

Modelos estocásticos orientados en el actor utilizando RSiena (VI). Coevolución de redes sociales y comportamientos individuales.

Alejandro Espinosa-Rada¹ y Alvaro Uzaheta
ETH Zürich, Social Networks Lab

RESUMEN

Este texto es el último guión de una serie de seis guiones escritos en el lenguaje y ambiente de programación R que conjuntamente constituyen la introducción al software RSiena para estimar modelos estocásticos orientados en el actor para redes dinámicas. El texto está acompañado por los guiones para reproducir y estimar los modelos.

Palabras clave: *Modelos estocásticos orientados en el actor – Análisis estadísticos de redes sociales – RSiena – R.*

ABSTRACT

This article is the last document of six scripts written in the language and environment R, which introduces the RSiena software to estimate the stochastic actor-oriented models for dynamic networks. The text incorporates the scripts into the document to reproduce and estimate the models.

Key words: *Stochastic actor-oriented model – Statistical analysis for social networks – RSiena – R.*

¹ Contacto con los autores: Alejandro Espinosa-Rada (alejandro.espinosa@qess.ethz.ch)

Este documento se basa en el guión oficial de la página web de SIENA:
<https://www.stats.ox.ac.uk/~snijders/siena/Rscript04SienaBehaviour.R>

INTRODUCCIÓN

Los materiales para esta serie incluyen seis guiones (*scripts*) para utilizar el paquete **RSiena**. Dicho paquete se encuentra escrito en el entorno y lenguaje de programación **R** (2022) enfocado a la estadística computacional y de gráficos. Los guiones, a su vez, se encuentran disponibles en la página oficial¹ del programa para el análisis estadístico de datos de redes SIENA (acrónimo de *Simulation Investigation for Empirical Network Analysis*), con énfasis en redes sociales, el cual cuenta con un manual de usuarios (Ripley et al., 2022).

Este guión es la última contribución de una serie de seis guiones que introducen el software **RSiena**. Este guión brinda información adicional sobre la definición de los datos como variables en **RSiena** y de como especificar modelos para su posterior estimación. El primer guión de esta serie (Espinosa-Rada, 2022a) entrega un panorama general de cómo se estima un modelo SAOM [acrónimo para *stochastic actor-oriented model*] (mayor detalle sobre los modelos en: Snijders, 2001, 2017 y en castellano ver Espinosa-Rada, 2023). El segundo guión, en cambio, introduce el ambiente **R** y los distintos tipos de estructuras de datos que pueden ser utilizados para importar o exportar los datos para los análisis (Espinosa-Rada, 2022b). El tercer guión, en cambio, presenta un ejemplo de análisis descriptivo a realizar antes de la estimación de un modelo SAOM (Espinosa-Rada y Uzaheta, 2022). El cuarto guión amplía la información sobre la definición de datos como variables para RSiena y especificación de modelos introducida en el primer guión (Uzaheta y Espinosa-Rada, 2022). El quinto guión (Uzaheta y Espinosa-Rada, 2023), estima un modelo SAOM utilizando los datos introducidos en el segundo guión (Espinosa-Rada, 2022b).

A continuación, se preseta en el texto los códigos a ejecutar en el programa **R** antecedidos por una breve explicación. En color “(_)” se encuentran las líneas de códigos o comandos a ejecutar y/o que pertenecen a funciones propias del programa. De la misma forma, líneas de códigos antecedidos por **#** corresponden a comentarios y no comandos. Finalmente, líneas precedidas por **>** corresponden a la salida que se imprime en la

consola del código presentado en el mismo bloque.

En general, hemos investigado por qué los actores de una red emiten lazos a otros actores considerando mecanismos propios de la red en combinación con los atributos de los actores (ver por ejemplo Rivera et al., 2010; Stadtfeld y Amati, 2021). No obstante, los modelos SAOMs son lo suficientemente flexibles para también investigar cómo los lazos entre los actores inciden en el comportamiento, actitud u otros atributos de los actores en el tiempo. Al primer tipo de estudios se le suele denominar preguntas de investigación que se enfocan en los procesos de selección (Lazarsfeld y Merton, 1954), mientras que la segunda pregunta se le suele denominar investigaciones que se enfocan en procesos de influencia (Friedkin, 1998). Este guión introduce una forma de investigar simultáneamente la coevolución de procesos de selección en redes sociales y de influencia en comportamientos individuales.

Estudio sobre Amigos Adolescentes y Estilos de Vida

Al igual que en tutoriales anteriores, en este tutorial utilizaremos el estudio sobre Amigos Adolescentes y Estilos de Vida, estudio realizado en 1995 (más información en: Uzaheta y Espinosa-Rada, 2023).

A continuación, tenemos dos opciones para comenzar a trabajar con los datos en **R**. La primera opción es importar los datos directamente desde el directorio de trabajo (asumiendo que hemos especificado adecuadamente la ruta de trabajo) (más información en: Espinosa-Rada, 2022b). Por ejemplo, a continuación, importaremos las matrices de adjacencia, las covariantes y la variable dependiente correspondiente a las variables de comportamiento:

```
friend.data.w1 <- as.matrix(read.table("s50-
network1.dat")) # red1
friend.data.w2 <- as.matrix(read.table("s50-
network2.dat")) # red2
friend.data.w3 <- as.matrix(read.table("s50-
network3.dat")) # red3
drink      <-   as.matrix(read.table("s50-
alcohol.dat")) # comportamiento
smoke      <-   as.matrix(read.table("s50-
smoke.dat")) # covariantes
```

¹ Página oficial del software RSiena: <https://www.stats.ox.ac.uk/~snijders/siena/>

Una segunda opción, que tal vez es más sencilla que la anterior, es utilizar los datos que se encuentran ya incluidos en **RSiena**. De tal forma de utilizar esta opción:

```
library(RSiena)
friend.data.w1 <- s501
friend.data.w2 <- s502
friend.data.w3 <- s503
drink <- s50a
smoke <- s50s
```

Antes de continuar con el análisis, se sugiere realizar un análisis descriptivo de los datos. Por ejemplo, utilizando el paquete **sna** o funciones tales como Moran I (i.e., `ncaf()`) para ver si existe similaridad entre "friends" y las variables de comportamiento de interés, tal como es descrito en un tutorial previo de **RSiena** (Espinosa-Rada y Uzaheta, 2022). Con ello podremos tener una idea preliminar sobre los datos.

Especificando un modelo SIENA para redes que co-evolucionan

En el contexto de los modelos SAOMs es posible investigar procesos que coevolucionan. Esto puede ser para distinguir como es que dos (o más) redes diferentes, también denominadas multipléxicas (ejemplo: red de asesoramiento y de amistad), evolucionan de forma paralela y se afectan mutuamente (Snijders et al., 2013). También, para investigar la evolución de una red y los posibles cambios que una persona puede tener en el tiempo. Teóricamente, con ello es posible utilizar SAOM para distinguir entre cambios personales y sobre los vínculos para investigar procesos de "influencia" y/o de "selección" (Steglich et al., 2010). Por ejemplo: es posible distinguir si personas fumadoras se juntan con otras personas fumadoras teniendo más posibilidades de generar más lazos (i.e., "selección") o si el hecho de tener amistades que fumen aumenta la tendencia a que las personas fumen (i.e., "influencia"). Una tercera opción sería investigar tanto la co-evolución de redes como de comportamientos simultáneamente. Este tutorial es sobre el segundo tipo de coevolución.

De tal forma de especificar la co-evolución de redes y comportamiento, debemos asignar la red y la variable de interés como variables dependientes. Con respecto a la red, podemos seguir el mismo principio que en los tutoriales anteriores y especificar que la red de amistad es una de las "variables dependientes":

```
friendship <-
sienaDependent(array(c(friend.data.w1,
```

```
friend.data.w2, friend.data.w3 ), dim = c(50,
50, 3 ))# crear variable dependiente
```

Posteriormente, asignaremos la variable de comportamiento, tomar alcohol (i.e., "`drink`"), como variable dependiente:

```
drinkingbeh <- sienaDependent(drink, type =
"behavior")
```

Luego, podemos especificar el resto de las covariantes. Para este caso, utilizaremos la variable fumar:

```
smoke1 <- coCovar(smoke[ , 1 ] )
```

A continuación, definiremos el conjunto de datos para obtener algunos de los efectos:

```
myCoEvolutionData <-
sienaDataCreate(friendship, smoke1,
drinkingbeh )
```

```
myCoEvolutionEff <-
getEffects(myCoEvolutionData )
```

Posteriormente, podemos crear un reporte para evaluar si los datos fueron correctamente formateados y disponer también de un informe con algunos descriptivos:

```
print01Report(myCoEvolutionData,
modelname = 's50_3_coevol' )
```

Definiremos a continuación los efectos a incluir en el modelo de coevolución. Partiendo por los efectos estructurales los cuales deben ser incorporados utilizando el nombre corto que se encuentra en la función `effectsDocumentation(myCoEvolutionEff)`.

```
myCoEvolutionEff <-
includeEffects(myCoEvolutionEff, transTrip,
transRecTrip)
```

También podemos incluir un efecto de homofilia para la covariante constante "fumar":

```
myCoEvolutionEff <-
includeEffects(myCoEvolutionEff, simX,
interaction1 = "smoke1" )
```

Para poder distinguir entre procesos de selección o influencia (o ambos) en la tendencia a ingerir alcohol, podemos incluir efectos relacionados al emisor, receptor o efecto de homofilia en la tendencia de tomar cuando se forman amistades:

```
myCoEvolutionEff <-
includeEffects(myCoEvolutionEff, egoX, altX,
simX, interaction1 = "drinkingbeh" )
```

Para la parte de influencia (i.e., el efecto de la red en el comportamiento de los estudiantes), podemos especificar los siguientes efectos: grados de entrada (`indegree`), grados de salida (`outdegree`) y la asimilación en el efecto de ingerir alcohol (otras especificaciones son posibles):

```
myCoEvolutionEff <-
  includeEffects(myCoEvolutionEff, name =
  "drinkingbeh", avAlt,indeg, outdeg,
  interaction1 = "friendship")
```

Ahora podemos ver qué efectos fueron incluidos:

```
myCoEvolutionEff
```

Por último, podemos definir la configuración del algoritmo. A menos que tengan otras razones, las opciones por defecto debiesen ser adecuadas. Sin embargo, procure especificar un nombre de archivo (en este caso: "`'s50coevol_3'`") en donde pueda entregar los estuldaos en formato de texto:

```
myCoEvAlgorithm <-
  sienaAlgorithmCreate(projname =
  's50coevol_3')
```

Finalmente, estimaremos el modelo (paréntesis en la función permite ejecutar los resultados directamente en la pantalla):

```
(ans <- siena07(myCoEvAlgorithm, data =
  myCoEvolutionData, effects =
  myCoEvolutionEff))
```

Resultados del modelo

Los resultados entregan distintos indicadores a ser considerados. Entre ellos, "*convergence t-ratio*" no corresponde al valor estadístico t para evaluar niveles de significancia estadística, sino que es utilizado para evaluar la convergencia del modelo (ver sección 6.3. en Ripley et al., 2022). En el contexto de los modelos SAOMs, valores adecuados de convergencia debiesen ser menores a 0,1 en valores absolutos y el "*maximum convergence ratio*" menor a 0,25. Si estos indicadores no se cumplen, se recomienda volver a estimar el modelo considerando la última estimación realizada (esto se realiza especificando el parámetro "`prevAns`"). Como ejemplo:

```
(ans1 <- siena07(myCoEvAlgorithm, data =
  myCoEvolutionData, effects =
  myCoEvolutionEff, prevAns = ans))
```

Este proceso puede realizarse nuevamente si es necesario.

Dado el tamaño de este conjunto de datos, el modelo sobre las dinámicas de comportamiento se encuentra sobre-especificadas, generando que algunos errores estándares sean muy grandes. Para este caso, es aconsejable eliminar los efectos de grados en la variable de comportamiento, dado que ésta no contiene suficiente información para realizar una buena estimación.

Para excluir efectos (en este caso `indeg` y `outdeg`):

```
myCoEvolutionEff2 <-
  includeEffects(myCoEvolutionEff, name =
  "drinkingbeh", indeg, outdeg, interaction1 =
  "friendship", include = FALSE)
```

Un modo alternativo para fijar y testear las variables puede realizarse especificando la siguiente función:

```
myCoEvolutionEff3 <-
  includeEffects(myCoEvolutionEff, name =
  "drinkingbeh", indeg, outdeg, interaction1 =
  "friendship", fix = TRUE, test = TRUE)
```

A través de dicho comando los parámetros quedarán fijos. Para ver que estos efectos estén efectivamente fijados, es posible observar el contenido del objeto `myCoEvolutionEff3` (nombre asignado en este tutorial) que en este caso debiera ser igual a cero. Este comando permitirá realizar un *score-type test*.

```
(ans3 <- siena07(myCoEvAlgorithm, data =
  myCoEvolutionData, effects =
  myCoEvolutionEff3))
```

Los errores estándar en el parámetro `avAlt` ahora serán más pequeños, debido a que existía una alta correlación entre los efectos de `indeg` y `outdeg` los cuales ya no son parte del modelo en esta nueva especificación.

En `ans3` se puede observar los resultados del *score-type test*. Estos resultados también pueden explorarse a través del siguiente comando:

```
score.Test(ans3)
```

También se puede explorar los valores del *score-type test* de forma separada:

```
score.Test(ans3, 15)
```

```
score.Test(ans3, 16)
```

Para familiarizarse de mejor manera con la especificación, sugerimos probar esta técnica con otros parámetros y ver qué pasa.

Especificando el modelo incorporando efectos adicionales

Los efectos disponibles para este conjunto de datos pueden ser obtenidos a través de la siguiente función:

```
effectsDocumentation( myCoEvolutionEff )
```

En Ripley et al. (2022), capítulo 12, se encuentra la expresión matemática y el significado de estos efectos.

Para estudiar el efecto directo del efecto de la covariante fumar sobre la variable dependiente de beber alcohol (nuestra variable de comportamiento en este ejemplo), se puede utilizar el efecto `effFrom`:

```
myCoEvolutionEff4 <-  
  includeEffects(myCoEvolutionEff2, name =  
    "drinkingbeh", effFrom, interaction1 =  
    "smoke1")
```

Dado que ahora tenemos un buen resultado sobre este modelo simple, puede ser útil partir la estimación considerando el último modelo explorado en este tutorial:

```
(ans4 <- siena07(myCoEvAlgorithm, data =  
  myCoEvolutionData, effects =  
  myCoEvolutionEff4, prevAns = ans3))
```

En algunos casos, la convergencia puede no ser idónea, en cuyo caso podemos continuar la estimación si es necesario:

```
(ans4 <- siena07(myCoEvAlgorithm, data =  
  myCoEvolutionData, effects =  
  myCoEvolutionEff3, prevAns = ans4))
```

Para exportar los datos de los resultados en formato LaTeX:

```
siena.table(ans4)
```

No obstante, si en cambio se quisiera tener un formato que pueda ser importado en programas tales como MS-Word, entonces podemos utilizar la misma función pero exportando en formato `html`:

```
siena.table(ans4, type="html")
```

Tablas de Influencia

Ahora mostraremos cómo obtiene una visualización para ver los procesos de influencia, tal como aparece en la sección 13.4 en Ripley et al. (2022).

Para realizar estas visualizaciones, utilizaremos una serie de comandos adicionales que deberán cargarse con anterioridad en R.

```
source("http://www.stats.ox.ac.uk/~snijders/  
siena/InfluenceTables.r")
```

De la misma forma, para realizar la visualización también necesitaremos el paquete "ggplot2". Si ya ha sido descargado con anterioridad no es necesario instalarlo, en caso contrario, utilizar la siguiente función: `install.packages("ggplot2")`. Una vez que la librería se encuentra disponible en su computadora, podrá llamar el paquete para disponer funciones adicionales para proceder con la visualización:

```
library(ggplot2)
```

Como se encuentra especificado en las últimas líneas del guión en `InfluenceTables.r`, especificaremos algunos parámetros adicionales:

```
behname <- "drinkingbeh"  
netname <- "friendship"  
levls <- 1:5 # The levels of the dependent actor  
variable drinkingbeh  
png(filename=paste("influenceTable_ ", behna-  
me, ".png", sep=""), width=1000, height=800)  
influenceTable.plot(ans4, myCoEvolutionData,  
netname, behname, levls, separation=0)  
graphics.off()
```

Utilizando estas funciones podremos generar una visualización con formato png que será guardado en su directorio de trabajo, con el nombre especificado en la función anterior.

Si quisiera saber más sobre tablas de influencia y cómo se realizan las visualizaciones correspondientes, por favor revisar el guión de `InfluenceTables.r`.

Comentarios finales

En este guión se revisó cómo especificar un modelo de co-evolución que considera tanto la evolución de una red como los cambios en una variable individual en el software **RSiena**. Para ello, se estimaron un conjunto de modelos, se incorporaron, eliminaron y se controló a través de `score-type test` distintos efectos. Finalmente se realizó un conjunto de estimaciones para disponer de un último modelo que fue exportado para luego ser reportado. Finalmente, se presentó brevemente el uso de las tablas de influencia de tal forma de ser utilizadas en la interpretación de parámetros.

Esperamos que esta introducción permita dar una aproximación a los modelos estocásticos basados en el actor a través del uso práctico del programa, acorde al estado actual de su desarrollo y que sea accesible a la comunidad que habla español.

Agradecimientos

Agradecemos particularmente a Tom A.B. Snijders (University of Oxford y University of Groningen) por su amabilidad en permitirnos difundir, modificar y extender los siguientes guiones los cuales contaron también con contribuciones previas de Robin Gauthier, Ruth Ripley, Johan Koskinen, Paulina Preciado, Zsófia Boda y Christian Steglich. En un principio estos documentos iban ser una traducción textual a los guiones disponibles en la página de RSiena. No obstante, por sugerencia de Tom A. B. Snijders los guiones traducidos y modificados adquieren con esta contribución independencia de sus homólogos en inglés con el objeto de facilitar su difusión, mantención y responsabilidad en su contenido en español del cual esperamos reflejar adecuadamente sus intenciones originales.

REFERENCIAS

Espinosa-Rada, Alejandro (2022a). Modelos estocásticos orientados en el actor utilizando RSiena (I): Guión básico introductorio. *Redes: revista hispana para el análisis de redes sociales*, 33(1), 0092-99. DOI: <https://doi.org/10.5565/rev/redes.936>

Espinosa-Rada, Alejandro (2022b). Modelos estocásticos orientados en el actor utilizando RSiena (II): Formato de los Datos. *Redes. Revista hispana para el análisis de redes sociales*, 33(1), 100-111. DOI: <https://doi.org/10.5565/rev/redes.937>

Espinosa-Rada, Alejandro (2023). "Usos contemporáneos de los modelos estocásticos orientados en el actor (SAOMs) para redes dinámicas". En: Francisca Ortiz y Alejandro Espinosa-Rada (Eds.). *Redes Sociales: teoría, métodos y aplicaciones en América Latina*. CIS-Madrid. *En prensa*.

Espinosa-Rada, Alejandro, y Uzaheta, Alvaro (2022). Modelos estocásticos orientados en el actor utilizando RSiena (III): Análisis Descriptivo. *Redes. Revista hispana para el análisis de redes sociales*, 33(2), 196-202. DOI: <https://doi.org/10.5565/rev/redes.956>

Friedkin, Noah E. (1998). A structural theory of social influence. Cambridge University Press.

Lazarsfeld, Paul F. and Merton, Robert K. (1954). Friendship as a social process: A substantive and methodological analysis. En: Morroe Berger, Theodore Abel y Charles H. Page (Eds.) *Freedom and Control in Modern Society* (pp. 18-66). D. Van Nostrand Company.

R Core Team (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: <https://www.R-project.org/>

Ripley, Ruth; Snijders, Tom A.B.; Boda, Zsófia; Vörös, András y Preciado, Paulina (2022). *Manual for SIENA version 4.0*. Oxford: University of Oxford, Department of Statistics, <http://www.stats.ox.ac.uk/~snijders/siena/>.

Rivera, Mark T., Soderstrom, Sara B., y Uzzi, Brian (2010). Dynamics of dyads in social networks: Assortative, relational, and proximity mechanisms. *annual Review of Sociology*, 36, 91-115. DOI: <https://doi.org/10.1146/annurev.soc.34.040507.134743>

Snijders, Tom A.B. (2001) The statistical evaluation of social network dynamics, *Sociological Methodology*, 40: 361-395.

Snijders, Tom A. B., Van de Bunt, Gerhard G., y Steglich, Christian E. (2010). Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1), 44-60. DOI: <https://doi.org/10.1016/j.socnet.2009.02.004>

Snijders, Tom A. B., Lomi, Alessandro y Torló, Vanina J. (2013). A model for the multiplex dynamics of two-mode and one-mode networks, with an application to employment preference, friendship, and advice. *Social Networks*, 35(2), 265-276. DOI: <https://doi.org/10.1016/j.socnet.2012.05.005>

Snijders, Tom A.B. (2017). Stochastic Actor-Oriented Models for Network Dynamics. *Annual Review of Statistics and Its Application*, 4, 343-363. DOI: <http://dx.doi.org/10.1146/annurev-statistics-060116-054035>
Here is the e-print access to this article.

Stadfeld, Christoph, y Amati, Viviana (2021). Network mechanisms and network models. En: Gianluca Manzo (Ed.). *Research Handbook on Analytical Sociology* (pp. 432-452). Edward Elgar Publishing. DOI: <https://doi.org/10.4337/9781789906851.00032>

Steglich, Christian, Snijders, Tom A. B., y Pearson, Michael. (2010). Dynamic networks and behavior: Separating selection from influence. *Sociological methodology*, 40(1), 329-393. DOI: <https://doi.org/10.1111/j.1467-9531.2010.01225.x>

Uzaheta, Alvaro y Espinosa-Rada, Alejandro (2022). Modelos estocásticos orientados en el actor utilizando RSiena (IV): Formato de las variables. *Redes: revista hispana para el análisis de redes sociales*, 33(2), 0203-209. DOI: <https://doi.org/10.5565/rev/redes.957>

Uzaheta, Alvaro y Espinosa-Rada, Alejandro (2023). Modelos estocásticos orientados en el actor utilizando RSiena (V): Estimación de un modelo para la evolución de redes sociales. *Redes: revista hispana para el análisis de redes sociales*, 34(1), 120-125.

Remitido: 03-11-2022

Corregido: 04-11-2022

Aceptado: 05-11-2022



© Los autores