the following that contacts are undirected and that for two successive contacts involving the same nodes, the trace is sorted according to starting time (t_s).

Exercice 4 — emeg *convergence*

We are interested in the state of the graph to which EMEG model converges when t goes to infinity.

Q1. What is the state of the graph to which EMEG model converges when $p = 0$ and $d > 0$? Same question when $p > 0$ and $d = 0$.

Q2. What is the state of the graph to which EMEG model converges for any given value of p and d? Is it coherent with answers for **Q1.**? Justify.

Exercice 5 — *Structure of the graph*

We consider the following trace of the contacts in the network through time, starting at time 0 and ending at time 10:

```
a b 1 1
a b 5 6
a b 8 10
a c 1 4
a c 6 8
a d 5 9
a e 2 2
a e 8 8
a f 4 10
b c 0 10
c d 2 4
c d 6 7
d e 0 4
d e 6 7
d e 9 10
e f 5 8
e f 10 10
```

Q1. Draw the graph at each time step.

Q2. Draw a curve showing the average degree of the graph at each time step.

Exercice 6 — *Inter-contacts* We want to study the average inter-contact duration in this evolving graph. Let us remind that the *average inter-contact duration* between two nodes i and j is the average duration between two successive contacts between i and j. If i and j have less than 2 contacts, then the value is -1.

Q1. What is the inter-contact duration between nodes (a, b), (a, c) and (a, d) in the former trace?

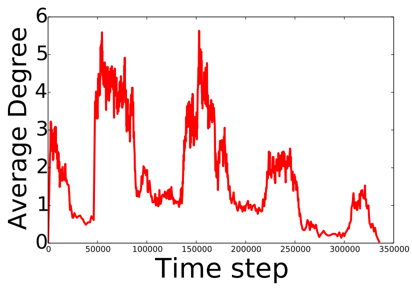
Q1. Consider the following algorithm.

Algorithm 1: Compute the average inter-contact duration between all pairs of nodes

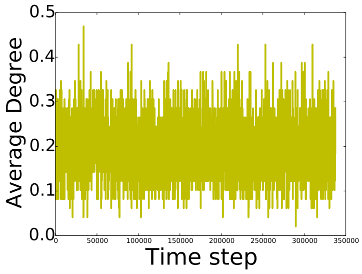
```
Input : F : file containing the contacts
        N : number of nodes
Output : Pair (Time, avgtime) where
             Time[i][j] is the average inter-contact duration between i and j.
             avgtime is the average inter-contact duration in the network.
begin
    Last ← Array of N × N cells   // time of the last contact between 2 nodes
    NB ← Array of N × N cells   // number of contacts between 2 nodes
    Sum ← Array of N × N cells   // sum of all inter-contact duration between 2 nodes
    Time ← Array of N × N cells   // average inter-contact duration between 2 nodes

    for i from 1 to N do
        for j from 1 to N do
            Last[i][j] = -1          // no contact at initial step
            Time[i][j] = -1          // no inter-contact duration at initial step
            NB[i][j] = 0             // no contact at initial step
            Sum[i][j] = 0            // Sum = 0 at initial step
        end
    end

    while F not empty do
        (n1, n2, td, tf) ← NextLine(F)
        ...                          // update of Sum
        ...
        ...
        ...                          // update of Last
        ...                          // update of NB
    end

    // compute array Time
    for i from 1 to N do
        for j from 1 to N do
            ...          // compute the average inter-contact duration between i and j
            ...
            ...
        end
    end

    avgtime = 0          // average inter-contact duration of the network
    ...                  // compute the average inter-contact duration of the network
    ...
    ...

    Return (Time, avgtime)
end
```

1. Complete lines 21 to 28 of the algorithm such that the arrays Last, NB and Sum are correct.
2. Complete lines 31 to 37 such that the average inter-contact duration between all pairs of nodes are correct.
3. Complete lines 40 such that it computes the average inter-contact duration in the network.

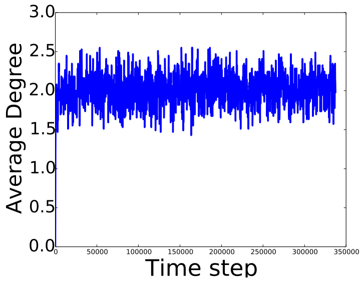
Exercice 7 — *Random networks* We consider now another trace (not shown here) describing the evolution of the contacts in a network of 100 nodes. We compute the average degree of the network at each time step and show the evolution of this value in the following plot.



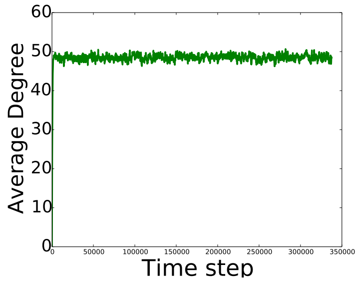
We now generates 3 random evolving graphs using EMEG model with respectively ($p = 0.001, d = 0.001$), ($p = 0.0001, d = 0.005$), and ($p = 0.001, d = 0.5$). The evolution of the average degree though time for the 3 generated graphs is given in the following plots.



(a)



(b)



(c)

Q1. Associate the parameter setting to each figure. Justify.

Q2. Which of the generated graph is the closest to the real graph? Justify.

Q3. Comparing the plots for the real graph to the one of the random network, what conclusion can be drawn regarding the relevance of the EMEG model to capture the dynamics of the real network? Justify.