

## distance\_w

Last Updated Monday, 03 August 2009

The shortest path length, or geodesic distance, between two nodes in a binary network is the minimum number of steps you need to make to go from one of them to the other. This distance is the quickest connection between nodes when all ties are the same. However, in a weighted network, all ties are not the same.

This function takes three arguments:

- edgelist
- directed: logical, if the edgelist is directed. Default: NULL, the function will check whether the edgelist is directed
- gconly: logical, if the measure should only be calculated for the giant component. Default: TRUE

If we have the following network:

The geodesic distance would be (if we dichotomise it with a cut-off parameter of greater than 0):

However, the connection between node A and node D might not be the most efficient path for transferring information or diseases as the connection is very weak. The distance\_w-function in tnet takes the weights into consideration by applying knowledge from electrical circuits. The function works as follows:

- We inverse the values of the weights, so the highest value of weights (most efficient connection) has the lowest value (lowest cost):

- The inverted values of the weights are then normalise with the average weight in the network (6.25 in this example). The path from {A,B,C,D} is then  $47/120 \times 6.25$  which is approximately 2.45 compared to the direct path which  $1/1 \times 6.25 = 6.25$ .

- Finally, we find the paths with the lowest costs between nodes, and sum their values (similar to Dijkstra's algorithm: E. W. Dijkstra: A note on two problems in connexion with graphs. In Numerische Mathematik, 1, 1959, S. 269&ndash;271). The average path length from each node to the others can be calculated. The average of the averages is comparable to the geodesic measure. This value will be higher as more shorter paths are not chosen in favour of longer, more efficient paths:

```
Example: Sample data## Load tnet
library(tnet)
## Define sample data
sample <- rbind(
  c(1,2,8),
  c(1,4,1),
  c(2,1,8),
  c(2,3,6),
  c(3,2,6),
  c(3,4,10),
  c(4,1,1),
  c(4,3,10))## Get distance matrix (non-normalised, and normalised)
distance_w(sample)
  1 2 3 4
1 NA 0.1250000 0.2916667 0.3916667
2 0.1250000 NA 0.1666667 0.2666667
3 0.2916667 0.1666667 NA 0.1000000
4 0.3916667 0.2666667 0.1000000 NA
distance_w(sample)*mean(sample[,3])
  1 2 3 4
1 NA 0.781250 1.822917 2.447917
2 0.781250 NA 1.041667 1.666667
3 1.822917 1.041667 NA 0.625000
4 2.447917 1.666667 0.625000 NA
## Find mean value
mean(distance_w(sample)*mean(sample[,3]), na.rm=TRUE)
1.397569
```

To cite this function, please use: Opsahl, T. (2007). tnet: Software for Analysis of Weighted and Longitudinal networks, version <version used>. Retrieved on <current date>, from <http://opsahl.co.uk/tnet/>