

A Comparison study of nodal attributes free social network models

Wilson Li

litianyi@mit.edu

May 17, 2018

1. Introduction

A social network is a set of people or groups. People or groups constitute the nodes and their interaction or “ties” constitute the edges. We study friendship among individuals in this paper. Modeling social networks helps us understand how social network form and evolve as well as the structure of interaction. A large variety of models have been presented in the complex networks literature in recent years to explore how local mechanisms of network formation produce global network structure. In this paper we review, classify, compare and analyze such typical models.

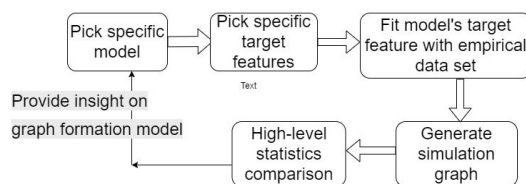
Three major features of empirical social network have been widely discussed and investigated.

(1) **Small-world effects**: phenomenon of strangers being linked by a short chain of acquaintances. It's usually reflected in a relatively small diameter for the network.

(2) **Triadic closure**: the probability of a tie between two individuals is much greater if they share one or several common friends. It's usually reflected in high clustering coefficient.

(3) **Skewed degree distribution**: a power law like degree distribution is usually observed in social network. Small proportion of nodes have unusually large number of ties.

These features provide intuition of how to set formation rules in random graph model, while different models may emphasize different properties of the network. Following graph shows the framework for such comparison study. The goal of this paper is not to rank models, but to examine pros and cons of each model to shed light on how local mechanisms of network formation is linked to the behaviour of global network structure.



2. Models

The formation models can be classified into two main categories: those in which the addition of new links is dependent on the local network structure (network evolution models, NEMs), and those in which the probability of each link existing depends only on nodal attributes (nodal attribute models, NAMs). NEMs can be further subdivided into growing models, in which nodes and links are added until the network contains the desired number N of nodes, and dynamical models, in which the steps for adding and removing ties on a fixed set of nodes are repeated until the structure of the network no longer statistically changes. One of them is based solely on nodal attributes, and the other incorporates structural

dependencies. All of these models produce undirected networks without multiple links or self-links, and all networks are treated as unweighted.

In our case, we mainly study how topology-related formation mechanisms influence the final topology of the network. They don't rely on special attributes on different nodes so every node is treated equally when link is established. We hope this will reveal some underlying and intrinsic constraints on properties of graph formulation process without interference of multi-dimensional information independent of the graph that could belong to a node. Thus, we pick DEB (one typical growing NEM), Vaz (one typical dynamical NEM), Watts–Strogatz small world model and configuration model.

1) DEB (Davidson et al., 2002)

2 free parameters N, p

Network starts with N nodes.

I) Select a node i randomly, and

a) if i has fewer than two ties, introduce it to a random node

b) otherwise pick two neighbors of i and introduce them if they are not already acquainted.

II) Select a random node and with prob. p remove all of its ties.

We modify original model such that in case 1), only if node i has fewer than k ties (k = average degree of the empirical network), we'll introduce it to a random node. This iterative process ended when average degree stabilizes. We'll use this **Modified DEB Model** for following simulations.

2) Váz (Vázquez, 2003)

2 free parameters: N, u

I) With probability $1 - u$, add a new node to the network, connecting it to a random node i . Potential edges are created between the newcomer n and the neighbors j of i (a potential edge means that n and j have a common neighbor, i , but no direct link between them).

II) With probability u , convert one of such potential edges generated on any previous time step to an edge. Potential edges generated by converting an edge are ignored.

Iterative formation process completes when network contains N nodes.

3) Small World Model

3 free parameters: N, K, β

1. Construct a regular ring lattice, a graph with N nodes each connected to K neighbors, $K/2$ on each side. That is, if the nodes are labeled

$$n_0 \dots n_{N-1}, \text{ there is an edge } (n_i, n_j) \text{ if and only if } 0 < |i - j| \bmod \left(N - 1 - \frac{K}{2}\right) \leq \frac{K}{2}.$$

2. For every node $n_i = n_0, \dots, n_{N-1}$ take every edge (n_i, n_j) with $i < j$, and rewire it with probability β . Rewiring is done by replacing (n_i, n_j) with (n_i, n_k) where k is chosen with uniform probability from all possible values that avoid self-loops ($k \neq i$) and link duplication (there is no edge $(n_i, n_{k'})$ with $k' = k$ at this point in the algorithm).

4) Configuration Model

The configuration model allows one to generate a network model that has exactly a prescribed degree distribution.

3. Datasets


We utilize one facebook social network datasets from SNAP(Stanford Network Analysis Project).The source URL is <https://snap.stanford.edu/data>. This dataset displays certain level of assortativity,high clustering and certain level of skewed degree distribution. Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value,so privacy isn't violated.Note that these statistics were compiled by combining the ego-networks, including the ego nodes themselves (along with an edge to each of their friends). Our empirical networks from this facebook dataset are unweighted, meaning that tie weight are not specified. Averaged basic statistics of this dataset is displayed in the following table. More plots of its statistics are shown in Section 5(Comparison) in connection with the fitted models.

Dataset statistics	
Nodes	4039
Edges	88234
Nodes in largest WCC	4039 (1.000)
Edges in largest WCC	88234 (1.000)
Nodes in largest SCC	4039 (1.000)
Edges in largest SCC	88234 (1.000)
Average clustering coefficient	0.6055
Number of triangles	1612010
Fraction of closed triangles	0.2647
Diameter (longest shortest path)	8
90-percentile effective diameter	4.7

4.Fitting model

In order to compare networks generated by different models,we need to pick hyperparameters to determine simulation. To guarantee the effectiveness of comparison,we should fit the models to real-world data with respect to as many of the most relevant network features as the model parameters allow. A natural and generalized way of such fitting would be via supervised machine learning.We could devise a error function computed by a weighted sum of error distance for each target fitting features so that optimizing set of hyperparameters could be obtained.But for small-scale study like we do,it's more feasible and straightforward to tune each target feature separately.

The most important properties that we wish to align between the models and the data are the number of nodes and links since they determine the scale and size of the network. Considering that our data set is a connected component,those two features could be reflected in largest component size and average degree. For models that requires more than two parameters,we could involve number of triangles or average clustering coefficient because triadic closure or high clustering coefficient is quite universal in social network. One special case is configuration model since they only depend on degree distribution.The following table shows details of how we fit selected models.For each model,we simulate ten models and take the average to offset the randomness of simulation.

	Facebook(dataset)	Modified DEB	Vaz	Small World	Configuration
Parameters		N = 4039 p = 0.015	N = 4039 u =	N = 4039 K = 21 $\beta = 0.063$	degree distribution 
Largest component size	4039	4039	4039	4039	4039
Average degree	43.69	42.74	44.08	42.98	43.69

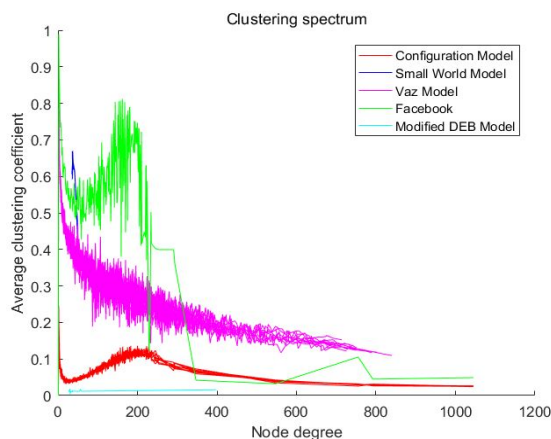
Besides,the intuition behind modifying original DBE model here is that original model converges too early on average degree for our relatively large and dense data set during

initial attempts to fit DEB, which couldn't be resolved by simply decreasing p (the probability of node deletion). It indirectly shows that the original DEB model isn't suitable for large-scale and dense network configuration. If we increase the upper bound degree requirement for establishing new global or "distant" connections, more new edges could be added to the graph when local structure is approaching saturation after many iterations. High average degree could be reached this way. Contrarily, this could highly reduce the clustering coefficient because global connection is more prioritized than local neighbour links.

5. Statistics Comparison

Having fitted the models, we assess some high-level properties that haven't been used to fit models including their clustering spectra, degree distributions, assortativity, shortest path length distributions and modularity. This way we can see how different models generalize to all desired properties. We'll compare how different models perform in each high-level statistics so that we could grasp a better understanding of what kind of formation mechanisms contribute more to some network structure characteristics.

Clustering spectrum

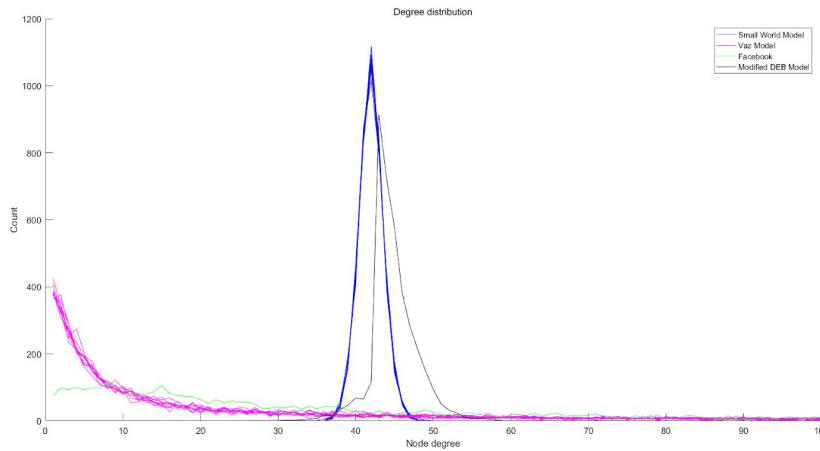


	Average clustering coefficient
Facebook	0.6055
Small World	0.6042
Vaz	0.4534
Modified DEB	0.0117
Configuration	0.0609

Indicated by previous findings, many network models display roughly an inverse relation between node degree and clustering coefficient. We could observe that at the head of curve, Facebook empirical network, small world model and Vaz model all showcase a slightly lower inverse-like decay speed. Previous findings hold true for these models. As the number of average clustering coefficient shows in above figures, Modified DEB model does damage the effect of triadic closure, resulting in the lowest clustering coefficient among all models, which supports the hypothesis in previous sections. Clustering coefficient also seems to be independent with degree in DEB model. Vaz model has the most inverse-like decaying pattern, while configuration model and real data all have a heap in the middle of the curve. Small world model has the most closest high clustering coefficient to real Facebook data and fastest decaying rate at the beginning of the curve.

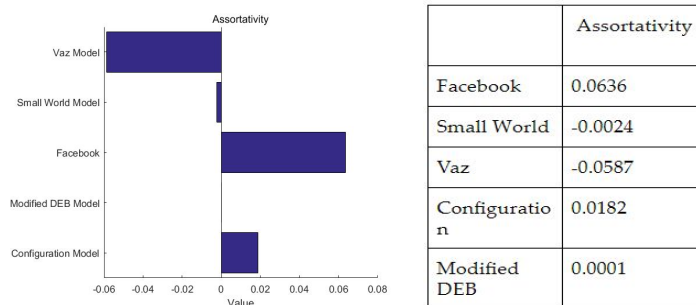
Based on observation, combining initial lattice configuration in Small World model and growing mechanisms in Vaz model would be a potentially interesting model for investigation in respect to clustering.

Degree Distribution



Empirical facebook network and Vaz network all have a quite broad degree distribution, power law shape with a long but low tail corresponds to a skewed degree distribution commonly found in social network. Modified DBE and Small World network, on the other hand, contains large amount of nodes with roughly same degrees, represented by the narrow and sharp curve. Vaz's model specifies rules for how new nodes connect to existing nodes and how new links are established between nodes that share common neighbours, which appropriately mimics the way how new members join in existing groups and constant interactions within old groups. Models with detailed and commonsensible rules for interaction usually result in a better fit for link/degree distribution.

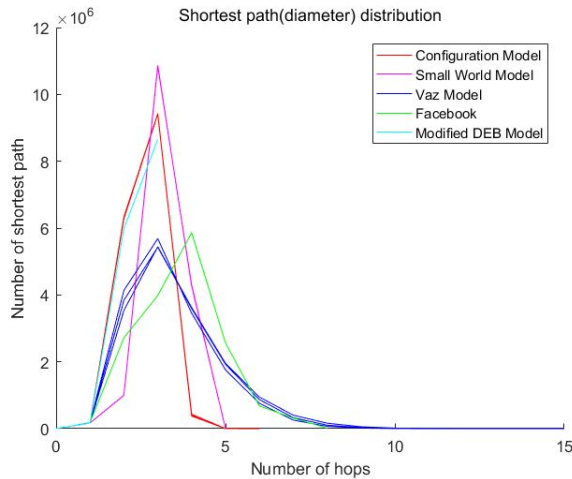
Assortativity



Assortativity represents a preference for a network's nodes to attach to others that are similar in degrees count. We use the degree_pearson_correlation_coefficient to quantify assortativity, with the formula in the following figure where k_i, k_j are the degrees of node i, j and E is the number of edges. High assortativity in the Facebook dataset reveals the phenomenon that popular people usually like to be friends with other popular or famous people and link between popular and unpopular people are usually more unlikely. All of our models show a poor assortativity performance. This makes sense since that nodal attribute is not a factor for these models and random node selection can't quite make link between high degree nodes more likely to be picked. If we could implement mechanisms where random link are picked for formation, high assortativity is more likely to be reached without the involvement of homophily nodal attributes.

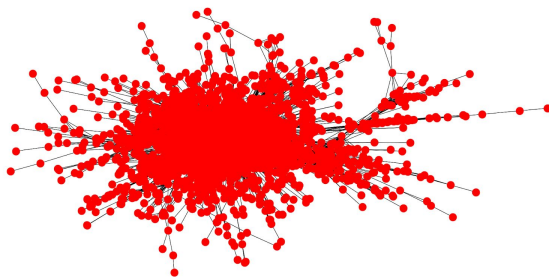
$$r = \frac{\sum_e k_i k_j / E - [\sum_e \frac{1}{2}(k_i + k_j)]^2 / E^2}{\sum_e \frac{1}{2}(k_i^2 + k_j^2) / E - [\sum_e \frac{1}{2}(k_i + k_j)]^2 / E^2},$$

Shortest path distribution

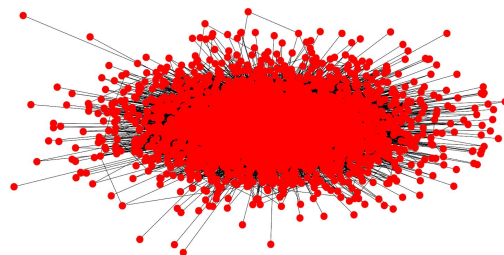


All networks display reasonable shortest path length distributions and roughly the same hop count distribution pattern. Most shortest path length lies between 0 and 5 with diameter lying between 5 and 10. The Váz model, surprisingly, has rather long path length despite its broad degree distribution. Generally, high degree nodes decrease path lengths across the network, but other properties in Vaz model seem to offset this effect. Modified DEB model has no doubt the shortest path length among all since a broad and extremely less skewed degree distribution is observed in its simulation.

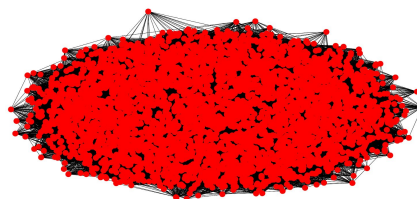
Modularity



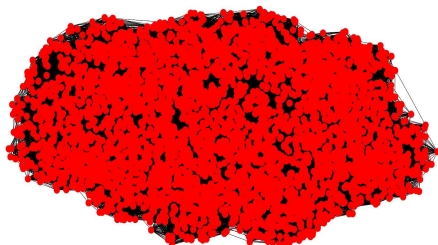
Vaz Model Network↑



Configuration Model Network↑



Modified DEB Model Network→



←Small World Model Network



Facebook Network↑

Model	Vaz	Configuration	Modified DEB	Small World	Facebook(Dataset)
Modularity	0.209	0.156	0.452	0.557	0.775

Modularity measures the strength of division of a network into modules. The table and the visualization are quite coherent with each other. Configuration model shows the weakest modularity. This indicates that degree distribution alone doesn't provide enough information for community configuration or existence of sub-structure.

6. Extensions

Re-examining the whole process, there are certain systematic aspects worth discussing and doing extensive research on. 1) **Biased data collection through ego-network**: Social network data is usually collected as multiple ego-networks and merged together to approximate the overall graph as this is the most practical way. Yet graph based on biased data could have underestimated impact on network structure. Level of such influence should be investigated and better approaches for collecting friendship network data could be proposed. 2) **Effects of different weights combination of target features**: For more complex and large-scale network, complex models that require more hyperparameters to approximate sometimes can't be avoided. Simulation of graph could take a long time, and hence one iteration of machine learning step would be extraordinary long. How to optimize training simulated graph is also an interesting topic.

7. Conclusions

We have described and analyzed four models: Modified DEB, Vaz, Small World and Configuration model. We applied these models to fit one empirical facebook network from SNAP using certain basic target features, and compared how they perform under other high-level characteristics. Based on observation, we suggest possible links between local mechanisms and global network for explanation: Merging Small world lattice configuration and Vaz's link formation could possibly enhance performance on clustering coefficient; Vaz model provides a standard way of mimicking interaction in friendship network; Link random selection could be implemented if model is required to emphasize on high assortativity; Modified DEB model alike tradeoff low shortest path length with less skewed degree distribution; Community structure requires more than degree distribution to realize, etc. It's impractical to focus on making every aspect of the simulation perfect. Learn to merge contributing part of different models and tradeoff different properties would always be the themes.

References

- [1]M.E.J.Newman, D.J.Watts, and S.H.Strongatz., 2002. *Random graph models of social networks*.Colloquium Paper.
- [2]Riitta Toivonen, Lauri Kovanena, Mikko Kiveläa, Jukka-Pekka Onnelab, Jari Saramäkia, Kimmo Kaskia., 2009. *A comparative study of social network models: network evolution models and nodal attribute models*. Social Networks,31(4):240-254.
- [3]Julian McAuley, Jure Leskovec, Stanford, USA. 2012.*Learning to Discover Social Circles in Ego Networks*.NIPS..

Appendix:

Code for model simulatoion with SNAP module in python

1)DEB.py:

```
import snap
import random

def DEB_create(N,p,max_step,avg_deg):
    deb = snap.TUNGraph.New()
    for i in range(0,N):
        deb.AddNode(i)
    Rnd = snap.TRnd(N-1)
    Rnd.Randomize()
    for i in range(0,max_step):
        flag = 0
        NId_1 = deb.GetRndNId(Rnd)
        if(deb.GetNI(NId_1).GetDeg()<avg_deg):
            while(True):
                NId_2 = deb.GetRndNId(Rnd)
                if(deb.AddEdge(NId_1,NId_2) == -1):
                    break
            else:
                for a in deb.Nodes():
                    if(deb.IsNode(a.GetId()) & deb.IsEdge(NId_1,a.GetId())):
                        for b in deb.Nodes():
                            if(deb.IsNode(b.GetId()) & deb.IsEdge(NId_1,b.GetId()) &
deb.IsEdge(a.GetId(),b.GetId())==False):
                                deb.AddEdge(a.GetId(),b.GetId())
                                flag = 1
                                break
                        if(flag==1):
                            break
                NId_3 = deb.GetRndNId(Rnd)
                if(random.random()<p):
                    deb.DelNode(NId_3)
```



```
print deb.GetEdges(),i //Used to monitor whether average degree is stablized
```

```
return deb
```

2) Vaz.py:

```
import snap
```

```
import random
```

```
def Vaz_create(N,u):
```

```
vaz = snap.TUNGraph.New()
```

```
vaz.AddNode(0)
```

```
vaz.AddNode(1)
```

```
vaz.AddEdge(0,1)
```

$$\text{PE} = []$$

```
while(vaz.GetNodes()<N):
```

```
if(random.random()
```

```
if(PE!=[]):
```

```
edge = PE[random.randint(0,len(PE)-1)]
```

```
vaz.AddEdge(edge[0],edge[1])
```

else:

```
new = vaz.GetNodes()
```

```
vaz.AddNode(new)
```

```
i = random.randint(0,new-1)
```

```
vaz.AddEdge(new,i)
```

```
for j in range(0,new):
```

```
if(vaz.IsEdge(i,j)):
```

PE.append([new,j])

```
return vaz
```