R exerices
Igraph

Gilles Tredan

Abstract

Graph Creation and sample statistics

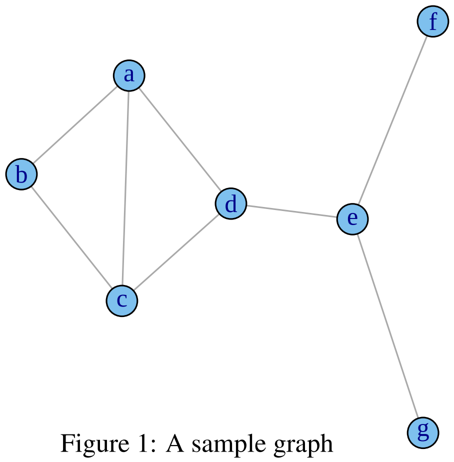

Figure 1: A sample graph

- create an empty undirected graph using the command `graph.empty`
- create the sample graph given Figure 1 using the `graph.formula` command.
- use the `plot` function to represent this graph
- igraph also provides different ways to generate random graphs. Generate the following graphs
  - `ga=barabasi.game(500,power=2.5,directed=F,m=5)`
  - `gb=erdos.renyi.game(496,0.016,directed=F)`
  - `gc=watts.strogatz.game(1,500,4,.1)`
- inspect their degree distribution `degree(g)` and plot the 3 degree distributions together.
- Use lapply to collect information about these graphs: `diameter`, `average.path.length`, and clustering coefficient (`transitivity`).
- Plot the `ga`'s degree distribution on a log-log plot. Use `power.law.fit`.
- How does the powerlaw coefficient evolves as you increase the graph size ? Create a function that returns the powerlaw exponent as a function of the Barabasi-Albert graph size. Plot its evolution for graphs of size $[500, 10000]$. How does the KS score evolves ?

- Use `replicate` to construct a more robust observation: replicate each estimation of the alpha parameter 10 times and represent the distribution obtained.
- Use ddply to speed up the evaluation of these last functions.

Graph attributes and manipulation

- gml (Graph Modelling Language) is a pretty standard format to exchange graphs.
- Use `read.graph` to read `lesmis.gml`, a graph representing the co-appearance network of characters in Hugo's book "Les misrables".
- igraph provides a bunch of different layouts: `layout.auto layout.bipartite,layout.circle layout.drl,layout.fruchterman.reingold layout.fruchterman.reingold.grid,layout.graphopt layout.grid,layout.grid.3d layout.kamada.kawai,layout.lgl,layout.mds,layout.merge layout.norm,layout.random layout.reingold.tilford,layout.show layout.sphere,layout.spring,layout.star,layout.sugiyama,layout.svd`. Try fruchterman.reingold and circle layouts using `layout=` option in plot. Note: the documentation of the called `plot` function for igraph objects is `?plot.igraph`
- use `edge.betweenness.community` to find the communities in Les misrables' graph.
- Use the `class` function to identify the class of the result provided by `edge.betweenness.community`. Maybe there is a plot function that can represent such object ? Plot the communities of Les misrables.
- Bonus: use `microbenchmark` to estimate the asymptotic complexity of `edge.betweenness.community` by running it on networks of increasing size. Which parameters impact the runtime of this method (besides graph size) ?

Visualizing Graphs as Sparse Matrixes

- Graphs can be represented by their adjacency matrix. The adjacency matrix A of a n-vertices graph G is a $n \times n$ matrix where $A[i, j] = 1$ if $(i, j) \in E(G)$ (*i.e.* there exists an edge between nodes i and j), 0 else.
- Create a function that represents a graph's adjacency matrix using `graph.adjacency`, `as.matrix`, `melt`, and `geom_raster`. Use it to represent `ga`,`gb` and `gc`