SURA 2021 Research: Social Networks Models with Stochastic Differential Equations

Edric Eun

June 2021

1 Introduction

A social network is a graph with nodes as social actors and edges indicating relationships. These networks can help to understand changes within the group of actors or the structure of the actors. Two networks are the same if they have the same number of actors and same relationships between such actors.[3]

In social networks, relationships can be formed an destroyed, new actors can be introduced, and old actors can leave the network. If we also apply an attribute to each actor, these attributes can also change based on the structure of the network and the attributes of other actors. For example, if a person who does not watch the Superbowl have friends who watch the Superbowl, the person may be influenced to watch the Superbowl or (in an extreme case) leave the friend group. These processes of influencing and other actions on the network happen simultaneously. The influence process can be modeled with differential equations.[3]

2 Method

2.1 Attribute Evolution

In a social network with attributes for each actor, these attributes are observed a finite number of times through a period of time which is discrete in nature. All attributes of all actors can be represented in matrix z such that z_{ih} denotes the value of actor i on attribute h. We can model the attributes of actor i between two observation moments t_m and t_{m+1} with the following stochastic differential equation:

$$dZ_i(t) = \tau_m[AZ_i(t) + Bu_i(t)]dt + \sqrt{\tau_m}GdW_i(t) \tag{1}$$

$$Z_i(t_m) = z_i(t_m) \tag{2}$$

We analyze Equation 1 in simpler terms. $Z_i(t)$ is a vector of the attributes of actor i. A is called the drift and indicates how attributes affect each other. $u_i(t)$ is a vector of effects which indicate the formula that the attributes follow based on the structure of the network (i.e. attribute h of actor i changes to the average of the h attributes of actors connected to actor i), and matrix B indicates which effects are most prevalent in the network. G is the same in the SDE model as the strength of the error term indicated by the Wiener process $dW_i(t)$. [2]

2.2 SDE Solution

The solution to the SDE of modeled networks is given as

$$Z_i(t) = e^{A(t-t_m)}z_i(t_m) + \int_{t_m}^{t} e^{A(t-s)}Bu_i(s)ds + \int_{t_m}^{t} e^{A(t-s)}GdW_i(s) \tag{3}$$

[1] with the last part of the equation (the Brownian motion) having mean of a $p \times p$ zero matrix and covariance of

$$ivec[(A \otimes I_p + I_p \otimes A)^{-1}(e^{A(t-t_m)} \otimes e^{A(t-t_m)} - I_p \otimes I_p)vec(GG^T] \tag{4}$$

[1] where if vec is the operation of stacking rows of a matrix into one column, then $ivec$ is the inverse operation of taking the column and returning a square matrix, $\otimes$ is the Kronecker product, and I_p is the $p \times p$ identity matrix.[1]

2.3 Parameter Interpretation

We will refer to the individual actor stochastic differential equation parameters as a, b, and g for A, B, and G. In SDE models, it is difficult to interpret the values of a, b, and g alone. Instead, we find that the coefficients in the expected value and variance are easier to interpret. For example, the coefficient e^a is defined as the dependence of random variable Z_i to value $z_i(t_m)$, the equilibrium variance $-\frac{(g^2)}{2a}$ is interpreted as the variance of the actor's attribute, and the intercept value $-\frac{b}{2a}$ is interpreted as the limit of the actor's attribute. [2]

The period of times between the observations are not necessarily the same, so to account for the different periods of time, we attempt to normalize the real time between t_m and t_{m+1} into "model time" with parameter τ_m. For example, if $\tau_m = 1$ and $t = \tau s$ where s is the model time from 0 to 1, then the stochastic differential equation model can be represented as

$$dZ_i(s) = \tau[AZ_i(s) + Bu_i(s)]ds + GdW_i(\tau s) \tag{5}$$

We will assume $\tau_m = 1$ from here on since we only consider the attribute evolution over one period of time. [2]

3 Examples

3.1 Simulations

I first attempted to calculate dZ or the change in Z values over small increments of time t of an ordinary differential equation. This is seen in the function ODE_step in section A.1 of the appendix. The ordinary differential equations that I tested were

$$\frac{dZ_1}{dt} = \frac{1}{2}(Z_2(t) + Z_3(t)) - 2Z_1(t) \tag{6}$$

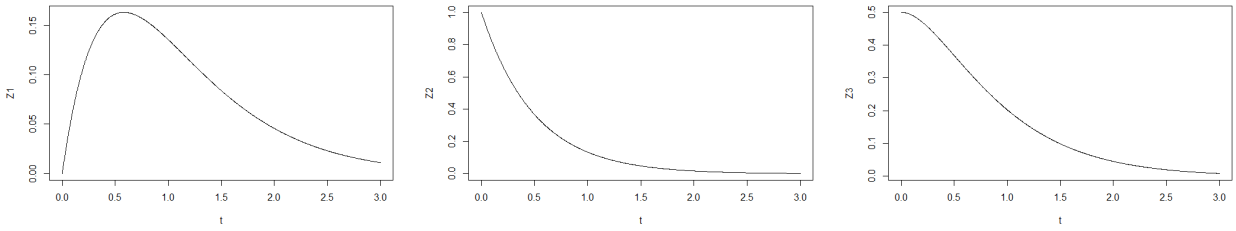
$$\frac{dZ_2}{dt} = -2Z_2(t) \tag{7}$$

$$\frac{dZ_3}{dt} = Z_2(t) - 2Z_3(t) \tag{8}$$

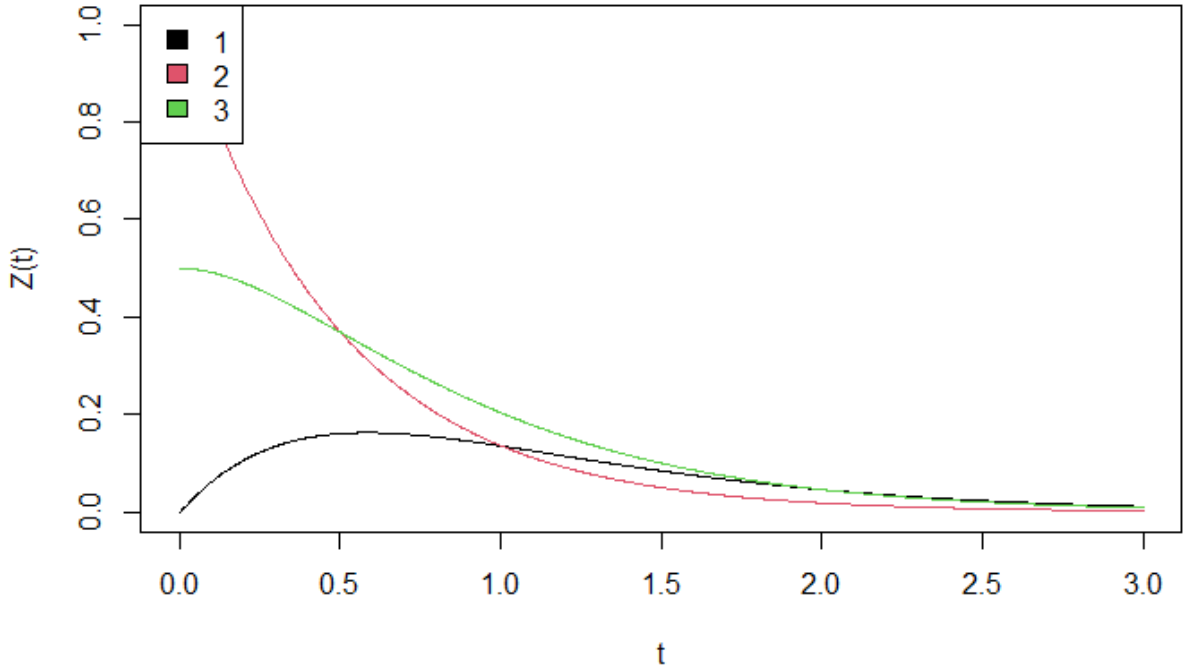
with initial values of

$$(Z_{1,0}, Z_{2,0}, Z_{3,0}) = (0, 1, 0.5) \tag{9}$$

This resulted in the following graphs:



I then attempted to write code to simulate an ordinary differential equation using integration as seen in Appendix A.2. The following graph is used with the same ordinary differential equations and initial parameters as in Equations 5, 6, and 7.



After optimizing the previous function, I coded a similar stochastic differential equation simulator with the following helper functions seen in Appendix A.3. This resulted in the code of SDE_integrate in A.4.

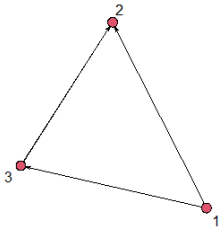
I used the same network as ODE_integrate with different value parameters in the stochastic differential equation. These stochastic differential equations have an additional intercept term.

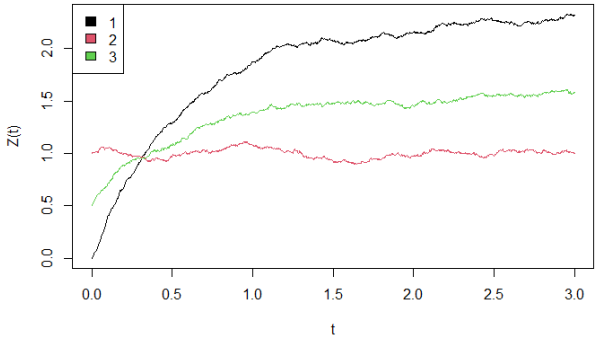
$$dZ_1(t) = -2Z_1(t) + \frac{1}{2}(Z_2(t) + Z_3(t)) + 2 + 0.1dW_1(t) \tag{10}$$

$$dZ_2(t) = -2Z_2(t) + 2 + 0.1dW_2(t) \tag{11}$$

$$dZ_3(t) = -2Z_3(t) + Z_2(t) + 2 + 0.1dW_3(t) \tag{12}$$

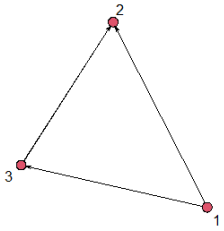
This gives the following:





The values of each actor were supposed to approach certain values, but the values were not what we expected. The problem was that I had not row-normalized the adjacency matrix. I applied the row-normalize function in Appendix A.5.

However, the shape of the graph was less than satisfactory as I wanted to see the effect of social influence. With social influence, the attribute value of each actor should all approach a certain value. As this was not the case, I tried modeling the actors by changing the value of the drift matrix A which states how the the attributes affect each other. I attempted to find the differential equation value based on the differences of the attributes with an example network graph.



$$dZ_1(t) = -aZ_1(t) + \frac{1}{2}(Z_2(t) - Z_1(t)) + \frac{1}{2}(Z_3(t) - Z_1(t)) + b + gdW_1(t) \tag{13}$$

After some calculations, I changed my A drift matrix to

$$A = (\tilde{X} - I) + a * I \tag{14}$$

where $\tilde{X}$ is the row-normalized matrix of the adjacency matrix X. This did not result in the actor attribute evolution that I was expecting, as this only changed the intercept value. Instead, I looked at a

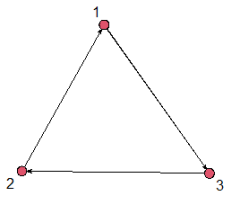
different stochastic differential equations using a different network graph.

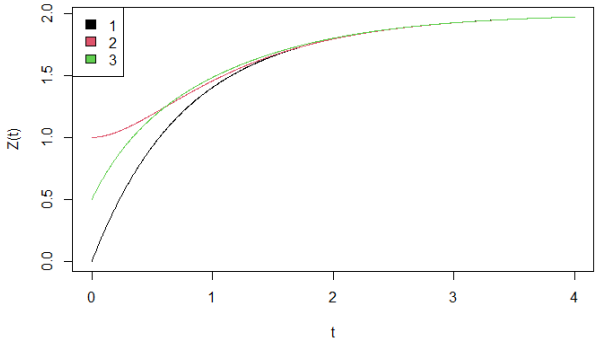
$$dZ_1(t) = -2Z_1(t) + Z_3(t) + 2 \tag{15}$$

$$dZ_2(t) = -2Z_2(t) + Z_1(t) + 2 \tag{16}$$

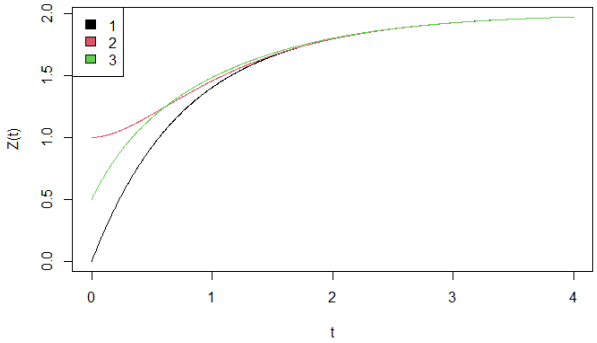
$$dZ_3(t) = -2Z_3(t) + Z_2(t) + 2 \tag{17}$$

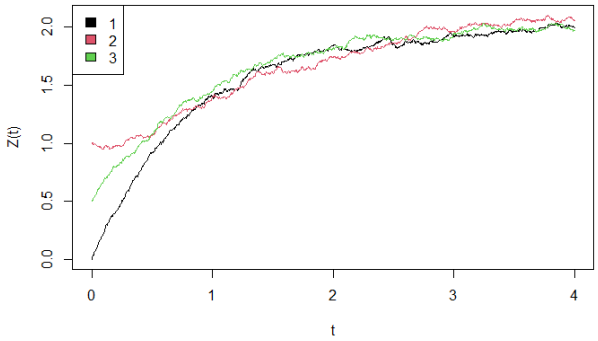
This results in the following graphs and value:





To test the stochastic term of the SDE, I set the value of g to 0.1 which resulted in the following graph on the right:

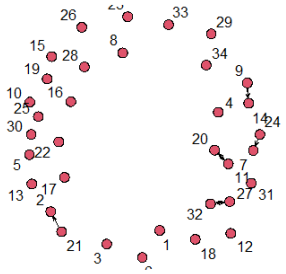


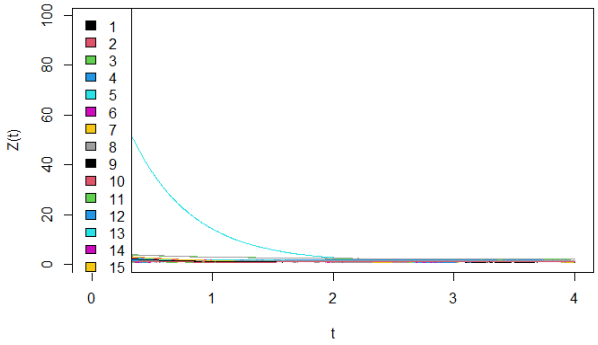


3.2 Real Data

I initially attempted to use the Glasgow data, which was a dataset consisting of 160 adolescents. This slowed the process of the SDE_integrate function, and ultimately the function was unable to run to completion due to the size of the vectors.

The Zeggelink datase was collected among 34 freshmen majoring in Sociology at the University of Gronigen in 1998. Data on gender, age, program (4-year or 2-year), and smoking was collected for each actor. The data is made publicly available and ready to use in R. I used the Zeggelink dataset with the SDE_integrate function in section A.4 to produce the following:

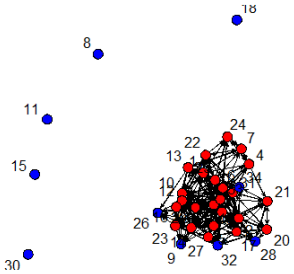


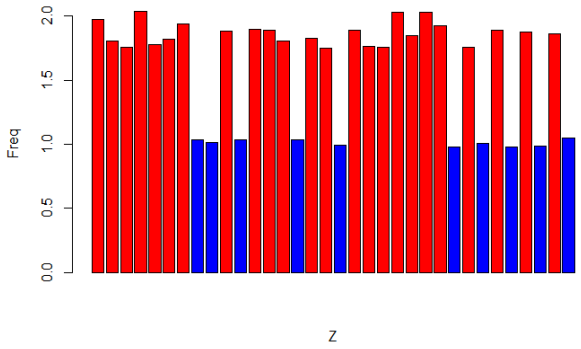


The Zeggelink **stu98t0** dataset provided an uninteresting network with almost no connections due to the fact that the data was gathered among freshmen in college at the beginning of the year. I decided to use the **stu98t6** dataset along with data on their smoking habits as initial attribute values to see more interesting results.

The SDE_integrate function is useful for simulation data as it can provide graphs of attribute values over times on small social networks. However, most real data are gathered over larger networks, and the calculations needed for SDE_integrate become CPU intensive. Another issue that occurs is the multitude of actor attributes making graphs near indecipherable. This requires a faster stochastic differential equation simulator that only calculates the final attribute values and plots.

I developed the function SDE_large found in A.6. It only calculates the final attribute values and Brownian motion and returns the values and a bar plot. The following graphs were produced on the **stu98t6** network:





There are two values that the final attribute values approximate. Most values are close to 2, which according to the Zeggelink data, means that these students smoke only at parties. There are also some values near 1, which means that these students do not smoke at all. The actors in the network graph that have an attribute value near 1 are colored blue, while the actors in the network graph that have an attribute value near 2 are colored red. While there are blue actors that are not connected to anyone in the network, there are some blue values inside the complex friend group. We notice that this is most similar to the problems faced when modeling Equations 10, 11, and 12, and that the blue actors have no out-degrees. This means that those who do not smoke do not consider anyone else to be friends.

4 Conclusion

The SURA program has been a great opportunity to do research on the topic that I am most interested in. I have done my best to work during this summer, and although there were weeks with not much progress, there were also weeks with a lot of progress. Overall, this has been a great experience working with Dr. Niezink, and I hope to do more research with her in the future. For future research, I hope to derive an estimator for the SDE parameters, implement this estimator in R and perhaps C++, and compare the method to other statistical models for social influence on networks. I also plan to use my research on real data. I have begun to do so, but I encounter the problem of the size of the vector being too large for large networks.

Appendix A: Code

A.1

```
ODE_step = function(X, z0, a, b, Q, t) {
  actors = dim(X)[1]
  dt = t/1000
  z = z0
  z_t = list()
  for (aa in 1:actors) {
    z_t[[aa]] = c(z0[aa, 1])
  }
  for (ii in 1:1000) {
    z = z + (((diag(actors)*a) + (Q*b))%*%z)*dt
    for (aa in 1:actors) {
      z_t[[aa]][ii+1] = z[aa,1]
    }
  }
  for (aa in 1:actors) {
    plot(seq(0,t, by = dt), z_t[[aa]], xlab = "t", ylab =
           paste("Z",toString(aa), sep = ""), type="l")
  }
  return(z)
}
```

A.2

```
ODE_integrate = function(z0, a, t) {
  actors = dim(a)[1]
  dt = t/1000
  t0 = 0
  z_t = list()
  for (aa in 1:actors) {
    z_t[[aa]] = c(z0[aa, 1])
```

```
  }
  for (ii in 1:1000) {
    V = expm(a*t0)%*%z0
    for (aa in 1:actors) {
      z_t[[aa]][ii+1] = V[aa,1]
    }
    t0 = t0 + dt
  }
  for (aa in 1:actors) {
    plot(seq(0,t, by = dt), z_t[[aa]], ylab = toString(aa))
  }
  return(V)
}
```

A.3

```
ivec = function(A) {
  nr = dim(A)[1]
  dim(A) = c(sqrt(nr), sqrt(nr))
  return(t(A))
}

vec = function(B) {
  B = t(B)
  dim(B) = c(dim(B)[1]*dim(B)[2], 1)
  return(B)
}

SDE_cov = function(A, t, G) {
  p = dim(A)[1]
  I_p = diag(p)
  p1 = solve(kronecker(A, I_p) + kronecker(I_p, A))
  p2 = kronecker(expm(A*t),expm(A*t)) - kronecker(I_p, I_p)
  p3 = vec(G%*%t(G))
  return(ivec(p1%*%p2%*%p3))
}
```

A.4

```
SDE_integrate = function(X, z0, t_final, a, b, g) {
    #Number of actors
    actors = dim(A)[1]

    #Graph network
```

```
plot(network(X), label = 1:actors, vertex.cex = 2)

#dt is interval for plotting
dt = t_final/1000

#Initial time is 0
t = 0

#Store z values of actors in z_t
z_t = matrix(0, nrow = 1001, ncol = actors)

#Set initial values z0 in z_t
z_t[1,] = z0[,1]

#Set initial values A, B, G
A = X + a*diag(actors)
B = b*as.matrix(rep(1, actors))
G = g*diag(actors)

#Calculate z values from equation and store
for (ii in 1:1000) {
    V = matrix(0, nrow = actors, ncol = 1)

    #Calculate Brownian motion
    W = rmvnorm(1, mean = rep(0, actors),
        sigma = SDE_cov(A, dt, G), checkSymmetry = FALSE)

    V = expm(A*dt)%*%as.matrix(z_t[ii,]) +
        solve(A)%*%(expm(A*dt) - diag(actors))%*%B + t(W)

    z_t[ii+1,] = V[,1]

    t = t + dt
}

y_min = min(z_t)
y_max = max(z_t)

#Plot all values of values of z by actor
plot(seq(0,t_final, by = dt), z_t[, 1], xlab = "t", ylab = "Z(t)",
    type="l", col = 1, ylim = c(y_min, y_max))
for (aa in 2:actors) {
    lines(seq(0,t_final, by = dt), z_t[, aa], col = aa)
}
legend("topleft", paste(1:actors), fill = 1:actors)
```

```
    #Return final value of z
    return(V)
}
```

A.5

```
    rowNormalize = function(X) {
      n = dim(X)[1]
      a = 1/rowSums(X)
      a[is.infinite(a)] = 0
      return(diag(a, nrow = n, ncol = n) %*% X)
    }
```

A.6

```
SDE_large = function(X, z0, t_final, a, b, c, g) {
  #Number of actors
  actors = dim(X)[1]

  #Graph network
  plot(network(X), label = 1:actors, vertex.cex = 2)

  #Set initial values A, B, G
  A = c*rowNormalize(X) + a*diag(actors)
  B = b*as.matrix(rep(1, actors))
  G = g*diag(actors)

  #Calculate final attribute values
  W = rmvnorm(1, mean = rep(0, actors), sigma = SDE_cov(A, t_final, G), checkSymmetry = FAI

  V = as.vector(expm(A*t_final)%*%z0 + solve(A)%*%(expm(A*t_final) - diag(actors))%*%B + t(

  #Plot final values of z by actor
  barplot(V, xlab = "Z", ylab = "Freq")

  #Return final value of z
  return(V)
}
```

References

[1] Nynke M. D. Niezink and Tom A. B. Snijders. "Co-evolution of social networks and continuous actor attributes". In: *The Annals of Applied Statistics* 11.4 (2017), pp. 1948–1973. DOI: 10.1214/17-AOAS1037. URL: https://doi.org/10.1214/17-AOAS1037.

[2] Nynke M. D. Niezink, Tom A. B. Snijders, and Marijtje A. J. van Duijn. "No Longer Discrete: Modeling the Dynamics of Social Networks and Continuous Behavior". In: *Sociological Methodology* 49.1 (2019), pp. 295–340. DOI: 10.1177/0081175019842263.

[3] Garry Robins. *Doing Social Network Research: Network-Based Research Design for Social Scientists*. Sage, 2015.