Prediction Of Arrival Of Nodes In A Scale Free Network

Vijay Mahantesh SM
intern
Indian Statistical Institute
Chennai, India
vijaym123@gmail.com

Vijesh M
Intern
Indian Statistical Institute
Chennai, India
mv.vijesh@gmail.com

Shruthi Nayak
Intern
Indian Statistical Institute
Chennai, India
rn.shruthi@gmail.com

Nikitha Shenoy
Intern
Indian Statistical Institute
Chennai, India
nikithashenoyk@gmail.com

ABSTRACT

Most of the networks observed in real life obey power-law degree distribution. It is hypothesized that the emergence of such a degree distribution is due to preferential attachment of the nodes. Barabasi-Albert model is a generative procedure that uses preferential attachment based on degree and one can use this model to generate networks with power-law degree distribution. In this model, the network is assumed to grow one node every time step. After the evolution of such a network, it is impossible for one to predict the exact order of node arrivals. We present in this article, a novel strategy to partially predict the order of node arrivals in such an evolved network. We show that our proposed method outperforms other centrality measure based approaches. We bin the nodes and predict the order of node arrivals between the bins with an accuracy of above 80%.

General Terms

Complex Network Theory

Keywords

preferential attachment, scale-free networks, node-arrival ordering, node aging

1. INTRODUCTION

Real world networks such as biological, social and technological networks are the products of an evolutionary process. These networks are generally classified as Scale Free Networks (SFN) by nature. SFNs are a class of networks in which degree distribution follows Power Law. Generative models such as Duplicate-Mutation, Forest Fire and Preferential Attachment [1] have been proposed to synthesize SFNs. The synthesis of dynamic SFNs involves a continuous addition of new nodes to the existing network. The behavior of each new node depends on the generative model being used. It is intersting to study how nodes get assembled in complex network. Given the snapshot of a dynamic network, is it possible to probabilistically predict the evolutionary sequence of the nodes in the network?

We propose a method that predicts the order of arrival of nodes in the given network, modeled and synthesized using a specified generative model. This approach first computes a vertex ranking of the given network based on a ranking methodology. We then synthesize several such networks using the generative model that was used in the construction of the given network. It is important to note that the order of arrival of nodes in the synthesized networks is known. The same ranking methodology is applied to compute the vertex ranking for each of the synthesized networks. The nodes in the given network are mapped to the nodes in a synthesized network, according to a bijection function between the vertex rankings. We then predict the probable order of arrival of nodes in the given network, based on the bijective mapping and the order of arrival of nodes in the synthesized network. This method of mapping, over several such synthesized networks, ssociates a probability with every pair of vertices. This probability denotes the arrival order of vertices in the corresponding vertex pair.

We then construct a Directed Graph (DG) by drawing an edge for every pair in their predicted order of arrival. We propose a binning methodology, wherein the nodes of the DG having similar characteristics are grouped into hypothetical containers called bins. The order of arrival of nodes within a bin is unknown. Hence, we determine the order of arrival of nodes across several such bins.

2. PRELIMINARIES AND NOTATIONS

2.1 Scale Free Networks

A Scale-Free Network (SFN) is a network whose degree distribution follows a *power law*. Many real world networks are known to exhibit a decaying degree distribution. This kind of distribution is called a power law. Mathematically, it is defined as

$$\boxed{\mathbf{P(k) \approx ck^{-\gamma}}} \quad (1)$$

where,
k is *degree*,
c is a *normalization constant* and
γ is a *parameter* whose value is typically in the range (2,3)

The high degree nodes in a SFN are often called as "hubs". The power law degree distribution of the SFNs suggests the existence of a small number of high degree nodes. Although the hubs are small in number, they dominate the network to a great extent. Removal of the hubs from the network might cause a network breakdown and disrupt the network characteristics. A typical degree distribution of a SFN with 200 nodes is shown in Figure 1.

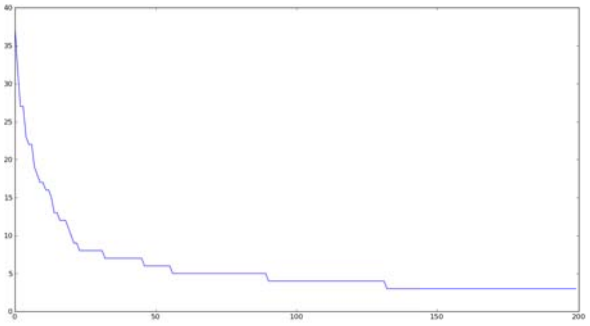

Figure 1: The degree distribution curve for a typical SFN of 200 nodes. This network follows a power law degree distribution.

2.1.1 Generative Model for Scale Free Networks

To explain the power law degree distribution in the real world networks, mechanisms such as preferential attachment and fitness model etc., have been proposed. Barabasi and Albert proposed a randomized algorithm for generating SFNs using a preferential attachment mechanism. This model is referred to as BA model [3].

Figure 2 illustrates the growth of a BA Network $G(9, 3)$.

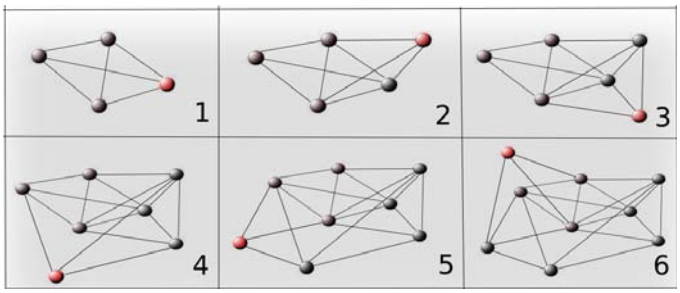

Figure 2: Growth of a BA Network with 9 nodes and 3 connections.

2.2 Directed Acyclic Graph

A Directed Acyclic Graph (DAG) is a directed graph containing no cycles. Indegree of a node v in a directed graph G is defined as $|S| : S \leftarrow \{(u, v)|(u, v) \in E_G\}$. It is denoted by $InDegree(v)$. Outdegree of a node v in a directed graph is defined as $|S| : S \leftarrow \{(v, u)|(v, u) \in E_G\}$. It is denoted by $OutDegree(v)$.

2.3 Lists and Index of an element

A list is an ordered set of elements. Index of an element, u in a list L is the position at which the element u occurs in L, denoted by $index_L(u)$.

2.4 Centrality Measures

A centrality measure is a function that associates a real value with each vertex in a network [6]. The value indicates how central or important the vertex is, in the network. Here, "important" is a subjective term. This gives rise to many centrality measures, each of which rates the nodes according to some property of the node.

2.4.1 Degree Centrality

Degree of a node is often interpreted as an effective measure of influence or importance of that node in a network. Degree of a node u in a graph in denoted by $deg(u)$ [4]. The Degree Centrality assigns a node u with a value that is proportional to $deg(u)$.
Mathematically, for a graph $G(V, E)$:

$$C_{degree}(v) = \frac{deg(v)}{|V| - 1} \qquad v \in V \quad (2)$$

2.4.2 Betweenness Centrality

Betweenness Centrality assigns a node v with a value that is proportional to the number of shortest paths [2] [9], between all other pairs of vertices, that pass through v.

$$\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (3)$$

$$C_{betweenness}(v) = \sum_{s \neq v \neq t \in G} \delta_{st}(v) \quad (4)$$

where $\delta(v)$ is the fraction of shortest paths between s and t that contain the vertex v, σ_{st} denotes number of all shortest paths from vertex s to t, $\sigma_{st}(v)$ denotes the number of shortest paths from s to t passing through v and $C_{betweenness}(v)$ denotes the betweenness centrality index of v.

In our experiments, we have used Brandes approach to compute betweenness centrality [7].

2.4.3 Eigenvector Centrality

The index in Eigenvector Centrality characterizes the individuals in connected networks according to their level of popularity [5] [8].

$$x_u = \frac{1}{\lambda} \sum_{v=1}^{|V|} A_{(u,v)} \; x_v \quad (5)$$

$$x = \frac{1}{\lambda} Ax \tag{6}$$

where A denotes the adjacency matrix of the network $G(V, E)$, x_u denote the centrality score of $u \in V_G$ and λ is a constant. On defining $x = [x_0\ x_1\ x_2\ ...\ x_{|V|-1}]$ as a vector of centrality scores, we can transform the above equation into a matrix form as equation 6.

2.5 Reference Network

In our experiments, we study the SFNs generated using the Barabasi-Albert Model. Let $G_m(V_m, C_m)$ represent a Barabasi-Albert Network whose vertex arrival order is to be deduced. For evaluative purposes, we record the order of arrival of vertices in G_m during its inception. Let $list_{true}$ be a sequence of vertices that represent the actual order of arrival of vertices in G_m. We will be referring to $G_m(V_m, C_m)$ in all the further sections as the input network to the proposed algorithm that predicts order of arrival of nodes.

3. CENTRALITY MEASURE BASED METHODS

3.1 Degree Binning

The degree of a node is the number of connections associated with that node. A naive approach towards the solution to the vertex arrival order prediction problem is to exploit and explore the contribution of this factor.

From the preferential model of SFN construction, it is evident that the last few nodes that get connected to the network will have a relatively low degree, as compared to the nodes that had arrived in the initial stages. Consider the network G_m from section 2.5. Intuitively, we hypothesize that higher the degree of a node, higher is its influence in the network, and earlier it has arrived during the network evolution. We can state with a high probability, that the notable hubs in G_m would have arrived prior to the nodes with a relatively low degree.

Hence, we rank the nodes in the decreasing order of their degree. The equi-degree nodes are assigned with the same ranking. We then place the vertices with the same ranking into a hypothetical container, referred to as a bin. The ranking of a bin is same as the ranking of node(s) inside the bin. The number of bins formed is the total number of unique ranks assigned to the nodes. We then apply a Binning Quality Measure (BQM) to compute the accuracy of our prediction of order of arrival of nodes *across* the bins. BQM quantifies the prediction accuracy on a scale of 0 to 1. Figure 4 illustrates the Binning Methodology that we use to predict the order of arrival of nodes across the bins.

The following mathematical formulation illustrates a technique to quantify the correctness of our prediction. We refer to the technique as Binning Quality Measure (BQM).
Let δ be the number of bins. Let $B = [B_0, B_1, B_2, ..., B_\delta]$ be the predicted chronological bin ordering. We associate a score β between every pair of bins. The final prediction measure η is computed as a ratio of sum of β for all bin-pairs and the total number of bin-pairs.

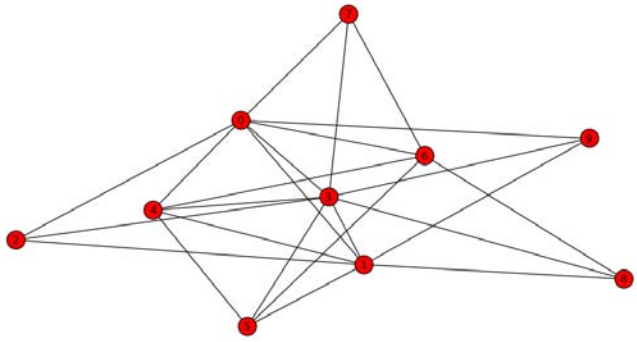

Figure 3: A SFN, constructed using BA model with 9 nodes and 3 connections.

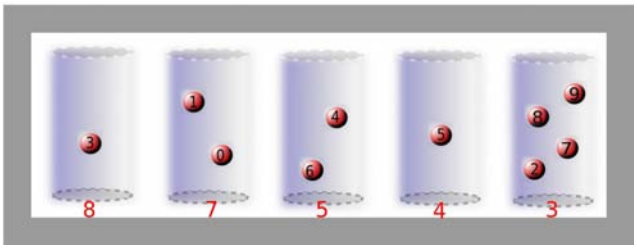

Figure 4: Binning the nodes of the network in Figure 3 based on degree. The numbers below the bins denote the degree of the nodes that are present in the bin.

To calculate β for a pair of bins B_i and B_j, with $i < j$:
Here, we claim that the nodes in B_i has arrived before the nodes in B_j Hence, we impose the condition $i < j$, with reference to the predicted chronological bin ordering B.
For a pair of vertices $u \in B_i$ and $v \in B_j$, we define

$$vertexOrder(u, v) = 1\ if\ index_{list_{true}}(u) < index_{list_{true}}(v)$$

$$vertexOrder(u, v) = 0\ if\ index_{list_{true}}(u) > index_{list_{true}}(v)$$

$$\beta(i, j) = \frac{\sum_{u \in B_i, v \in B_j} vertexOrder(u, v)}{|B_i||B_j|}$$

The final prediction measure η is given by

$$\eta = \frac{\sum_{0<i<j\leq\delta} \beta(i, j)}{{}^{\delta}C_2} \tag{7}$$

3.2 Binning based on Centrality Meaures

The main drawback of binning based on degree is that the degree centrality indices associated with the nodes are not distinct in G_m. This is because there can exist many number of nodes with the same degree. Hence, binning based on degree centrality results in a small number of bins, with a large number of nodes per bin. Ideally, it is desirable to have more number of bins with a less number of nodes

per bin.

We move on to yet another approach which could provide us with a large number of bins. In this approach, we apply χ centrality to main graph. Based on an intuitive conjecture, higher the χ centrality a node, earlier it has arrived in the network evolution. Hence, we sort the vertices in the decreasing order of their χ centrality indices. We group the nodes from this sorted ordering into δ number of bins, each bin containing $\frac{|V_{G_m}|}{\delta}$ number of nodes. We refer to the list of bins thus obtained as $binOrdering_\chi$. In our experiments, we choose χ to be Betweenness Centrality and Eigenvector Centrality. We use BQM (refer section 3.1) to quantify the accuracy of the prediction using binning based on centrality.

4. A NEW VERTEX RANKING: DIFFERENTIAL CORE RANKING

In this section, we formulate a new method of ranking nodes. Let $G(V, E)$ be any graph. Let DCR_G represent the Differential Core Ranking of G.

Let χ be any centrality measure. Let G_0 be the initial graph. Let G_1 be the graph obtained from G_0 after removal of nodes with the minimum degree. The change in χ centrality value of the nodes in G_0 is set as the attribute of the corresponding node. We then apply the above procedure starting with G_1. Let G_2 be the graph obtained from G_1 after the removal of nodes with the minimum degree. The change in the χ centrality value of the nodes in G_1 is added to the attribute of the corresponding node.
In general, let G_{i+1} be the graph obtained from G_i after the removal of nodes with the minimum degree. The change in the χ centrality value of the nodes in G_i is added to the attribute of the corresponding node. This procedure is repeated until there are no nodes left in G_i.

The algorithm to compute DCR_G is as follows:

Let χ represent any centrality measures
Let G_0 represent the given graph G
Let $u \in V(G)$. Let the Differential Core Measure DCM_u be a value associated with u.
Set $DCM_u = 0\ \forall u \in V(G)$
Let χ_{u,G_k} represent the χ centrality value of u.
Let $i \leftarrow 0$
while $|V_{G_i}| > 0$ **do**
  Let $minDeg \leftarrow argmin(deg(u)), u \in V(G_i)$
  Let $minVertices \leftarrow \{u_0, u_1....u_n\}, deg(u_m) = minDeg$
  Let $G_{i+1} \leftarrow graphobtained\ after\ removing\ minVertices\ from\ G_i$
  $DCM_u \leftarrow DCM_u + abs(\chi_{u,G_{i+1}} - \chi_{u,G_i})\ \forall u \in V(G_i)\ and\ u \in V(G_{i+1})$
  $DCM_u \leftarrow DCM_u + abs(0 - \chi_{u,G_i})\ \forall u \in V(G_i)\ and\ u \notin V(G_{i+1})$
  $i \leftarrow i + 1$
end while
$DCR_G \leftarrow \{(DCM_{u_0}, u_0), (DCM_{u_1}, u_1)..., (DCM_{u_{|V_G|}}, u_{|V_G|})\}$

DCR_G gives the Differential Core Ranking of the vertices. DCM_u denotes the centrality score of the node u. Higher the sum of changes in the χ centrality values of a node, higher is its importance in the network.

5. NETWORK RECONSTRUCTION ALGORITHM

In this section of the paper, we describe our algorithm to predict the order of arrival of nodes in G_m. Our Algorithm is mainly divided into 4 subsections. Section 5.1 aims at generation of Synthetic Networks that resemble G_m. Section 5.2 describes a mapping procedure and derivation of prediction lists. In section 5.3, we analyze the prediction list and construct a directed graph. Section 5.4 deals with the transformation of directed graph to a directed acyclic graph and binning of nodes.

5.1 Generation of Synthetic Networks

The main focus of this section of the algorithm is to recreate the growth environment of the reference network G_m. Since the exact replication of G_m is not possible, we generate networks that are similar to G_m in certain characteristics. We refer to these set of networks as Synthetic Networks.

Let α be the number of Synthetic Networks generated. Let S_i and $chronology_{S_i}$ denote the Synthetic Network and the order of arrival of nodes in the corresponding S_i. In our experiments, we use BA model to generate S_i, with $|V_m|$ number of nodes and C_m connections. It is worth noting that every time we generate a Synthetic Network S_i, we keep track of the network growth by recording $chronology_{S_i}$. Since the Synthetic Networks are built on the same model as that of G_m, we hypothesize that the chronology of S_i is *similar* to the actual order of arrival of nodes in G_m. Hence, it is righteous to make use of $chronology_{S_i}$ in predicting the probable order of arrival of nodes in G_m.

5.2 Mapping and Derivation of Prediction Lists

We have now generated α number of BA Synthetic Networks that is similar to G_m in terms of the number of vertices $|V_m|$ and connections C_m. The chronology of the Synthetic Networks S_i, where $1 \leq i \leq \alpha$, is known. In this section, we intend to derive an ordering of nodes in V_m, corresponding to each S_i. This ordering of nodes is the predicted order of arrival of nodes in G_m (during its inception), derived in accordance with $chronology_{S_i}$. We refer the node ordering corresponding to S_i as $PredList_i$. The procedure that we follow to deduce $PredList_i$ is explained in the remainder of the section.

We apply DCR, with χ as the base centrality measure [refer to section 2.4], to G_m in order to obtain DCR_{G_m}. DCR_{G_m} is a list of vertex rankings sorted according to their DCM values. [refer to section 4]

Consider a Synthetic Network S_i. We apply DCR, with χ Centrality as the base centrality measure, to S_i in order to obtain DCR_{S_i}.

Both DCR_{G_m} and DCR_{S_i} lists the vertices of G_m and S_i respectively in the decreasing of their importance. Lower the position of a vertex in these lists, higher its importance in the corresponding network. A direct bijection mapping is carried out between DCR_{G_m} and DCR_{S_i}. This mapping maps the equi-important vertices in both the networks.

Mathematically, we define a mapping function as:
Let $f_{map} : V_{S_i} \rightarrow V_{G_m}$ be a direct bijection between V_{S_i} and V_{G_m}
i.e, $f_{map}(u) = v$ where $u \in V_{S_i}, v \in V_{G_m}$ and $index_M(u) = index_N(v)$

We propose that the nodes of equal importance in G_m and S_i have the same chronological ranking. Since we know $chronology_{S_i}$, we deduce $PredList_{S_i}$ by replacing each vertex u in $chronology_{S_i}$ with $f_{map}(u)$.

We repeat the above procedure for each S_i. At this stage, we have α prediction lists, denoted by $PredList_i$, each corresponding to a particular S_i.

Algorithm for Mapping:

Input: The Reference Network G_m and Synthetic Networks $\{S_1, S_2, ...S_\alpha\}$
Output: α Prediction Lists
Apply DCR, with χ as the base centrality measure, to G_m
Let $u_i \in V_m : 1 \leq i \leq |V_m|$
Let $DCR_{G_m}(u_i)$ denote the DCR associated with the vertex u_i
Let the tuple list $M \leftarrow \{(DCR_{G_m}(u_1), u_1), (DCR_{G_m}(u_2), u_2), ...(DCR_{G_m}(u_{|V_m|}), u_{|V_m|})\}$
Sort M in the descending order of $DCR_{G_m}(u_i)$
for all $i = 1$ *to* α **do**
  Let $v_j \in V_{S_i} : 1 \leq j \leq |V_{S_i}|$
  Let $(v_1, v_2, ...v_{|V_{S_i}|})$ denote $chronology_{S_i}$
  Apply DCR, with χ centrality as the base centrality, to the Synthetic Network S_i
  Let $DCR_{S_i}(v_j)$ denote the DCR of the vertex v_j
  Let the tuple list $N \leftarrow \{(DCR_{S_i}(v_1), v_1), (DCR_{S_i}(v_2), v_2), ...(DCR_{S_i}(v_{|V_{S_i}|}), v_{|V_{S_i}|})\}$
  Sort N in the descending order of $DCR_{S_i}(v_j)$
  Let $f_{map} : V_{S_i} \rightarrow V_{G_m}$ be a bijection between V_{S_i} and V_{G_m}
  $f_{map}(u) = v$ where $u \in V_{S_i}, v \in V_{G_m}$ and $index_M(u) = index_N(v)$
  $PredList_i \leftarrow (f_{map}(v_1), f_{map}(v_2), ...f_{map}(v_{|V_{S_i}|}))$.
end for

Figures [5 to 8] illustrate an instance of Mapping of nodes between G_m and any $S_i : 1 \leq i \leq \alpha$.

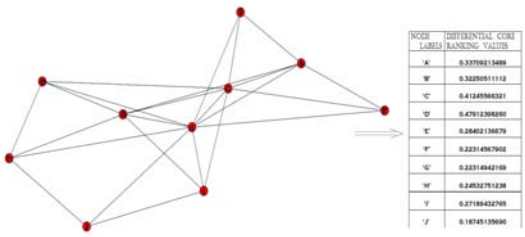

Figure 5: Applying Differential Core Ranking, with Betweenness Centrality as the base centrality, to G_m.

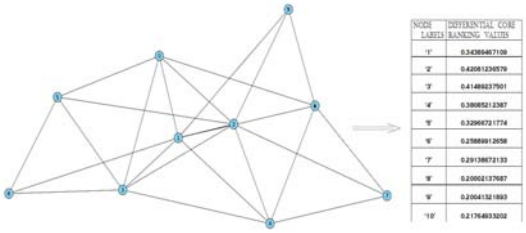

Figure 6: Applying Differential Core Ranking, with Betweenness Centrality as the base centrality, to one of the $S_i : 1 \leq i \leq \alpha$.

vertex ordering of S_i based on DCR

'2'	0.42081236579
'3'	0.41489237501
'4'	0.38085212387
'1'	0.34389467109
'5'	0.32966721774
'7'	0.29138672133
'6'	0.25889912658
'10'	0.21764933202
'9'	0.20041321893
'8'	0.20002137687

vertex ordering of G_m based on DCR

'D'	0.47912398260
'C'	0.41245566321
'A'	0.33709213489
'B'	0.32250511112
'E'	0.28402136679
'I'	0.27189432765
'H'	0.24532751238
'G'	0.22314942169
'F'	0.22314567902
'J'	0.18745135690

Figure 7: Vertex ordering based on decreasing Differential Core Ranking for V_{G_m} and V_{S_i}.

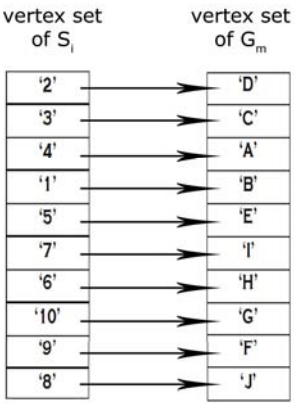

Figure 8: Direct bijection mapping of vertices between Lists in figure 8.

$chronology_{S_i}$	1	2	3	4	5	6	7	8	9	10
$PredList_i$	'B'	'D'	'C'	'A'	'E'	'H'	'I'	'J'	'F'	'G'

Figure 9: Deduction of $PredList_i$ by reordering the nodes of V_m according to $chronology_{S_i}$.

5.3 Analysis of Prediction Lists and Construction of Directed Graph

In the previous section, we have deduced α number of Prediction Lists, $PredList_i : 1 \leq i \leq \alpha$. For every pair of vertices $(u, v) : u, v \in V_{G_m}$, we find the order of occurrence of u and v in each $PredList_i$. Let $P_{(u,v)}$ denote the probability of u arriving before v during the inception of G_m. We compute $P_{(u,v)}$ as the fraction of the number of times u has occurred before v in the α Prediction Lists. By intuitive reasoning, it is not hard to infer that, if $P_{(u,v)} < 0.5$, then v has probably arrived before u during the inception of G_m. Hence, we set $P_{(v,u)} = 1 - P_{(u,v)}$. We then construct a Directed Graph DG with vertex set $V_{DG} = V_m$, and edge set $E_{DG} = \phi$. A directed edge from u to v in DG indicates that u has arrived before v during the construction of G_m. For a pair of vertices (u, v):
if $P_{(u,v)} > 0.5$, then we say that u has arrived before v with a probability $P_{(u,v)}$
if $P_{(u,v)} < 0.5$, then we say that v has arrived before u with a probability $1 - P_{(u,v)}$

The algorithm to deduce DG is presented below:

Let $S \leftarrow \{S_1, S_2, S_3, ...S_\alpha\}$ denote the set of Synthetic Networks
Let $PredList_i$ denote the Prediction List corresponding to $S_i : 1 \leq i \leq \alpha$ [Refer to algorithm in section (5.2)]
Construct a Directed Graph DG_m with $V_{DG_m} = V_m$ and $E_{DG_m} = \phi$
Let $P_{(u,v)}$ be the probability associated with $(u, v) : u, v \in V_{DG_m}$ in determining if u has come before v.
for all unordered pairs $(u, v) : u, v \in V_m$ *and* $u \neq v$ **do**
  $count \leftarrow 0$
  for $i \leftarrow 1$ *to* α **do**
    if $index_{S_i}(u) < index_{S_i}(v)$ **then**
      $count \leftarrow count + 1$
    end if
  end for
  $P_{(u,v)} \leftarrow count/\alpha$
  if $P_{(u,v)} > 0.5$ **then**
    append (u, v) to E_{DG_m} with a weight $P_{(u,v)}$
  else
    append (v, u) to E_{DG_m} with a weight $1 - P_{(u,v)}$
  end if
end for

In the next section, we analyze DG to obtain final predicted order of arrival of nodes in V_{G_m}.

5.4 Transformation of Directed Graph and Node Binning

In this section, we process DG obtained from the previous section to deduce the final prediction of order of arrival of nodes in G_m. Ideally we expect DG to be acyclic in nature, as cycles would give rise to inconsistent prediction order among the nodes involved in the cycle. For example, lets say, (u, v) and (v, w) are in E_{DG}. This implies that u has arrived before v and v has arrived before w. Hence, w must have arrived before u. If (w, u) also an edge, then it leads to a contradiction in the chronological ordering of u, v and w. Since there is a fair possibility that DG can be a cyclic graph, we intend to transform it into a Directed Acyclic Graph (DAG) and remove the inconsistencies involved. In our algorithm, we use a greedy technique to achieve the above.

The algorithm to transform DG into DAG is presented below:
  Input: Directed Graph DG.
  Output: Directed Acyclic Graph DAG.
  while DG contains cycles **do**
    Remove the edge (u, v) with the least $P_{(u,v)} : (u, v) \in E_{DG}$.
  end while

The DAG thus obtained is free from inconsistencies. $InDegree(v)$ represents the number of nodes that have been predicted to arrive after the arrival of v. Ideally, the node that had arrived earliest should have zero InDegree. The next earliest node should have an InDegree equal to 1 and so on. Since we are probabilistically simulating the growth environment of G_m, it is practically not always possible for the nodes to have the same sequence of InDegree as that of their order of arrival.

In the final step binning, We will find all the vertices v in DAG having the least $InDegree(v)$ and bunch them into a bin. The binned vertices are removed from DAG, we call these vertices to have arrived first. Later iterating this procedure over and over till there are no nodes left in DAG. We obtain Final predicted bin ordering.

Algorithm to bin the nodes from DAG is presented below:
  Input: Directed Acyclic Graph DAG
  Output: Bin Ordering
  $count \leftarrow 1$
  while $|V_{DAG}| \neq 0$ **do**
    $minInDeg \leftarrow arg\ min(InDegree(u))\ where\ u \in V_{DAG}$
    Let $B_{count} \leftarrow \{u : \forall u \in V_{DAG}$
      $and\ InDegree(u) = minInDeg\}$
    Remove all the nodes in B_{count} from V_{DAG}
      i.e, $V_{DAG} \leftarrow V_{DAG} - B_{count}$
    $Count \leftarrow Count + 1$
  end while
  Let $binOrdering \leftarrow [B_1, B_2, B_3, ...B_{Count}]$

binOrdering gives the predicted chronological sequence of bins. The order of arrival of nodes within a bin is unknown. But the order of arrival of nodes across several such bins can be determined. The accuracy of this prediction, in contrast with accuracy of prediction using centrality measures, is discussed in the next section.

6. RESULTS AND DISCUSSIONS

6.1 Comparison between the predictions from Differential Core Ranking and Plain Centrality

Centrality Index of a vertex in a network indicates its relative importance in the network (refer section 2.4). Let χ be a base centrality measure. We hypothesize that, higher the relative importance of a vertex in a network G_m, earlier it has arrived during its evolution. Hence, the vertices in the network are arranged in the descending order of their χ

centrality indices. Let this ordering of the nodes be denoted by $Plain\chi_{G_m}$. We apply DCR (refer section 4), with the same centrality χ as the base centrality, to the network G_m. The vertices in the network are arranged in the descending order of their DCR values. Let this ordering of the nodes be denoted by $Differential\chi_{G_m}$.

For experimental purposes, the actual order of arrival of nodes in G_m is recorded. It is denoted by $list_{true}$. Let the predicted order be denoted by $list_{pred}$. To compute the accuracy of our prediction, we define a new quality measure called $\eta(list_{true}, list_{pred})$.

$$\eta(list_{true}, list_{pred}) = \frac{n_c}{|V_{G_m}|C_2} \tag{8}$$

where n_c is the number of pairs in $list_{pred}$ that are in correct relative order with respect to $list_{true}$. To compare the prediction accuracy for the lists $Plain\chi_{G_m}$ and $Differential\chi_{G_m}$, we just compare the values of $\eta(list_{true}, Plain\chi_{G_m})$ and $\eta(list_{true}, Differential\chi_{G_m})$. In our experiments we consider the cases where χ represents Degree Centrality, Betweenness Centrality and Eigenvector Centrality.

The figures [10 to 12] represent the plots used to compare the values of $\eta(list_{true}, Plain\chi_{G_m})$ and $\eta(list_{true}, Differential\chi_{G_m})$ for varying number of nodes. Note that the number of connections C_m is kept constant. The red line indicates the η-measure using Differential Core Centrality, with χ as the base centrality. The blue line indicates the η-measure obtained using plain χ centrality. The x-axis represents the number of nodes. The y-axis denotes $\eta(list_{true}, Differential\chi_{G_m})$ and $\eta(list_{true}, Plain\chi_{G_m})$. In each of the plots, the choice of χ varies accordingly.

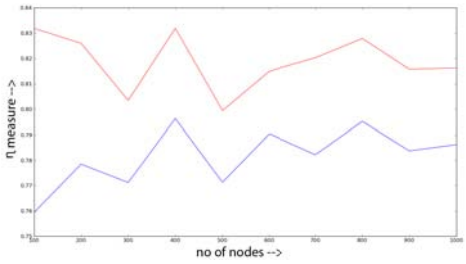

Figure 10: χ : Betweenness Centrality

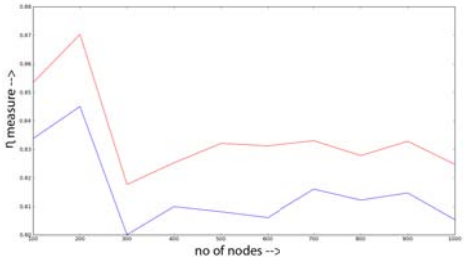

Figure 11: χ : Degree Centrality

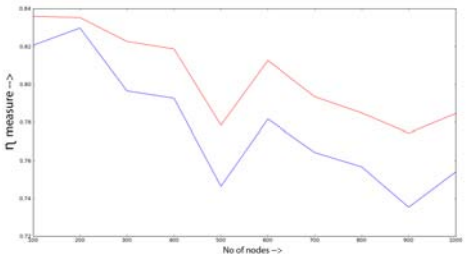

Figure 12: χ : Eigen Vector Centrality

We now run the same exeriment on BA networks with 1000 nodes and varying number of connections. The results are depicted in figures [13 to 15]. The x-axis represents the number of connections. The y-axis denotes $\eta(list_{true}, Differential\chi_{G_m})$ and $\eta(list_{true}, Plain\chi_{G_m})$. In each of the plots, the choice of χ varies accordingly.

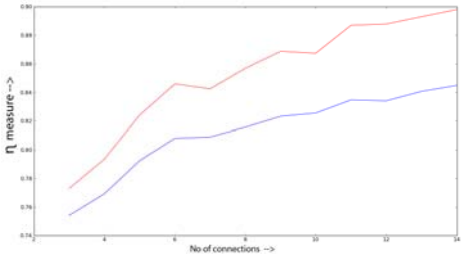

Figure 13: χ : Betweenness Centrality

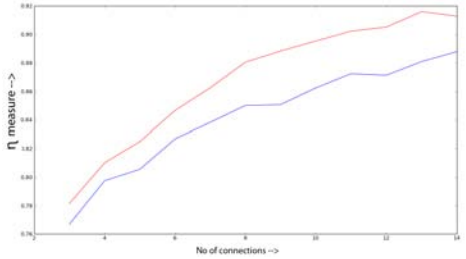

Figure 14: χ : Degree Centrality

6.2 Prediction of arrival order in every node pair with an attached probability

The outcome of section 5.3 is a weighted directed graph DG. We have associated a probability $P_{(u,v)}$ with every directed edge $(u, v) \in E_G$. $P_{(u,v)}$ indicates the probability with which u has arrived before v. From the construction mechanism of DG, it is clear that $P_{(u,v)} > 0.5$. Closer the value of $P_{(u,v)}$ to 0.5, harder it is to ascertain the chronological ordering of u and v. Note that there is a fair possibility that DG can contain cycles. We claim that the inconsistencies in the prediction might be caused due to edges with $P_{(u,v)}$ close to 0.5. This may lead to a formation of cycles.

We now present the analytical results that we have obtained, considering G_m as reference network. We have generated G_m using a BA model with 1000 nodes and 3 connections. We generate 50 synthetic networks. So, we set $\alpha = 50$. The analytical results thus obtained is given below:

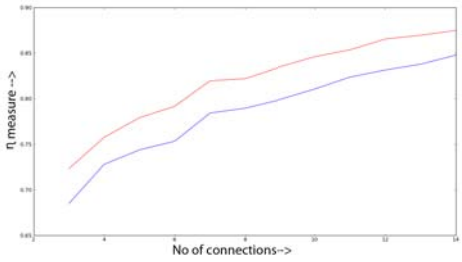

Figure 15: χ : Eigen Vector Centrality

R=range of $P_{(u,v)}$	No of edges whose $P_{(u,v)} \in R$ / $\lvert E_{DG} \rvert$	Fraction of pairs in the correct relative order with $list_{true}$
(0.5, 0.6]	0.216606606607	0.546827487407
(0.6, 0.7]	0.156592592593	0.652739778568
(0.7, 0.8]	0.137975975976	0.767770861446
(0.8, 0.9]	0.156284284284	0.864482988317
(0.9, 1.0]	0.308434434434	0.967824850873

Statistically, from the above table, we observe that the edges (u, v) having $P_{(u,v)}$ in $(0.5, 0.6]$ constitute around 20% of the edges. We also note that only around 50% of these edges are in the correct relative order with $list_{true}$. Since a large fraction of edges belonging to this range are in incorrect relative ordering, they contribute to the cycle formation. Cycles introduce inconsistencies in node arrival order, hence they have to be removed. From our experiments, we have found out that DG will become acyclic when we remove the edges (u, v) continually in the increasing order until $P_{(u,v)} \approx 0.6$. We implement the same technique in section 5.4 to transform DG to DAG.

Based on the facts and figures from the table, we observe that the fraction of pairs that are in correct relative order with $list_{true}$ increases as the sampled range increases. Hence we conclude that, higher $P_{(u,v)}$ implies a stronger notion of relative ordering of (u, v).

6.3 Comparison between the predictions from DCR binning and Plain Centrality binning

The end result of our method (section 5.4) is the ordering of the bins, referred to as $binOrdering_{DCR\chi}$. Let Δ be the number of bins in $binOrdering_{DCR\chi}$. Let $\eta_{DCR\chi}$ denote the BQM value of $binOrdering_{DCR\chi}$, where χ refers to the base centrality measure for DCR.

We derive the $binOrdering_{\chi}$ (refer section 3.2) with Δ number of bins, and χ indicating the centrality measures. Let $binOrdering_{betweenness}$, $binOrdering_{eigen}$ and $binOrdering_{degree}$ denote the chronology of bins with χ set as Betweenness, Eigenvector and Degree Centralities respectively.

Let $\eta_{betweenness}$, η_{eigen} and η_{degree} denote the BQM value of $binOrdering_{betweenness}$, $binOrdering_{eigen}$ and $binOrdering_{degree}$ respectively. Finally, we compare $\eta_{betweenness}$, η_{eigen}, η_{degree} and $\eta_{DCR\chi}$ where χ is the base centrality (refer section 4).

Figures [16 to 18] denote the BQM scores for various binning methodologies for the reference graphs G_m of 1000 nodes and 3 connections. In our experiment, we have set $\alpha = 50$. We perform the above said experiment three times. For each experiment, we choose different base centralities and different G_m. We observe that the DCR method yields more accurate results compared to any other plain centrality based approaches.

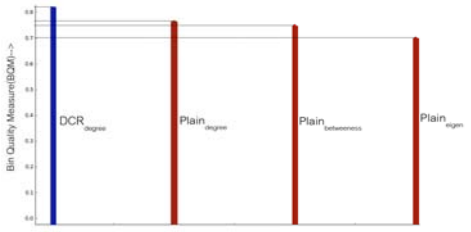

Figure 16: $\eta_{DCR_{degree}} = 0.804513946531$, $\eta_{degree} = 0.767615011251$, $\eta_{betweenness} = 0.759827243464$, $\eta_{eigen} = 0.695466553648$, **number of bins=91**

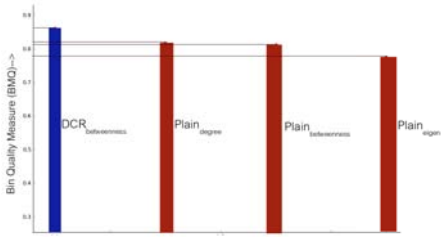

Figure 17: $\eta_{DCR_{betweenness}} = 0.87153926121$, $\eta_{degree} = 0.8251012352$, $\eta_{betweenness} = 0.8158246115$, $\eta_{eigen} = 0.7823167778$, **number of bins=63**

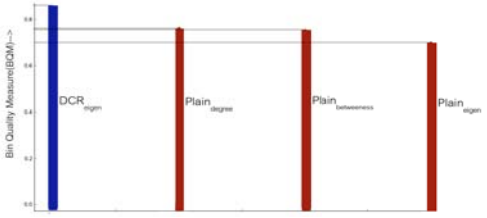

Figure 18: $\eta_{DCR_{eigen}} = 0.84654821986$, $\eta_{degree} = 0.7697124538121$, $\eta_{betweenness} = 0.753169421166$, $\eta_{eigen} = 6899122714632$, **number of bins=77**

7. CONCLUSION

We presented a novel framework for uncovering the precursor of a SFN evolved by BA model. Our approach involves the synthesis of many such SFNs, mapping these SFNs with the reference network based on DCR score associated with the nodes and arriving at the final prediction order. We presented 3 results. 1. DCR based prediction, which proved to provide better predicted node arrival results than any other centrality based approaches. 2. Arrival order of every pair of nodes in a SFN, with an associated probability. We empirically proved that most of the node pairs with high probability indeed arrived in the order that we predicted. 3. We also proved that DCR based prediction, when applied in

conjunction with the binning methodologies, offered a better accuracy compared to any other plain centrality based approaches.

8. REFERENCES

[1] R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.

[2] J.M. Anthonisse. The rush in a directed graph. Technical Report BN 9/71, Stichting Mathematisch Centrum, 1971.

[3] A. L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, October 1999.

[4] A. Bavelas. Communication Patterns in Task-Oriented Groups. *The Journal of the Acoustical Society of America*, 22(6):725–730, 1950.

[5] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972.

[6] S. Borgatti and M. Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, October 2006.

[7] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[8] U. Brandes and T. Erlebach. Network Analysis. Methodological Foundations. *Network Analysis, Lecture Notes in Computer Science*, 3418, 2005.

[9] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.