

1 Attribute growth

Consider an initial graph $G_0 = (V_0, E_0)$ have vertex set V_0 and edge set E_0 . We will have a set of variable length vectors A where A_v represents the vector of attributes of vertex v . We will additionally define a threshold S representing the proportion of attributes that two vertices must share in order for an edge to exist. To determine whether an edge exists between u and v where possibly $|A_u| \neq |A_v|$ we will compare only the first $\min(|A_u|, |A_v|)$ attributes. Next we define C , a vector of the count of attributes at added as well as B a vector of possible attributes. Then we define I_i , the number of new vertices introduced at step i . Finally we define L , the length of the random walk new vertices take when added to the graph.

In order to track how long vertices have existed in the network we will denote the time they have existed as T_v for some vector v .

To create the initial E_0 we assign first each $v \in V_0$ an attribute in $\{1, \dots, B_0\}$. Next we add edges for pairs that meet threshold S . In this initial case for any non-zero S we only have edges between vertices u, v where $A_u[0] = A_v[0]$

In all subsequent iterations k we first add C_{T_v} attributes to all $v \in V_{k-1}$. We then add I_k vertices and assign them each one attribute with value in $\{1, \dots, B_0\}$. Next for each new vertex u we randomly select a vertex $v \in V_{k-1}$ s.t. $A_u[0] = A_v[0]$ and take a random walk of length L . At each step in the walk we determine if the edge existance threshold is met and if so add an edge. If there is no suitable initial vertex we add this new vertex as an isolated component in the graph. After all new vertices have been added we check all existing edges and remove them if they no longer meet the threshold. Denote this new total set of vertices as V_k and new total set of edges as E_k .

Note that it is possible for an edge that was removed for not meeting the threshold can meet it in a future iteration.

A more formal definition follows below:

Algorithm 1: Network formation by attribute growth

```
let  $E_0 = \emptyset$  and an initial vertex set  $V_0$ ;
let  $G_0 = (V_0, E_0)$ ;
let  $S$  denote the threshold for edge preservation;
let  $B$  define the sequence of possible attributes;
let  $I$  be the sequence defining the number of new vertices at each iteration;
let  $C$  be the sequence defining the number of new attributes gained by each vertex in each iteration of it's
  existence;
let  $T$  be the vector of how many iterations a vertex has existed;
for  $v \in V_0$  do
  | IID assign  $C_0$  elements to  $A_v$  from  $\{1, \dots, B_0\}$ ;
end
for  $u, v \in V_0$  do
  | if  $A_u[0] = A_v[0]$  then
  |   |  $E_0 = E_0 \cup \{uv\}$ ;
  |   end
end
let  $i = 1$  denoting the iteration number;
while Network grows do
  | create a set of new vertices  $V_{i'}$  of size  $I_i$ ;
  | let  $E_i = \emptyset$  for  $v' \in V_{i'}$  do
  |   | IID assign  $A_{v'}[0] \in \{1, \dots, B_0\}$ ;
  |   end
  | for  $v \in V_i$  do
  |   | IID assign  $C_{T_v}$  elements to  $A_v$  from  $\{1, \dots, B_{T_v}\}$ 
  |   end
  | for  $v' \in V_{i'}$  do
  |   | Select  $L$  vertices via a random walk seeded at a random vertex  $u'$  s.t.
  |   |    $\frac{|\{A_{u'}[j] | A_{u'}[j] = A_{v'}[j] \text{ for } j \in \min(|A_{u'}|, |A_{v'}|)\}|}{\min(|A_{v'}|, |A_{v'}|)} \geq S$ ;
  |   | let  $E_i = E_i \cup \{v'w' | \frac{v'w' | A_{v'}[j] = A_{w'}[j] \text{ for } j \in \min(|A_{v'}|, |A_{w'}|)}{\min(|A_{v'}|, |A_{w'}|)} \geq S\}$ ;
  |   end
  |  $V_i = V_i \cup V_{i'}$ ;
  | for  $u, v \in V_i$  do
  |   | if  $\frac{|\{A_u[j] | A_u[j] = A_v[j] \text{ for } j \in \min(|A_u|, |A_v|)\}|}{\min(|A_u|, |A_v|)} \geq S$  then
  |   |   |  $E_i = E_i \cup \{uv\}$ ;
  |   |   end
  |   end
  | end
end
```
