Simple Network Analysis with MatLab

Gergana Bounova
ESD.342
February 23, 2006
MatLab Basics

- Official MathWorks tutorial:
- List of all MatLab functions
  - Search by name, topic, description
- MatLab prompt:
  >> date
  >> help 'what'
  >> lookfor 'something'
  >> help lookfor
  LOOKFOR Search all M-files for keyword. See also dir, help, who, what, which.
  >> diary (filename, on, off)
  >> load mydata.mat
  >> type filename.m (.txt)
- Loading data: ExcelLink, etc

```matlab
% Checks whether a matrix is symmetric
% Gergana Bounova, November 2, 2005
function [issym] = issymmetric(mat)
%  inputs: (square) matrix
%  outputs: boolean variable, (0,1)
%  check whether mat(i,j)==mat(j,i) for all i,j
n=length(mat);
for i=1:n
    for j=i+1:n
        if not(mat(i,j)==mat(j,i))
            issym = false;
            return
        end
    end
end
```

Gergana Bounova
ESD.342
Feb 23, 2006
Working Example: Bike

Graph Representation in MatLab

- Depends on what you are going to do! Computation, extracting data/properties, visualization…
- Adjacency matrix $A$
  - node by node ($n \times n$), if $i$ and $j$ are connected $A(i,j)=1$, otherwise $A(i,j)=0$; for multiple edges $A(i,j)=2,3,…$
  - $\text{sum}(A) =$ graph degree sequence (self-loops give an exception)
- Incidence matrix $C$
  - node by edge ($n \times m$), if node $i$ is an endpoint for edge $j$, then $C(i,j)=1$, otherwise $C(i,j)=0$
  - $\text{sum}(C) = [2 \ 2 \ 2 \ …2]$ – every edge has 2 endpoints
  - Cool formula: $A = C \times C^T - 2I$
- List: every node points to the nodes it’s connected to
- Text: for other programs input (Pajek)
MatLab Code I

>> clear all

>> % Load bike data
>> load bike_data

>> who

>> size(adj_bike)
>> size(deg_bike)
>> deg_bike – sum(adj_bike)

>> % Graph Representations

>> inc = adj2inc(adj_bike);
>> size(inc)
>> sum(inc)  % should give a vector of 2s

>> % num_edges = sum(sum(adj_bike))/2 for an undirected graph
>> % should be same as second dimension of incidence matrix
>> numedges = num_edges(adj_bike)

>> % Lists
>> str = adj2str(adj_bike);
>> % number of children and number of degrees should be the same
>> str(1).child
>> degrees(1)

Gergana Bounova
ESD.342
Feb 23, 2006
Graph Diagnostics

- issymmetric.m: $A = A^T$?
- issimple.m: are there self-loops or double edges?
- isdirected: $A = A^T$?
- isconnected.m: is there a path from every node to every other node?
- issparse.m: $k << n \times m$, $k$ – # of non-zero entries
- All of the above return Boolean variables
MatLab Code II

```matlab
>> issymmetric(adj_bike)
an =
1
>> issimple(adj_bike)
an =
1
>> isdirected(adj_bike)
an =
0
>> isconnected(adj_bike)
an =
1
```

bikenet (Pajek)
Graph Properties

**LOCAL**
- shortest path (i-j)
- betweenness (i)
- degree (ave, max, in-out)
- clustering coefficient (i)
- harmonic path length (i-j)
- number of k-neighbors (i)

**GLOBAL**
- mean path length
- betweenness distribution
- degree distributions
- mean clustering coeff.
- mean harmonic path
- k-neighbors distribution
- diameter
MatLab Code III

>> degree_dist(adj_bike)
ans = 0.3933

>> clust_coeff(adj_bike)
ans = 0.3933

>> diagnose_powerlaw?
ans = 0.9973

>> ave_path_length(adj_bike)
ans = 0.9973

>> diameter(adj_bike)
ans = 9
Graph Construction/Structure

- min spanning tree
- connected components
- dual graphs (dual.m)
- subgraph and motifs
- growth models
  - Random, preferential, min-cost, max-span
  - k-regular graphs
  - random_graph.m
MatLab Code IV: Random Graphs

>> adj = random_graph(10)
>> adj = random_graph(10,0.1)
>> adj = random_graph(10,0.1,20)
>> adj = random_graph(10,0.1,20,’normal’)
>> adj = random_graph(10,0.1,20,’custom’,@mypdf)

>> degrees = [3 1 1 1];
>> adj = graph_from_degrees(degrees)
  
  • 2 versions
  • ~50% success rate (in producing simple graphs)
Bike Distribution Random Graphs
Network Visualization - Pajek

- `draw_circ_graph.m`

- **MatLab-Pajek:**
  - `>> adj2pajek(adj,’graph.net’)`
  - Open graph.net in Pajek
  - `Gtrl + G`
Specialized (Engineering) Metrics

- **Robustness:**
  - Topological: if you knock out x% of nodes/edges, how many % survive
  - Functional: by how much (%-wise) function degrades

- **Span, coverage, geometry considerations**

- **Cost, performance**
  - total launch mass (space), network throughput, comm link costs, number of cities served
Off the Shelf (MatLab Functions)

- dist, pdist, linkage, dendrogram, cluster, kmeans
- Simulink (circuits), neural net toolbox, comp bio toolbox
- Statistics toolbox: distributions
- List of all MatLab functions
  [http://www.mathworks.com/support/functions/alpha_list.html](http://www.mathworks.com/support/functions/alpha_list.html)

Gergana Bounova
ESD.342
Feb 23, 2006
Your Own Toolbox

- Write functions
- Save them in the same directory
- Set workpath (accessible from anywhere)
- Have Fun!

Gergana Bounova
ESD.342
Feb 23, 2006
Appendix: tutorial.m

% MatLab tutorial script
% Gergana Bounova, February 22, 2006

clear all

% Load bike data
load bike_data
who

size(adj_bike)
size(deg_bike)

% Graph Representations ********************************************
inc = adj2inc(adj_bike);
size(inc)
sum(inc) % should give a vector of 2s

% num_edges = sum(sum(adj_bike))/2 for an undirected graph
% should be same as second dimension of incidence matrix
numedges = num_edges(adj_bike)

% Lists
str = adj2str(adj_bike);
% number of children and number of degrees should be the same
str(1).child.degrees(1)

% Graph Diagnostics ***********************************************
issymmetric(adj_bike)
issimple(adj_bike)
isdirected(adj_bike)
isconnected(adj_bike)

% Global metrics
degree_dist(adj_bike)
clust_coeff(adj_bike)
diameter(adj_bike)
average_path_length(adj_bike)
betweenness(adj_bike)
harmonic_path_length(adj_bike)

% Graph construction **********************************************
random_graph(10)
random_graph(10,0.1,20)
random_graph(10,0,0,'normal')
random_graph(10,0,0,'custom',@mypdf)
degs = [3 1 1 1];
random_graph(10,0,0,'custom',@mypdf,degs)
adj_bike1 = graph_from_degrees(degs);

% Graph visualization **********************************************
adj2pajek(adj_bike,'bike.net');