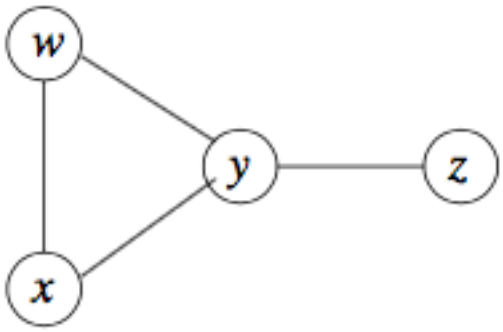
IS622 Week 14 Exercises

Brian Chu | Nov 24, 2015

Exercise 10.5.1: Suppose graphs are generated by picking a probability p and choosing each edge independently with probability p, as in Example 10.21. For the graph of Fig. 10.20, what value of p gives the maximum likelihood of seeing that graph? What is the probability this graph is generated?



Nodes: 4
Edges: 4

Possible edges: $\binom{4}{2} = 6$

Prob(Fig. 10.20) = $p^4(1-p)^2$

Set derivative = 0 to find MLE of p:

$4p^3(1-p)^2 - 2p^4(1-p) = 0$
$2p^3(1-p)(2(1-p)-p) = 0$

Set right side, 2(1-p) - p = 0, to find maximum (i.e. p= 0,1 produce minimums)
$2 - 2p - p = 0$
$3p = 2$
$p = \frac{2}{3}$

Probability of graph:
$(\frac{2}{3})^4(1-(\frac{2}{3}))^2$

$=\frac{16}{81} * \frac{1}{9}$

$=\frac{16}{729} = 0.0219$

Exercise 10.7.1 : How many triangles are there in the graphs:

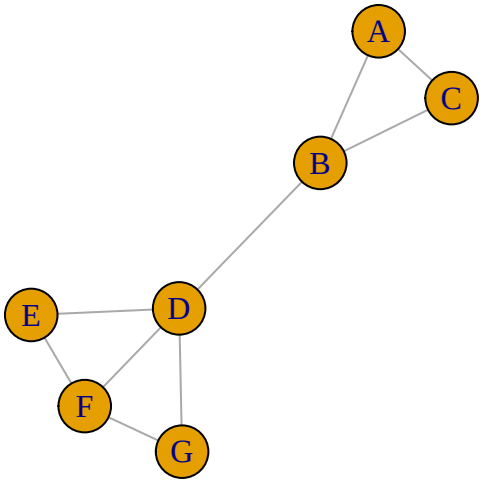
First I define a function to count unique triangles given the edges as input

```
tricount <- function(edges, node_length) {

  # Sort each edge and then vector of edges
  edges <- do.call(rbind, lapply(edges,
                                 function(x) paste(sort(unlist(strsplit(x,"|", fixed=TRUE))),
                                                   collapse="")))
  edges <- sort(edges)

  # Store all unique triangles
  all_tris <- character()

  # Find triangles
  for (i in 1:length(edges)) {

    # Define nodes 1 and 2 and edges starting from them
    n1 <- substr(edges[i], 1, node_length)
    n2 <- substr(edges[i], 1+node_length, node_length*2)
    n1edges <- edges[substr(edges,1, node_length)==n1]
    n2edges <- edges[substr(edges,1, node_length)==n2]

    # Find matching third nodes
    n3 <- character()
    for (x in n1edges) {
      candidate <- substr(x,1+node_length, node_length*2)
      if(candidate %in% substr(n2edges,1+node_length, node_length*2)) {
        n3 <- c(n3, candidate)
      }
    }

    # If matching third node found, count as triangle
    if(length(n3)>0) {
      triangle <- sort(paste(n1,n2,n3, sep='-'))
      all_tris <- c(all_tris, triangle)
    }
  }

  return(all_tris)
}
```

(a) Figure 10.1.

Note: I used iGraph to plot the figures but my defined function above to count the triangles

```
library(igraph)
g1 <- graph_from_literal(A-B, A-C, B-C, B-D, D-E, D-F, D-G, E-F, F-G)
plot(g1, vertex.size=25)
```




```
e1 <- as_ids(E(g1))
t1 <- tricount(e1,1) #MANUAL FUNCTION, not an iGraph one
print(t1)
```

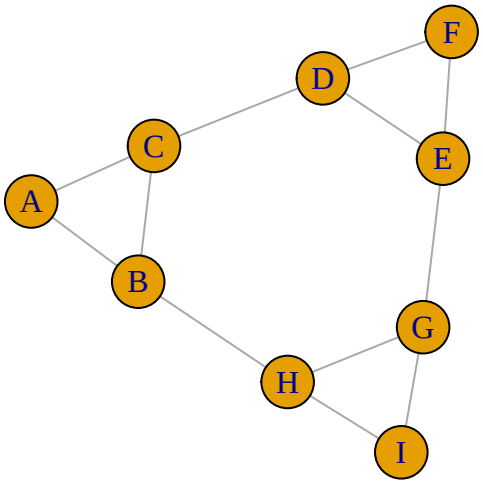
```
## [1] "A-B-C" "D-E-F" "D-F-G"
```

```
print(paste("Number of unique triangles:", length(t1)))
```

```
## [1] "Number of unique triangles: 3"
```

(b) Figure 10.9.

```
g2 <- graph_from_literal(A-B, A-C, B-C, C-D, D-E, D-F, E-F, B-H, H-I, H-G, I-G, G-E)
plot(g2, vertex.size=25)
```



```
e2 <- as_ids(E(g2))
t2 <- tricount(e2,1) #MANUAL FUNCTION, not an iGraph one
print(t2)
```

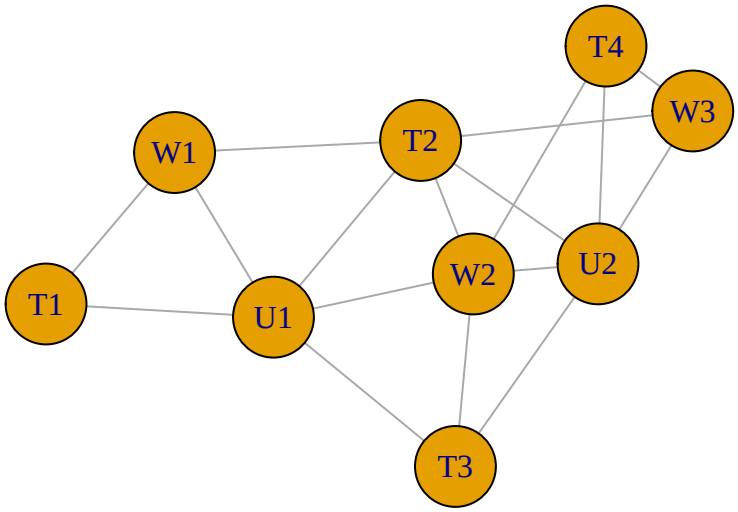
```
## [1] "A-B-C" "D-E-F" "G-H-I"
```

```
print(paste("Number of unique triangles:", length(t2)))
```

```
## [1] "Number of unique triangles: 3"
```

(c) Figure 10.2.

```
g3 <- graph_from_literal(U1-T1, U1-T2, U1-T3, U2-T2, U2-T3, U2-T4,
                         T1-W1, T2-W1, T2-W2, T2-W3, T3-W2, T4-W2, T4-W3,
                         U1-W1, U1-W2, U2-W2, U2-W3)
plot(g3, vertex.size=25, asp=0.65)
```




```
e3 <- as_ids(E(g3))
t3 <- tricount(e3,2) #MANUAL FUNCTION, not an iGraph one
print(t3)
```

```
## [1] "T1-U1-W1" "T2-U1-W1" "T2-U1-W2" "T2-U2-W2" "T2-U2-W3" "T3-U1-W2"
## [7] "T3-U2-W2" "T4-U2-W2" "T4-U2-W3"
```

```
print(paste("Number of unique triangles:", length(t3)))
```

```
## [1] "Number of unique triangles: 9"
```