Biblioteca Digital
F C E N - U B A

BIBLIOTECA CENTRAL LUIS F LELOIR
BIBLIOTECA CENTRAL LELOIR
FACULTAD DE CIENCIAS EXACTAS Y NATURALES UBA

Tesis de Posgrado

Procesos de ramificación con tipos no acotados y simulación de redes con pérdida

Tetzlaff, Guillermo Tomás

2004

Tesis presentada para obtener el grado de Doctor en Ciencias de la Computación de la Universidad de Buenos Aires

Este documento forma parte de la colección de tesis doctorales y de maestría de la Biblioteca Central Dr. Luis Federico Leloir, disponible en digital.bl.fcen.uba.ar. Su utilización debe ser acompañada por la cita bibliográfica con reconocimiento de la fuente.

This document is part of the doctoral theses collection of the Central Library Dr. Luis Federico Leloir, available in digital.bl.fcen.uba.ar. It should be used accompanied by the corresponding citation acknowledging the source.

Cita tipo APA:
Tetzlaff, Guillermo Tomás. (2004). Procesos de ramificación con tipos no acotados y simulación de redes con pérdida. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. http://digital.bl.fcen.uba.ar/Download/Tesis/Tesis_3709_Tetzlaff.pdf

Cita tipo Chicago:
Tetzlaff, Guillermo Tomás. "Procesos de ramificación con tipos no acotados y simulación de redes con pérdida". Tesis de Doctor. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. 2004. http://digital.bl.fcen.uba.ar/Download/Tesis/Tesis_3709_Tetzlaff.pdf

EXACTAS UBA
Facultad de Ciencias Exactas y Naturales

UBA
Universidad de Buenos Aires

Dirección: Biblioteca Central Dr. Luis F. Leloir, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. Intendente Güiraldes 2160 - C1428EGA - Tel. (++54 +11) 4789-9293

Contacto: digital@bl.fcen.uba.ar

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Tesis de doctorado

Título:

Procesos de ramificación con tipos no acotados y simulación de redes con pérdida

Autor: Guillermo Tomás Tetzlaff

Director: Prof. Dr. Pablo A. Ferrari (Universidade de São Paulo)

№3709

A la memoria del profesor Jorge Samur

Procesos de ramificación con tipos no acotados y simulación de redes con pérdida

Abstract

Esta tesis extiende la teoría de procesos estocásticos de ramificación al caso de tipos no acotados. Éstos surgen de no asumir que el número esperado de hijos es una función acotada del tipo del padre. Tampoco se supone que existe un entero m tal que se tiene una cota inferior positiva, uniforme sobre el tipo del ancestro, para la probabilidad de que una población esté extinguida en la m-ésima generación. Demostramos que una condición más débil que la existencia de tal m lleva a la extinción con probabilidad 1 si la sucesión de las esperanzas de los tamaños de las generaciones no tiende a infinito. También se obtienen criterios que aseguran una probabilidad positiva de que no haya extinción. Se proveen ejemplos extendiendo a nuestro contexto de tipos no acotados dos aplicaciones bien conocidas: la dinámica poblacional de Leslie y los procesos de ramificación asociados a percolación continua. Ésta última se desarrolla hasta obtener resultados sobre percolación continua orientada, que dan condiciones suficientes de factibilidad para la simulación exacta de redes con pérdida con objetos no uniformemente acotados. Luego de resolver el problema de la generación de clusters de percolación con este tipo de objetos, se proponen algoritmos de simulación de percolación y de una red con pérdida que se implementa en lenguaje C.

Unbounded-types branching processes and loss networks simulation

Abstract

This thesis extends the theory of stochastic branching processes to the case of unbounded types. These arise from not assuming that the expected number of children is a bounded function of the parent's type. It is neither supposed that there exists an integer m such that there is a lower positive bound, uniform over the ancestor's type, for the probability that a population is extinct at the m-th generation. We prove that a weaker condition than the existence of such an m leads to extinction almost surely if the sequence of expected generation sizes does not tend to infinity. Some criteria for a positive probability of nonextinction are also obtained. Examples are provided by extending to our unbounded-types setting two well known applications, namely Leslie population dynamics and processes associated to continuum percolation. The latter application is developed until results are obtained about oriented continuum percolation, which give conditions of feasibility for the exact simulation of loss networks with not uniformly bounded objects. After solving the problem of generating percolation clusters with this kind of objects, algorithms are proposed for the simulation of percolation and of a loss network, which is implemented in C language.

Contenido

Introducción

Primera Parte: Procesos de ramificación con tipos no acotados

Capítulo 1: Procesos de ramificación
1.1 Procesos de ramificación de tiempo discreto. Tipos no acotados
1.2 Resultados conocidos para procesos con un único tipo
1.3 Fórmula iterativa general para las medias

Capítulo 2: Condiciones para extinción casi segura
2.1 Criterio de la esperanza del número total de descendientes
2.2 Criterio basado en acotación condicional de tipos

Capítulo 3: Condiciones para probabilidad positiva de no extinción.
3.1 Procesos con un solo tipo
3.2 Criterios para infinitos tipos no uniformemente acotados

Capítulo 4: Extensión a tipos no acotados de aplicaciones discretas y continuas
4.1 Dinámica poblacional de Leslie
4.2 Procesos que dominan clusters de percolación continua

Segunda Parte: Simulación exacta de redes con pérdida con objetos no acotados

Capítulo 5: Simulación de clusters de percolación continua
5.1 Percolación continua de esferas uniformemente acotadas
5.2 Percolación continua de esferas no uniformemente acotadas
5.3 Valor de λ que asegura finitud de los clusters

Capítulo 6: Simulación de clusters de percolación continua orientada
6.1 Percolación orientada de rectángulos en el plano
6.2 Valor de λ que asegura finitud de los clusters

Capítulo 7: Simulación de redes con pérdida mediante el algoritmo Backward-Forward
7.1 Redes con pérdida
7.2 El Algoritmo Backward-Forward
7.3 Simulación de llamadas no uniformemente acotadas en un cable infinito

Referencias

Introducción

Los procesos estocásticos de ramificación surgen como resultado de aplicar la Teoría de Probabilidad a fenómenos de dinámica poblacional. Mode (1971) describe de la siguiente manera el inicio de este área de investigación.

En la década de 1870 en Inglaterra, el especialista en biometría Francis Galton estaba interesado en la declinación y extinción de familias que en el pasado habían ocupado posiciones relevantes. La explicación que se ofrecía decía que tanto el bienestar físico como la capacidad intelectual van acompañados por una disminución en la fertilidad. Sin embargo de Candolle (1873) señalaba que una alta proporción de apellidos debe extinguirse debido a las leyes del azar y que mientras no se sepa cuál es esta proporción, no se puede decidir si la extinción de apellidos famosos se debe o no a una fertilidad disminuida. Galton (1873) le dió al problema una formulación matemática que fue publicada por *Educational Times*. Allí se preguntaba por la proporción de apellidos que van desapareciendo a lo largo de las generaciones en una población en la que sólo se consideran los varones adultos. El planteo indicaba que un porcentaje a_0 de los individuos no tiene hijos, a_1 tiene un hijo y así hasta a_5, que es el porcentaje de individuos que tienen cinco hijos.

Galton recibió una sola respuesta al problema, que resultó no satisfactoria, y persuadió al clérigo y matemático Henry William Watson para que abordara el tema. Watson (1873) lo resolvió parcialmente pero dejó inaugurado el estudio de los procesos de ramificación que reciben este nombre a partir del trabajo de Kolmogorov y Dmitriev (1947). Cabe mencionar que según Guttorp (1991), una presentación de Iréneé-Jules Bienaymé en la Societé Philomatique de Paris en 1845 sería un antecedente previo al de Galton y Watson para el enfoque matemático del problema de extinción.

El problema planteado más arriba corresponde a un caso de ramificación con un solo *tipo* porque todos los individuos involucrados producen descendencia siguiendo la misma distribución de probabilidad. Este esquema se fue generalizando a varios y hasta a infinitos tipos. El tipo es la clase a la que pertenece un individuo, según la distribución de probabilidad que les corresponde para producir hijos de los distintos tipos. La teoría encontró numerosas aplicaciones en demografía, biología, física (reacciones en cadena en sustancias radiactivas) y en computación (algoritmos de almacenamiento de datos, Biggins 1997, y dinámica de la producción de copias de archivos, Dawson y Wu 1997).

En todo proceso de ramificación interesa encontrar condiciones que aseguran que la probabilidad de extinción de una población sea 1 (lo que en probabilidades se denomina extinción casi segura) y condiciones para que sea menor que 1. El presente trabajo aporta en su Primera Parte, resultados a esta cuestión para el caso de *tipos no acotados*, concepto que luego se hará preciso y que se refiere a procesos en los que hay un conjunto infinito de tipos y éstos pueden ser arbitrariamente grandes en cuanto a su capacidad de dejar descendencia. Estos procesos no habían sido abordados aún en la literatura de procesos de ramificación. Aquí se presentan ejemplos que resultan de extender al caso no acotado aplicaciones conocidas con conjuntos de tipos numerables y no numerables.

Los procesos con tipos no acotados se aplican a la obtención de condiciones de factibilidad del algoritmo de simulación exacta de redes con pérdida de Fernández, Ferrari y García (1998, 2001, 2002) para el caso en que los objetos involucrados tienen tamaño no uniformemente acotado. La Segunda Parte de esta tesis desarrolla esta posibilidad y lleva a cabo una simulación de llamadas no uniformemente acotadas en una ventana finita de un cable infinito. Esta simulación extiende el trabajo de Maric (2002), en el cual las llamadas están uniformemente acotadas.

Primera Parte:

Procesos de ramificación con tipos no acotados

Capítulo 1

Procesos de ramificación.

1.1 Procesos de ramificación de tiempo discreto. Tipos no acotados

Los procesos de ramificación ocupan un capítulo relevante en la Teoría de Probabilidades. Importantes obras sobre el tema son por ejemplo el libro de Harris (1963) y el de Athreya y Ney (1972).

En su versión más simple, describen la descendencia de un individuo que tiene hijos iguales a él en cantidad aleatoria (0, 1 o más), siguiendo una distribución probabilística. Estos nuevos individuos tienen a su vez hijos en número aleatorio independiente pero siguiendo la misma distribución y así sucesivamente. La población así generada puede extinguirse en un número finito de generaciones o extenderse a lo largo de infinitas generaciones. Todos los individuos son aquí de *un mismo tipo* en el sentido que producen descendencia siguiendo la misma ley de probabilidad.

En los procesos *multitipo con finitos tipos* $(1,2,...,n)$ hay más de una distribución de probabilidad. Los individuos son de distinto tipo, según cuál de estas distribuciones les corresponde. Dada una composición inicial de la población, el árbol aleatorio de las sucesivas generaciones está entonces completamente definido por las probabilidades de la forma P(el tipo i tiene k hijos de tipo j) para $1 \leq i,j \leq n$ y $k = 0,1,2,....$ Aquí también el proceso que sigue la descendencia de un individuo siempre es independiente del que sigue su hermano, lo cual, aunque significa una limitación para la posibilidad de modelar algunas situaciones reales, es la característica de los procesos de ramificación.

Cuando el *conjunto de tipos es infinito numerable*, tendremos probabilidades análogas a las anteriores, mientras que con *no numerables tipos*, tendremos un continuo de tipos y esas probabilidades puntuales pueden ser todas iguales a cero. En este caso hablamos de la probabilidad de que un individuo de tipo x tenga un cierto número de descendientes dentro de cierto subconjunto del conjunto de tipos.

Resulta de especial interés obtener condiciones necesarias y suficientes para que sea igual a 1 la probabilidad de extinción, es decir, la probabilidad de que en un número finito de generaciones no quede individuo alguno en la población.

Para deducir tales condiciones se utiliza, principalmente, una información más resumida que el conjunto de probabilidades de más arriba, cual es el conjunto de esperanzas de la forma E(número de hijos de tipo j que da un individuo de tipo i) en el caso numerable, y de la forma E(número de hijos con tipos en A que tiene un individuo de tipo x) en el caso no numerable, donde A es un subconjunto medible de X, el conjunto de tipos.

Tipos no acotados

Cuando el número de tipos es infinito (numerable o no) puede darse el caso común en ciertas aplicaciones, que no haya una cota uniforme sobre los tipos para la esperanza del número de hijos que un individuo puede producir. Diremos que un tal proceso tiene *tipos no acotados*. Se trata del caso en que hay tipos arbitrariamente grandes en cuanto a su capacidad de producir descendencia. Existe otra propiedad que puede tener un proceso de ramificación y que justifica que sus tipos se denominen no acotados. Esta se refiere a la capacidad de producir descendencia a largo plazo haciendo poco probable la extinción a lo largo de las sucesivas generaciones, aunque la descendencia no sea mucha en la primera generación. Expresaremos más rigurosamente ambas propiedades luego de introducir notación.

Sea $Z_0, Z_1, ..., Z_n, Z_{n+1}, ...$ un proceso de ramificación de tiempo discreto con tipos en un conjunto X, numerable o no. Notemos que los valores que adopta la variable aleatoria Z_n deben caracterizar tanto al número de individuos presentes en la generación n como a sus tipos. Ahora bien, el número de individuos con tipo en un conjunto $A \subset X$ en la n-ésima generación se denotará con $Z_n(A)$. Los símbolos $P_x()$, E_x y q_x se usarán para probabilidades, esperanzas y probabilidades de extinción de un proceso que comienza con un único individuo de tipo x. La no acotación de las esperanzas a la que hacíamos referencia es entonces

$$\sup_{x \in X} E_x Z_1(X) = \infty$$

mientras que la no acotación que podemos encontrar en cuanto a la capacidad de producir descendencia a largo plazo la podemos expresar como

$$\inf_{x \in X} P_x(Z_m(X) = 0) = 0$$

para todo entero positivo m. Los resultados conocidos sobre la obtención de condiciones de extinción casi segura (el problema de criticalidad) usan el hecho de que ninguna de las condiciones precedentes se verifica.

El presente trabajo amplía la teoría de procesos de ramificación a procesos con tipos no acotados y proporciona una aplicación de los resultados obtenidos a la simulación de objetos interactuantes en el espacio euclídeo $\mathbb{R}^d$. En nuestros ejemplos y aplicaciones los tipos no son acotados en ninguno de los dos sentidos que expusimos, pero cabe señalar que ninguna de las dos condiciones implica la otra. Para ello podemos dar ejemplos. Para ver que la primera condición de no acotación de los tipos no implica la segunda considérese un proceso con tipos $x_0, x_1, ...$ y probabilidad de que x_k ($k = 0, 1, ..$) tenga k hijos de tipo x_0 igual a 1, mientras que toda otra probabilidad de dar descendencia es cero. Se trata de un caso bastante trivial en que los distintos tipos dan cantidades crecientes de individuos x_0 pero estos no tienen descendencia. Para ver que la segunda no implica la primera tenemos un proceso de un solo tipo en que cada individuo da otro individuo con probabilidad 1.

1.2 Resultados conocidos para procesos con un único tipo

Abordamos aquí los procesos de ramificación con un único tipo, exponiendo con demostración los resultados conocidos sobre condiciones para extinción casi segura y sobre

condiciones que aseguran no extinción con probabilidad positiva. Estos últimos serán utilizados más adelante para obtener nuestros resultados en el caso de infinitos tipos no acotados. Los procesos con un tipo y sus propiedades aparecen en numerosos textos como por ejemplo Harris (1963), Karlin (1966), Athreya y Ney (1972), Woodroofe (1975) y Ross (1983).

Consideremos una población en la que los individuos tienen hijos del mismo tipo que los padres. Supongamos que cada individuo, entre su nacimiento y el final de su vida, tiene una cantidad j de hijos ($j = 0, 1, 2, \ldots$) con probabilidad p_j, independientemente del número de hijos que produce cualquier otro individuo. Sea m la media de esta distribución. El número de individuos presente inicialmente se denotará S_0. Consideraremos esta cantidad inicial como no aleatoria e igual a 1. Los hijos de este individuo constituyen la primera generación, cuya cantidad de integrantes se denota con S_1. Se tiene entonces $p_j = P(S_1 = j)$ y $m = E(S_1)$. A su vez, el tamaño del conjunto de hijos de los individuos de la primera generación es S_2 y así sucesivamente.

Tenemos entonces

$$S_0 = 1$$

y para $n \geq 1$

$$S_n = \sum_{i=1}^{S_{n-1}} R_{n-1,i},$$

donde $R_{n-1,i}$ denota el núnero de hijos del individuo i-ésimo de la generación $n-1$.

Fórmula iterativa para las medias

Condicionando respecto de S_{n-1},

$$\begin{aligned} E(S_n) &= E(E[S_n|S_{n-1}]) \\ &= E(mS_{n-1}) \\ &= mE(S_{n-1}) \\ &= m^2E(S_{n-2}) \\ &= m^n. \end{aligned}$$

Probabilidad de extinción

Sea q la probabilidad de que una población que inicialmente tiene un solo individuo se extinga en algún momento. Condicionando respecto de todos los posibles números de hijos que puede producir aquel individuo inicial obtenemos la ecuación

$$q = P(extinción) = \sum_{j=0}^{\infty} P(extinción|S_1 = j)p_j.$$

Notemos ahora que si hay j individuos en la primera generación, la población se extinguirá si y sólo si las respectivas poblaciones a las que ellos dan origen se extinguen. Por el supuesto de independencia resulta entonces que la probabilidad de extinción satisface

$$q = \sum_{j=0}^{\infty} q^j p_j.$$

Definiendo la función generatriz $g(x) = \sum_{j=0}^{\infty} x^j p_j$ nos queda que q debe satisfacer

$$q = g(q).$$

Utilizando esta ecuación se puede probar que si $p_0 > 0$ y $p_0 + p_1 < 1$ se tiene que $q = 1$ si y sólo si $m \leq 1$. Este es el resultado más conocido de procesos de ramificación y será invocado en uno de nuestros teoremas. Daremos aquí una demostración. Señalemos antes que si $p_0 = 0$, está claro que $q = 0$, y que si $p_0 > 0$ pero $p_0 + p_1 = 1$, la extinción es casi segura porque el número de generaciones hasta la extinción tiene distribución geométrica.

Demostración de $q = 1$ si y sólo si $m \leq 1$

Vimos que q debe satisfacer $q = g(q)$. Probaremos en primer lugar que q es la menor de las soluciones de $x = g(x)$. Sea $x \geq 0$ tal que satisface dicha ecuación. Veremos por inducción que para todo n se tiene $x \geq P(S_n = 0)$. Ahora

$$P(S_1 = 0) = p_0 = x^0 p_0 \leq g(x) = x,$$

mientras que

$$P(S_{n+1} = 0) = \sum_{j=0}^{\infty} P(S_{n+1} = 0|S_1 = j)p_j = \sum_{j=0}^{\infty} (P(S_n = 0))^j p_j.$$

Asumiendo por hipótesis inductiva que $P(S_n = 0) \leq x$,

$$P(S_{n+1} = 0) \leq \sum_{j=0}^{\infty} x^j p_j = g(x) = x.$$

Ahora con $x \geq P(S_n = 0)$ y dado que las $P(S_n = 0)$ forman una sucesión creciente por inclusión de los eventos $\{S_n = 0\}$, tomamos límite para obtener

$$x \geq \lim_n P(S_n = 0) = P(\cup_n \{S_n = 0\}) = q,$$

como queríamos.

Para ver ahora que $q = 1$ si y sólo si $m \leq 1$ empezamos por suponer que $m > 1$. Para obtener $q < 1$ basta probar que $g(x) - x$ tiene una raíz menor que 1. Utilizaremos el hecho de que la derivada de g por la izquierda cumple con

$$g'(1-) = \sum_{j=1}^{\infty} jp_j = m.$$

Analicemos la función $g(x) - x$ en el intervalo $[0, 1]$. Por ser su derivada por la izquierda en 1 igual a $g'(1-) - 1 = m - 1 > 0$ y dado que $g(1) - 1 = 0$, existe $x_0 \in (0, 1)$ tal que $g(x_0) - x_0 < 0$. Además $g(0) - 0 = p_0 > 0$, por nuestra suposición inicial. Luego por la continuidad de $g(x) - x$, existe un punto entre 0 y x_0 en el que la función se anula, y como q es la menor solución de $g(x) = x$, $q < 1$.

En el caso $m \leq 1$ tenemos $g'(1-) = m \leq 1$. Además, como supusimos $p_0 + p_1 < 1$ también tendremos $g''(x) = \sum_{j=2}^{\infty}(j-1)jx^{j-2}p_j > 0$ para todo $x \in (0, 1)$ y por lo tanto g' es estrictamente creciente en $[0, 1)$. Luego $g'(x) < 1$ para todo $x \in [0, 1)$. Ahora, dado cualquier x en $[0, 1)$, por el Teorema del Valor Medio, para algún $\xi \in (x, 1)$ se verifica $g(1) - g(x) = g'(\xi)(1 - x)$. Como $g'(\xi) < 1$ y $g(1) = 1$, nos queda $1 - g(x) < 1 - x$ y por lo tanto $g(x) - x > 0$, descartando la existencia de una raíz en $[0, 1)$, con lo cual concluye la demostración.

Notemos que con $m = 1$ se puede tener probabilidad de extinción nula, si $p_1 = 1$, o extinción casi segura, si $p_1 \neq 1$. Un proceso con $m = 1$ se denomina *crítico*, mientras que el caso $m < 1$ se llama *subcrítico* y el $m > 1$, *supercrítico*.

1.3 Fórmula iterativa general para las medias

Vimos en la sección anterior que la esperanza del número de individuos en la n-ésima generación en un proceso de un tipo es $E(S_n) = m^n$, donde $m = E(S_1)$. Es decir que podemos escribir

$$E(S_{n+1}) = E(S_n)E(S_1).$$

Es un resultado conocido que la posibilidad de calcular medias de manera iterativa es general en procesos de ramificación. En todos los procesos de tiempo discreto, las esperanzas $E_xZ_n(A)$ verifican la fórmula iterativa

$$E_xZ_{n+1}(A) = \int_X E_yZ_n(A)dE_xZ_1(y),$$

para $n = 1, 2, \ldots$, que no requiere que los tipos sean acotados. El conjunto A es cualquier subconjunto medible de X y $dE_xZ_1(y)$ significa que la integración se lleva a cabo con respecto a la medida definida por $E_xZ_1(S)$ para cualquier subconjunto medible S de X. Esta fórmula se prueba en Harris (1963), Capítulo 3, mediante funcionales generadores de momentos y en Mode (1971), Capítulo 6, condicionando respecto de la composición de la población en la primera generación producida por el individuo de tipo x. Será utilizada cuando desarrollemos ejemplos de procesos con tipos no acotados en conjuntos numerables y no numerables.

Cuando los tipos son discretos tenemos numerables medias de la forma $m_{ij} = E$(número de hijos de tipo j que tiene un individuo de tipo i). Éstas definen una *matriz de medias* que es finita o infinita según la cantidad de tipos. Bajo interpretación discreta, el miembro de la derecha de la fórmula iterativa, $\int_X E_yZ_n(A)dE_xZ_1(y)$, es el resultado de elevar a la potencia $n + 1$ la matriz de medias y tomar la suma de los elementos de la fila de x que se encuentran en las columas que

corresponden a los elementos de A. Es decir, es consistente con la conocida fórmula para procesos con finitos tipos que establece que la esperanza del número de descendientes de tipo j que tiene un individuo de tipo i en la n-ésima generación es el elemento (i,j) de la matriz de medias elevada a la n.

En el caso continuo, supongamos que tenemos un núcleo $k(x,y)$ tal que para cualquier conjunto medible $A \subset X$ se cumple que $E_xZ_1(A) = \int_A k(x,y)d\mu(y)$ para alguna medida μ en X, es decir el análogo continuo de la matriz de medias. Veremos que la fórmula iterativa general deviene en la iteración del operador integral $K : L^2(X,\mu) \to L^2(X,\mu)$ definido por $Kf(x) = \int_A k(x,y)f(y)d\mu(y)$ para $f \in L^2(X,\mu)$ dándonos

$$E_xZ_{n+1}(A) = \int_A K^n(k(x,y))d\mu(y).$$

Probaremos esta igualdad. Para $n = 1$ la fórmula iterativa general es

$$E_xZ_2(A) = \int_X E_yZ_1(A)dE_xZ_1(y),$$

donde reemplazamos $E_yZ_1(A)$ por su expresión basada en el núcleo k para obtener

$$E_xZ_2(A) = \int_X \int_A k(y,w)d\mu(w)dE_xZ_1(y)$$

$$= \int_A \int_X k(y,w)dE_xZ_1(y)d\mu(w).$$

Ahora usamos el hecho de que para cualquier función medible f se tiene $\int_X f(y)dE_xZ_1(y) = \int_X k(x,y)f(y)d\mu(y)$, dado que por $E_xZ_1(A) = \int_A k(x,y)d\mu(y)$ vale para indicadores y entonces también para funciones simples y cualquier función medible. Nos queda

$$E_xZ_2(A) = \int_A \int_X k(x,y)k(y,w)d\mu(y)d\mu(w)$$

$$= \int_A K(k(x,w))d\mu(w).$$

Utilizando este resultado y repitiendo el procedimiento

$$E_xZ_3(A) = \int_X E_yZ_2(A)dE_xZ_1(y)$$

$$= \int_X \int_A K(k(y,w))d\mu(w)dE_xZ_1(y)$$

$$= \int_A K^2(k(x,w))d\mu(w),$$

y en general $E_xZ_{n+1}(A)$ resulta $\int_A K^n(k(x,w))d\mu(w)$, como queríamos mostrar.

Capítulo 2

Condiciones para extinción casi segura

En este capítulo exponemos dos condiciones suficientes para que en un proceso de ramificación de tiempo discreto la extinción tenga lugar con probabilidad 1, si la población tiene inicialmente un número finito de individuos de tipos cualesquiera. Ambas se pueden aplicar a procesos con tipos no acotados. La primera involucra la esperanza del número total de descendientes de un individuo y se la ha utilizado para probar extinción casi segura de procesos de ramificación que dominan clusters de percolación (Hall 1985, Meester y Roy 1996, Fernández, Ferrari y García 2001, Maric, 2002). La Sección 2.1 trata el criterio que resulta de esta condición. La segunda condición se basa en una acotación condicional de los tipos. Pide que si un individuo tiene pocos hijos, la capacidad de éstos de dejar descendencia esté acotada, en un sentido que luego se hará preciso. Esta segunda condición se puede verificar en casos críticos en los que no se cumple la primera. La presentamos en la Sección 2.2. Los resultados de este capítulo y los dos siguientes constituyen el artículo de Tetzlaff (2004).

2.1 Criterio de la esperanza del número total de descendientes

El resultado que se aplica en este caso es el que indica que si en una población que comienza con un individuo de tipo x, la suma de las esperanzas del número de individuos a lo largo de todas las generaciones es finito entonces se tiene que la extinción de la población es casi segura, es decir

$$\sum_{n=1}^{\infty} E_x Z_n(X) < \infty \Rightarrow q_x = 1.$$

Esto se justifica considerando que $\sum_{n=1}^{n} E_x Z_n(X)$ es una integral doble con integrando no negativo. Tenemos entonces $E_x \sum_{n=1}^{\infty} Z_n(X) < \infty$. Pero la variable aleatoria $\sum_{n=1}^{\infty} Z_n(X)$ es el tamaño total de la descendencia del individuo de tipo x y si ésta tiene esperanza finita no puede ser $q_x < 1$.

Ejemplos de la limitación de este criterio son casos críticos en los que $q_x = 1$ y sin embargo $E_x \sum_{n=1}^{\infty} Z_n(X) = \infty$, lo cual demuestra que no siempre la condición es necesaria para extinción casi segura. El ejemplo más elemental es el de un proceso de un solo tipo en el que la media de hijos de un individuo es $E_x Z_1(X) = m = 1$ pero la probabilidad de no tener hijo alguno es positiva. Aquí la teoría de procesos de ramificación indica que $q_x = 1$ pero como $E_x Z_n(X) = (E_x Z_1(X))^n = m^n = 1$, se tiene $\sum_{n=1}^{\infty} E_x Z_n(X) = \infty$.

2.2 Criterio basado en acotación condicional de tipos

Sea $Z_0, Z_1, \ldots$ un proceso de ramificación de tiempo discreto con tipos en un espacio X, numerable o no numerable. Daremos a continuación una condición sobre los elementos de X que

expresa que si un individuo tiene pocos descendientes, éstos no son demasiado prolíficos (acotación condicional). Bajo esta hipótesis, que se hará luego más precisa, el teorema siguiente prueba que si la sucesión de esperanzas del número de individuos en cada generación no tiende a infinito, aún si oscila sin una cota superior, la extinción es casi segura. Esto ocurre porque, como se verá en la demostración, si dicha esperanza se ubica infinitas veces debajo de una cota, entonces el número de individuos también deberá ser menor que una cota para infinitas generaciones con una probabilidad arbitrariamente cercana a 1. Esto significa que la población pasa infinitas veces por "cuellos de botella". Aquí es donde la condición de acotación condicional actúa porque asegura una cota inferior positiva para la probabilidad de que el número de individuos caiga a 0 cuando la población pasa por uno de estos cuellos de botella.

Condición 2.2.1 Para cualquier número de hijos $k > 0$, existe un número de generaciones $m(k)$ y un número real $q(k) > 0$ tal que

$$\inf_{x \in X} P_x(Z_{m(k)}(X) = 0 | 1 \leq Z_1(X) \leq k) \geq q(k).$$

Teorema 2.2.1 Si se cumple la Condición 2.2.1 y $\liminf_n E_x Z_n(X) < \infty$, entonces $q_x = 1$.

Demostración. Por el lema de Fatou, $\liminf_n E_x Z_n(X) < \infty$ implica $E_x \liminf_n Z_n(X) < \infty$, o sea $\sum_{k \geq 0} P_x(\liminf_n Z_n(X) > k) < \infty$. Luego, dado $\epsilon > 0$, existe un entero positivo k_ϵ tal que $P_x(\liminf_n Z_n(X) > k_\epsilon) \leq \epsilon$.

Es decir $P_x(Z_n(X) > k_\epsilon$ para todos salvo finitos valores de $n) \leq \epsilon$ y por lo tanto

$$P_x(Z_n(X) \leq k_\epsilon, i.v.) > 1 - \epsilon.$$

Veremos que para cualquier entero positivo k, $P_x(1 \leq Z_n(X) \leq k, i.v.) = 0$. Esto completará la demostración porque entonces para todo ϵ, $P_x(Z_n(X) = 0, i.v.) > 1 - \epsilon$ y la probabilidad de extinción deberá ser 1. Sea

$$N_0(k) = 0$$

y para $i = 1, 2, \ldots,$

$$N_i(k) = \min\{n > N_{i-1}(k) - 1 + m(k) : 1 \leq Z_n(X) \leq k\}$$

con $\min \emptyset = \infty$. Ahora,

$$P_x(1 \leq Z_n(X) \leq k, i.o.) = P_x(\cap_i \{N_i(k) < \infty\}) = \lim_{i \to \infty} P_x(N_i(k) < \infty),$$

así que es suficiente probar que $P_x(N_i(k) < \infty) \leq (1 - q(k)^k)^i$.

Para $i = 0$ tenemos

$$P_x(N_0(k) < \infty) = 1 = (1 - q(k)^k)^0$$

y para $i > 0$

$$P_x(N_{i+1}(k) < \infty) \leq P_x(N_i(k) < \infty, Z_{N_i(k)-1+m(k)}(X) > 0)$$

$$= \sum_{n=1}^{\infty} P_x(Z_{n-1+m(k)}(X) > 0 | N_i(k) = n) P_x(N_i(k) = n).$$

Pero dado $N_i(k) = n$, hay como máximo k individuos en la generación n. Estos deben ser los hijos de a lo sumo k individuos presentes en la generación $n-1$, que no pueden haber tenido más que k hijos cada uno. Por lo tanto por la Condición 2.2.1, $1 - q(k)^k$ es una cota superior para $P_x(Z_{n-1+m(k)}(X) > 0 | N_i(k) = n)$, dado que $q(k)^k$ es menor o igual que la probabilidad de que k o menos individuos de cualquier tipo no dejen descendencia alguna en $m(k)$ generaciones. Tenemos así

$$P_x(N_{i+1}(k) < \infty) \leq (1 - q(k)^k) \sum_{n=1}^{\infty} P_x(N_i(k) = n)$$

$$= (1 - q(k)^k) P_x(N_i(k) < \infty).$$

El resultado buscado queda entonces establecido por inducción.

Capítulo 3

Condiciones para probabilidad positiva de no extinción

En este capítulo obtenemos condiciones suficientes para asegurar que la probabilidad de extinción de la población no es 1, es decir, es positiva la probabilidad de que haya descendencia por infinitas generaciones. Nuestros resultados no dependen de que tengamos tipos acotados pero sí de una relación de orden parcial en el conjunto de tipos, la cual podremos verificar en nuestras aplicaciones en el Capítulo 4.

3.1 Procesos con un solo tipo

En la Sección 1.2 hemos repasado el conocido resultado de procesos de ramificación con un único tipo según el cual tenemos en particular que la extinción no tiene probabilidad 1 si la media m del número de hijos es mayor que 1. Es decir que tenemos para un proceso $S_0, S_1, \ldots$ con un único tipo

$$E(S_1) > 1 \Rightarrow q < 1$$

donde $E(S_1) = m$ es la esperanza del número de hijos que tiene un individuo y q es la probabilidad de que la población originada en un individuo se extinga alguna vez. Argumentos de acoplamiento nos permitirán usar este resultado en la sección siguiente para obtener resultados análogos para procesos con múltiples tipos, aún si son no acotados.

3.2 Criterios para infinitos tipos no uniformemente acotados

Definiendo un orden en el conjunto de tipos según su capacidad de producir descendencia, se obtendrán aquí condiciones que implican que la probabilidad de extinción es menor que 1. Cabe destacar que no se consigue aquí el valor de esta probabilidad, salvo el hecho de ser distinta de 1. Esto es suficiente en algunas aplicaciones, en particular aquella de la que se hace uso en la segunda parte de este trabajo. Allí se definirá un algoritmo de simulación de dos etapas. En la primera se genera un cluster de objetos del cual se sabe que es menos numeroso que una población de cierto proceso de ramificación. En la segunda etapa se utiliza este cluster como input, por lo cual el algoritmo es factible sólo si el cluster es finito. Saber que la población del proceso de ramificación es finita implica entonces factibilidad del algoritmo. La limitación de este método está en que podrían existir situaciones en que el cluster de objetos sea finito con probabilidad 1 y la población de ramificación que lo acota superiormente no lo sea. Pero a falta de mayor conocimiento sobre la manera de acotar el tamaño del cluster, mientras haya probabilidad positiva de no extinción en el proceso de ramificación no estaremos en condiciones de afirmar que el algoritmo va a ser factible.

Los resultados que se dan a continuación no dependen de que los tipos sean tales que $\sup_{x \in \mathcal{X}} E_x Z_1(\mathcal{X}) < \infty$ ni que exista un número de generaciones m tal que $\inf_{x \in \mathcal{X}} P_x(Z_m(\mathcal{X}) = 0) > 0$, es decir que se adaptan a nuestro contexto de tipos no acotados, a diferencia de los que pueden encontrarse en la literatura sobre procesos de ramificación.

Sea $Z_0, Z_1, \ldots$ un proceso de ramificación de tiempo discreto con tipos en un conjunto X numerable o no numerable. Dado un tipo x y un entero positivo n, llamaremos *descendientes de la n-ésima generación de x* a los individuos de la n-ésima generación de una población que comenzó con un único individuo de tipo x. Definamos la siguiente relación de orden en el conjunto de tipos.

Definición 3.2.1 Dados dos tipos x e y pertenecientes a X diremos que x es mayor que y si para cualquier conjunto medible $A \subset X$ y cualquier n ($n = 1, 2, \ldots$), el número de descendientes de n-ésima generación con tipos en A es estocásticamente mayor para x que para y, es decir, $P_x(Z_n(A) > k) \geq P_y(Z_n(A) > k)$ para cualquier entero k.

En base a este orden parcial definiremos un proceso de ramificación de un tipo que domina inferiormente al proceso original, de modo que si el de un tipo es supercrítico, el original también tiene una probabilidad positiva de no encontrar la extinción. Recordemos que $E_y Z_i(A)$ denota la esperanza del numero de descendientes de la i-ésima generación de un individuo de tipo y, mientras que q_x es la probabilidad de que una población que comienza con un individuo de tipo x se extinga en alguna generación.

Teorema 3.2.1 Sea y un tipo en X. Supongamos que existe un conjunto medible A cuyos elementos son mayores que y, y un entero positivo i tal que $E_y Z_i(A) > 1$. Entonces para cualquier x mayor que y se tiene $q_x < 1$.

Demostración. Sea $S_0, S_1, \ldots$ un proceso de ramificación de un tipo definido por

$$S_0 = 1$$

y para $k = 0, 1, 2, \ldots,$

$$p_k = P(S_1 = k) = P_y(Z_i(A) = k).$$

Como la media de este proceso es $E_y Z_i(A)$, la cual es mayor que 1 por hipótesis, tenemos que el proceso es supercrítico, es decir que hay una probabilidad positiva de no extinción, según vimos en el Capítulo 1. Ahora consideremos $Z_0, Z_i, Z_{2i}, \ldots$, el proceso de ramificación multitipo en las generaciones que son múltiplos de i, empezando con un único individuo cuyo tipo es x. Como x y todos los elementos de A son mayores que y, $Z_{ni}(A)$ es estocásticamente mayor que S_n para todo n. Esta afirmación se puede probar por acoplamiento, construyendo conjuntamente el proceso unitipo y el multitipo de manera tal que en toda realización conjunta ω se cumpla que $S_n(\omega) \leq Z_{ni}(A)(\omega)$. Para esto, usando el hecho de que x es mayor que y en el sentido definido arriba, asignamos al individuo inicial de tipo x del proceso multitipo los descendientes de i-ésima generación con tipos en A de un individuo y. Marcamos estos individuos y completamos la descendencia de i-ésima generación del individuo de tipo x para que ésta siga la distribución que corresponde al tipo x. El número de individuos marcados corresponderá a nuestra construcción del proceso de un tipo en la primera generación. Ahora, como todos los individuos marcados son mayores que y, procedemos con cada uno de ellos como lo hicimos con el individuo inicial de tipo x, marcando una parte de sus descendientes de i-ésima generación de la misma manera. La construcción del proceso multitipo se completa asignando la descendencia que corresponde a los individuos no marcados. Así sucesivamente se continua la construcción de las generaciones que son múltiplos de i. El número de individuos marcados sigue entonces la distribución del proceso

de un tipo y es siempre menor o igual que el número de individuos con tipo en A. Como además $Z_{ni}(A) \geq Z_{ni}(X)$, resulta $Z_0(X), Z_i(X), Z_{2i}(X), \ldots$ estocásticamente mayor que un proceso supercritico. Luego $q_x < 1$.

Corolario 3.2.1 Si para un tipo x existe i tal que $E_x Z_i(\{x\}) > 1$ entonces $q_x < 1$.

Demostración. Se toma $y = x$ y $A = \{x\}$ en el teorema.

Corolario 3.2.2 Supongamos que existen y, i y A como en el teorema. Si se tiene un tipo x, un entero j y un conjunto $B \subset X$ cuyos elementos son mayores que y, tales que $E_x Z_j(B) > 0$, entonces $q_x < 1$.

Demostración. Por el teorema sabemos que $q_b < 1$ para todo $b \in B$ porque un tal b es mayor que y. Además, como $E_x Z_j(B) > 0$, se cumple que $P_x(Z_j(B) > 0) > 0$. Luego,

$$P_x(no_extinción) \geq P_x(no_extinción | Z_j(B) > 0) P_x(Z_j(B) > 0) > 0.$$

Corolario 3.2.3 Si existe un tipo s menor que todos, en el sentido que todo tipo $x \in X$ es mayor que s, y además se tiene que $E_s Z_i(X) > 1$ para algún i, entonces $q_x < 1$ para todo x.

Demostración. Se toma $y = s$ y $A = X$ en el teorema.

Capítulo 4

Extensión a tipos no acotados de aplicaciones discretas y continuas

Este capítulo proporciona ejemplos de procesos de ramificación con tipos no acotados. Para eso se extienden aplicaciones conocidas de la teoría de procesos de ramificación con múltiples tipos, al caso de infinitos tipos no acotados. En primer lugar se trata de procesos que modelizan la dinámica de poblaciones biológicas (modelo de Leslie) en las que el conjunto de tipos es discreto. En la segunda aplicación tenemos un conjunto de tipos continuo que surge en el contexto de percolación continua no orientada. De aquí se desprenderá en la Segunda Parte la aplicación a percolación continua orientada, para obtener condiciones suficientes de aplicabilidad del algoritmo de simulación Backward-Forward de Fernández, Ferrari y García en el caso de objetos no uniformemente acotados.

La dinámica de Leslie nos dará también la oportunidad de plantear un problema de criticalidad en procesos de ramificación que quedará aquí abierto. En efecto, explorando los límites de aplicabilidad de los resultados de los capítulos precedentes, hemos llegado a definir un proceso que no corresponde a una aplicación práctica de la dinámica de Leslie pero tiene interés teórico porque no se puede resolver el problema de determinar si la extinción es o no casi segura mediante el Teorema 2.2.1 ni el 3.2.1, ni mediante otro resultado de la literatura de procesos de ramificación.

En todos los ejemplos que desarrollaremos tendremos tanto $\sup_{x\in X} E_x Z_1(X) = \infty$ como $\inf_{x\in X} P_x(Z_m(X) = 0) = 0$ para cualquier m, es decir, las dos condiciones que caracterizan a los procesos que hemos definido como de tipos no acotados. La verificación de la segunda de estas dos condiciones no será inmediata. Para comprobar que se cumple presentamos una proposición que facilitará esa tarea. Esta afirma que $\inf_{x\in X} P_x(Z_m(X) = 0) = 0$ para todo m se cumple si se verifican dos condiciones que en nuestras aplicaciones serán fáciles de comprobar. La primera pide que para cualquier cantidad de individuos h, la probabilidad $P_x(Z_1(X) \leq h)$ se pueda hacer arbitrariamente pequeña al recorrer los tipos $x \in X$. La segunda pide que exista una cota inferior positiva α, uniforme sobre los tipos $x \in X$, para la probabilidad de que un individuo de tipo x deje algún descendiente.

Proposición 4.1 Supongamos que para todo entero positivo h, $\inf_{x\in X} P_x(Z_1(X) \leq h) = 0$ y que existe $\alpha > 0$ tal que para todo $x \in X$, $P_x(Z_1(X) > 0) > \alpha$. Entonces $\inf_{x\in X} P_x(Z_m(X) = 0) = 0$ para cualquier número de generaciones m.

Demostración. Dado que para todo h, $\inf_{x\in X} P_x(Z_1(X) \leq h) = 0$, existe una sucesión $\{x_h : h = 1,2,...\} \subset X$ tal que $P_{x_h}(Z_1(X) \leq h) \to 0$ cuando $h \to \infty$. Ahora, para todo m,

$$P_{x_h}(Z_m(X) = 0) =$$

$$= P_{x_h}(Z_m(X) = 0, Z_1(X) \leq h) + P_{x_h}(Z_m(X) = 0, Z_1(X) > h)$$

$$\leq P_{x_h}(Z_1(X) \leq h) + P_{x_h}(Z_m(X) = 0 | Z_1(X) > h) P_{x_h}(Z_1(X) > h).$$

En este último miembro, el segundo término no es mayor que la probabilidad de que h individuos de primera la generación no tengan descendientes en la m-ésima generación, es decir, la probabilidad de que h individuos no tengan descendientes en la $(m-1)$-ésima generación de las poblaciones que ellos originan. Esta probabilidad esta acotada superiormente por $(1-\alpha^{m-1})^h$ porque por hipótesis, un individuo de cualquier tipo que sea, deja descendencia en la generación $m-1$ con probabilidad mayor o igual a α^{m-1}. Luego

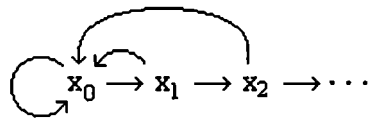
$$P_{x_h}(Z_m(X)=0) \leq P_{x_h}(Z_1(X) \leq h) + (1-\alpha^{m-1})^h.$$

Tomando límite inferior a ambos lados de esta desigualdad para h tendiendo a infinito, obtenemos $\lim\inf_h P_{x_h}(Z_m(X)=0) \leq 0$ y la proposición queda demostrada.

4.1 Dinámica poblacional de Leslie

Consideraremos el análogo con infinitos tipos del proceso de ramificación con finitos tipos estudiado por Pollard (1966) que a su vez es la versión estocástica de la dinámica poblacional determinista descripta por Leslie (1945). Por el lado determinístico, este autor dio origen al estudio de las llamadas matrices de Leslie, ampliamente desarrollado (un survey de Hansen, 1986, cita 208 artículos sobre el tema). El conjunto de tipos en los trabajos citados es $\{x_0,...,x_n\}$ y aquí lo extendemos a $\{x_0,x_1,...\}$. En la aplicación original del modelo a la dinámica de poblaciones biológicas, los tipos representan clases de edades. Así, los individuos de tipo x_0 son los más jóvenes. Un individuo de tipo x_0 puede o no tener hijos de tipo x_0 y puede producir un individuo de tipo x_1 si sobrevive lo suficiente como para pasar a la siguiente edad. En general un x_i $(i=0,1,...)$ sólo puede producir individuos x_0 y eventualmente un x_{i+1}, de manera que la matriz infinita de medias tiene sólo dos elementos que pueden ser positivos en cada fila, $E_{x_i}Z_1(\{x_0\})$ y $E_{x_i}Z_1(\{x_{i+1}\})$. Por simplicidad, llamaremos hijos de un x_i tanto a los x_0 como a los x_{i+1} que aquél genera, a pesar de que el x_{i+1} no es un nuevo individuo en la interpretación biológica. La figura esquematiza las relaciones entre tipos para la dinámica que estamos describiendo.

Figura 4.1.1



Esquema de las relaciones de descendencia en el modelo de Leslie con infinitos tipos.

En principio nos interesa solamente exhibir un ejemplo de tipos discretos no uniformemente acotados para generalizar el problema matemático de la dinámica de Leslie, independientemente de la aplicación a la biología. Sin embargo, cabe notar que Edelstein-Keshet (1987) cita el caso de algas y hongos filamentosos segmentados que se propagan por alargamiento de los segmentos existentes y producción a partir de éstos de nuevos segmentos de longitud inicial mínima. Cuando no existiere una cota superior uniforme para la longitud de los segmentos, estos organismos, como población de segmentos, podrían ser modelados con tipos no acotados, en los que cada tipo correspondería a un intervalo de longitudes.

Veamos una familia de ejemplos de dinámica de Leslie con tipos no acotados, dentro de la cual varía un parámetro $p \in (0,1)$. Tendremos como medias en la primera generación

$E_{x_i}Z_1(\{x_0\}) = p^{1-i}$ y $E_{x_i}Z_1(\{x_{i+1}\}) = p^2$, de manera que la matriz infinita de medias es

$$\begin{pmatrix} p & p^2 & 0 & 0 & 0 & \\ 1 & 0 & p^2 & 0 & 0 & \\ p^{-1} & 0 & 0 & p^2 & 0 & \\ p^{-2} & 0 & 0 & 0 & p^2 & \\ \cdots & & & & & \cdots \end{pmatrix}.$$

En virtud de la relación recursiva entre las medias de las sucesivas generaciones, la primera fila de la n-ésima potencia de esta matriz consistirá en las medias de la n-ésima generación $E_{x_0}Z_n(\{x_0\})$, $E_{x_0}Z_n(\{x_1\})$,.... Veremos por inducción sobre n que estos elementos son

$$E_{x_0}Z_n(\{x_i\}) = \begin{cases} (1/2)(2p)^n(p/2)^i & si \quad 0 \le i < n \\ p^{2n} & si \quad i = n \\ 0 & si \quad i > n \end{cases}$$

Para esto hacemos el paso inductivo sobre n para $i = 0$

$$E_{x_0}Z_{n+1}(\{x_0\}) = \sum_{i=0}^{n-1}(1/2)(2p)^n(p/2)^i p^{1-i} + p^{2n}p^{1-n}$$

$$= (1/2)(2p)^n p\frac{1-(1/2)^n}{1-1/2} + p^{n+1} = (1/2)(2p)^{n+1},$$

para $i = 1,...,n$

$$E_{x_0}Z_{n+1}(\{x_i\}) = (1/2)(2p)^n(p/2)^{i-1}p^2 = (1/2)(2p)^{n+1}(p/2)^i,$$

y para $i = n+1$

$$E_{x_0}Z_{n+1}(\{x_{n+1}\}) = p^{2n}p^2 = p^{2(n+1)}.$$

En particular obtuvimos $E_{x_0}Z_n(\{x_0\}) = (1/2)(2p)^n$, de modo que por el Corolario 3.2.1, $q_{x_0} < 1$ si $p > 1/2$. Por un caso simple del Corolario 3.2.2, $q_{x_i} < 1$ para cualquier i.

En el caso $p \le 1/2$ la esperanza del número de individuos de cualquier tipo en la n-ésima generación ($n = 1,2,...$) en una población que comenzó con un solo individuo de tipo x_0 está acotada por una constante. En efecto

$$E_{x_0}Z_n(X) = \sum_{i=0}^{n} E_{x_0}Z_n(\{x_i\}) = (1/2)(2p)^n\frac{1-(p/2)^n}{1-p/2} + p^{2n} < 1.$$

De manera que se satisface la condición $\lim\inf_n E_{x_0}Z_n(X) < \infty$. Si $p < 1/2$ podemos decir más aún porque el número esperado total de descendientes de un x_0 en todas las generaciones está acotado en virtud de

$$\sum_{n=1}^{\infty} E_{x_0}Z_n(X) < \frac{1}{2-p}\sum_{n=1}^{\infty}(2p)^n + \sum_{n=1}^{\infty}p^{2n} < \infty.$$

Así $E_{x_0}\sum_{n=1}^{\infty} Z_n(X) < \infty$ y por lo tanto $q_{x_0} = 1$ como vimos en la Sección 2.1. Si $p = 1/2$ aparecen distintas posibilidades según la distribución subyacente y entonces el Teorema 2.2.1 puede ser útil como veremos a continuación.

Ejemplo de aplicación del Teorema 2.2.1

Consideraremos dos distribuciones de probabilidad distintas para la descendencia, ambas con las mismas medias de más arriba. En ambos casos el número de hijos de tipo x_0 de un padre x_i $(i = 0, 1, ...)$ seguirá una distribución de Poisson con media p^{1-i} mientras que el número de hijos de tipo x_{i+1} va a ser 1 con probabilidad p^2 y 0 con probabilidad $1 - p^2$. La diferencia residirá en la distribución conjunta para estas marginales, como veremos más abajo. Lo que ya puede verificarse con estas distribuciones marginales es que $\inf_{x\in X} P_x(Z_m(X) = 0) = 0$ para cualquier m. Veamos para eso que se cumplen las hipótesis de la Proposición 4.1. Dado un entero positivo h, para cualquier tipo x_i tenemos

$$P_{x_i}(Z_1(X) \leq h) \leq P_{x_i}(Z_1(\{x_0\}) \ \leq h).$$

Pero en virtud de la distribución marginal del número de hijos de tipo x_0, $P_{x_i}(Z_1(\{x_0\}) \ \leq h) \to 0$ cuando $i \to \infty$. Luego $\inf_{x\in X} P_{x_i}(Z_1(X) \leq h) = 0$. Además para todo i,

$$P_{x_i}(Z_1(X) > 0) \geq P_{x_i}(Z_1(\{x_{i+1}\}) \ = 1) = p^2,$$

lo cual completa las hipótesis.

Las distribuciones conjuntas que tomamos en primer lugar para los hijos de los x_i $(i = 0, 1, ...)$ son tales que la Condición 2.2.1 se satisface. Para cada x_i la distribución será aquella en la que el número de hijos de tipo x_0 es independendiente del número de hijos x_{i+1}. Esta familia de distribuciones satisface la Condición 2.2.1 para $m(k) = 2$ porque

$$P_{x_i}(Z_2(X) = 0 | 1 \leq Z_1(X) \leq k) >$$

$$> P_{x_i}(Z_2(X) = 0 | Z_1(\{x_{i+1}\}) \ = 0, 1 \leq Z_1(X) \leq k) \bullet$$

$$\bullet\, P_{x_i}(Z_1(\{x_{i+1}\}) \ = 0 | 1 \leq Z_1(X) \leq k),$$

donde

$$P_{x_i}(Z_2(X) = 0 | Z_1(\{x_{i+1}\}) \ = 0, 1 \leq Z_1(X) \leq k) \geq (P_{x_0}(Z_1(X) = 0))^k,$$

mientras que para cualquier entero h entre 1 y k,

$$P_{x_i}(Z_1(\{x_{i+1}\}) \ = 0 | Z_1(X) = h) =$$

$$= \frac{(1-p^2)e^{-p^{1-i}}(p^{1-i})^h/h!}{(1-p^2)e^{-p^{1-i}}(p^{1-i})^h/h! + p^2 e^{-p^{1-i}}(p^{1-i})^{h-1}/(h-1)!} = \frac{1}{1 + hp^{i+1}/(1-p^2)} > \frac{1}{1+k},$$

teniendo en cuenta que estamos considerando $p \leq 1/2$. Luego,

$$\inf_{x_i\in X} P_{x_i}(Z_2(X) = 0 | 1 \leq Z_1(X) \leq k) > (P_{x_0}(Z_1(X) = 0))^k \frac{1}{1+k}.$$

Dado que se probó que $\lim\inf_n E_{x_0} Z_n(X) < \infty$ para $p \leq 1/2$, tenemos $q_{x_0} = 1$ por el Teorema

2.2.1.

Contraejemplo y problema abierto

La segunda familia de distribuciones no satisface la Condición 2.2.1. Lo conseguiremos requiriendo que al menos para $k = 1$ y para infinitos valores de i,

$$P_{x_i}(Z_1(\{x_{i+1}\}) = 1 | 1 \leq Z_1(X) \leq k) = 1.$$

Esto es posible porque con nuestras distribuciones marginales se tiene $P_{x_i}(Z_1(X) \leq 1) \to 0$ cuando $i \to \infty$. Así que para cualquier i más grande que algún $\tilde{i}$, $P_{x_i}(Z_1(X) \leq 1) < p^2$. Dado que p^2 es la probabilidad de que un x_i tenga un hijo x_{i+1}, podemos definir para $i > \tilde{i}$ una distribución conjunta que carga $P_{x_i}(Z_1(X) \leq 1)$ sólo en la probabilidad de tener ningún x_0 y un x_{i+1}. Entonces para $i > \tilde{i}$, $P_{x_i}(Z_1(\{x_{i+1}\}) = 1 | Z_1(X) = 1) = 1$, con lo cual cumplimos con el requerimiento. Ahora estamos satisfaciendo una hipótesis análoga a la de la Proposición 4.1, cual es que para cualquier entero h, $\inf_{x_i \in X} P_{x_i}(Z_2(X) \leq h | Z_1(X) = 1) = 0$. Por esto y por el hecho de que existe $\alpha > 0$ tal que para todo x_i, $P_{x_i}(Z_1(X) > 0) > \alpha$, podemos probar en forma análoga a como procedimos en la Proposición 4.1, pero condicionando respecto del evento $\{Z_1(X) = 1\}$, que para cualquier número de generaciones m,

$$\inf_{x_i \in X} P_{x_i}(Z_m(X) = 0 | Z_1(X) = 1) = 0.$$

Esto significa que la Condición 2.2.1 no se satisface porque falla para $k = 1$. Como ya probamos que $q_{x_0} = 1$ para $p < 1/2$, una consecuencia de este ejemplo es que las hipótesis de la Teorema 2.2.1 no son necesarias para que la probabilidad de extinción sea 1. Para $p = 1/2$, el problema de criticalidad queda abierto.

Finalmente podemos decir del caso $p \leq 1/2$ para $i = 1, 2, \ldots$, que $q_{x_i} = 1$ si $q_{x_0} = 1$. En efecto, $q_{x_i} < 1$ implica $q_{x_{i-1}} < 1$ por ser positiva la probabilidad de que un individuo de tipo x_{i-1} tenga un hijo x_i.

Variante no recurrente

Presentamos aquí una variante de la dinámica poblacional de Leslie que es no recurrente, es decir, un proceso en el que no se cumple que dados dos tipos cualesquiera x e y, existe un número finito n de generaciones tal que $P_x(Z_n(\{y\}) > 0) > 0$.

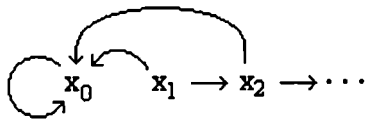
Sea $X = \{x_0, x_1, \ldots\}$ con $p \in (0, 1)$ y la matriz de medias

$$\begin{pmatrix} p & 0 & 0 & 0 & 0 & \\ 1 & 0 & p & 0 & 0 & \\ p^{-1} & 0 & 0 & p & 0 & \\ p^{-2} & 0 & 0 & 0 & p & \\ \cdots & & & & & \cdots \end{pmatrix},$$

o sea que las únicas esperanzas positivas son $E_{x_i} Z_1(\{x_0\}) = p^{1-i}$ para $i = 0, 1, \ldots$ y

$E_{x_i}Z_1(\{x_{i+1}\}) = p$ para $i = 1,2,\ldots$. La siguiente figura ilustra la dinámica entre tipos para este caso.

Figura 4.1.2



Esquema de las relaciones de descendencia entre tipos en el ejemplo no recurrente.

En una población que comienza con un único individuo de tipo x_1, sólo los tipos x_0 y x_{n+1} pueden estar presentes en la n-ésima generación. Las esperanzas de los números de individuos de esos tipos son, siempre usando la posibilidad de obtener las medias en una generación mediante el producto de medias de la generación anterior,

$$E_{x_1}Z_1(\{x_0\}) = 1,$$

$$E_{x_1}Z_1(\{x_2\}) = p,$$

$$E_{x_1}Z_2(\{x_0\}) = 1 \bullet p + p \bullet p^{-1} = 1 + p,$$

$$E_{x_1}Z_2(\{x_3\}) = p \bullet p = p^2,$$

$$E_{x_1}Z_3(\{x_0\}) = (1+p) \bullet p + p^2 \bullet p^{-2} = 1 + p + p^2,$$

$$E_{x_1}Z_3(\{x_4\}) = p^2 \bullet p = p^3,$$

y en general para $n = 1,2,\ldots$

$$E_{x_1}Z_n(\{x_0\}) = \sum_{i=0}^{n-1} p^i$$

y

$$E_{x_1}Z_n(\{x_{n+1}\}) = p^n.$$

Luego

$$E_{x_1}Z_n(X) = \sum_{i=0}^{n} p^i < \frac{1}{1-p}.$$

De manera que no podemos aplicar el criterio de la Sección 2.1 porque $\sum_{n=1}^{\infty} E_{x_1}Z_n(X) > \sum_{n=1}^{\infty} 1$. Se cumple sin embargo que $E_{x_1}Z_n(X)$ está acotada uniformemente sobre n, cumpliendo así con la hipótesis $\liminf_n E_{x_1}Z_n(X) < \infty$ del Teorema 2.2.1. Pero veamos en qué consiste esta dinámica no recurrente. Por un lado $q_{x_0} = 1$ porque la historia poblacional a partir de un individuo de tipo x_0 es la de un proceso con un único tipo y media $p < 1$. Además la línea de descendencia $x_1, x_2, \ldots$ se interrumpe con probabilidad 1 en virtud del criterio de la Sección 2.1, o visto de otro modo, en virtud de que en un esquema de Leslie cada x_i puede dar un solo x_{i+1}, en este caso con probabilidad p, y entonces el número de la generación en la cual esta

línea encuentra la extinción tiene distribución geométrica. Por otro lado, esta línea es una fuente de cantidades crecientes de individuos de tipo x_0 y desearíamos saber si esto es suficiente como para que haya una probabilidad positiva de que no se produzca extinción. Veamos que $q_{x_i} = 1$ para todo $i \geq 1$ cualquiera sea la distribución conjunta detrás de la matriz de medias. Sea $E_{i,k}$ el evento en el cual la línea de descendencia $x_i, x_{i+1}, \ldots$ llega hasta el tipo x_{i+k} y no más allá. Entonces

$$q_{x_i} = \sum_{k=0}^{\infty} P_{x_i}(extinción|E_{i,k}) P_{x_i}(E_{i,k})$$

Pero $P_{x_i}(extinción|E_{i,k}) = 1$ porque la extinción es casi segura si no llegamos a tener tipos x_{i+k+1} dado que sólo quedarán finitos individuos de tipo x_0, los cuales se extinguirán con probabilidad 1. Luego

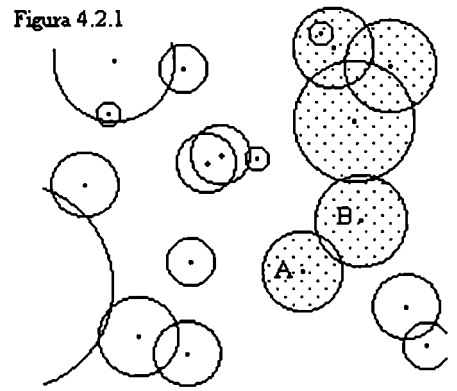
$$q_{x_i} = \sum_{k=0}^{\infty} P_{x_i}(E_{i,k}) = 1.$$

Así hemos resuelto en este caso el problema de criticalidad. En cuanto a la posibilidad de resolverlo mediante el Teorema 2.2.1, la situación aquí es que del mismo modo que en los ejemplos anteriores podemos definir una familia de distribuciones con las medias dadas que hace $q_{x_1} = 1$ como consecuencia del teorema. Por otro lado, también como antes se puede definir en este contexto no recurrente una familia de distribuciones tal que el problema de criticalidad no se resuelve con dicho teorema.

4.2 Procesos que dominan clusters de percolación continua

El proceso espacial de percolación continua consiste en puntos ubicados al azar en el espacio continuo $\mathbb{R}^d$ según un proceso de Poisson, en donde en cada punto se ubica un objeto, como por ejemplo una esfera que se centra en el punto. Los objetos pueden tener tamaño aleatorio pero éste es independiente del punto en el que se sitúan. Dado uno de estos objetos cualquiera, el *cluster* al que pertenece es el conjunto de objetos que lo intersecan, más todos lo que intersecan a éstos y así sucesivamente. Los clusters de percolación continua en $\mathbb{R}^d$ pueden dominarse mediante procesos de ramificación de manera que la extinción en el proceso de ramificación implica que el cluster de percolación es finito. Esto se consigue definiendo una relación padre-hijo entre objetos que se intersecan en el esquema de percolación. Eventualmente se agregan individuos adicionales a las familias así definidas, de modo que las familias resultantes cumplen con la independencia entre las descendencias de distintos individuos propia de los procesos de ramificación.

Figura 4.2.1



Se domina estocásticamente el tamaño del cluster de una esfera A con el de una familia en la que B es hija de A, B recibe una descendencia independiente con la distribución de las esferas de percolación que la intersecan, lo mismo para las hijas de B y así sucesivamente.

Este método para acotar el tamaño de los clusters aparece en Hall (1985). Daremos una justificación del mismo al final de esta sección utilizando acoplamiento de procesos estocásticos con una herramienta de representación de esferas que aquí desarrollaremos. El objetivo de la dominación es obtener una cota superior para λ, la intensidad del proceso de Poisson de la percolación, tal que bajo esa cota, la finitud de los clusters es casi segura.

El problema de criticalidad que analizaremos será el de procesos de ramificación que dominan clusters de percolación continua de objetos cuyo tamaño puede ser arbitrariamente grande. Consideraremos el caso de esferas en $\mathbb{R}^d$ $(d \geq 1)$ con radios no uniformemente acotados, los cuales serán los tipos en el proceso de ramificación. Dada una esfera en $\mathbb{R}^d$, su descendencia está determinada por esferas que tienen centros en un proceso de Poisson homogéneo en $\mathbb{R}^d$ con intensidad λ y radios independientes e idénticamente distribuidos. Los hijos son aquellas esferas que intersecan la esfera dada. A su vez la descendencia de cada uno de estos hijos está generada por un nuevo proceso de Poisson independiente de los demás y radios independientes, y así sucesivamente.

No acotación de los tipos

Vemos que $\sup_{x \in X} E_x Z_1(X) = \infty$ porque la esperanza del número de esferas que intersecan un esfera dada es mayor o igual que la del número de puntos del proceso de Poisson que se encuentran dentro de la esfera y que son centros de esferas hijas. Esta última esperanza no tiene una cota superior si el tamaño de la esfera madre puede ser arbitrariamente grande.

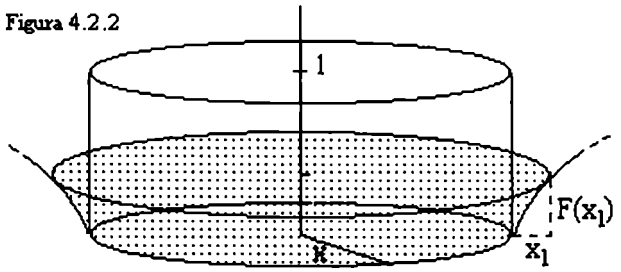
Para comprobar que $\inf_{x \in X} P_x(Z_m(X) = 0) = 0$ para cualquier m usamos la Proposición 4.1. Tenemos que para todo entero positivo h, $\inf_{x \in X} P_x(Z_1(X) \leq h) = 0$ porque tomando una esfera de radio suficientemente grande podemos ubicar por debajo de cualquier cota la probabilidad de que haya h o menos puntos dentro de la esfera. Además podemos ver que existe $\alpha > 0$ tal que para todo $x \in X$ se tiene $P_x(Z_1(X) > 0) > \alpha$, tomando para esta cota el valor de la probabilidad de que alguna esfera interseque el centro de la esfera madre. Esta probabilidad es positiva y no depende del radio x de la esfera madre.

Núcleo de medias y operador integral del proceso

Sea F la función de distribución acumulada de los radios, de la cual asumimos que tiene una inversa $F^{-1} : [0,1) \to [0,\infty)$ cuando se la restringe a $[0,\infty)$. Vamos a deducir la expresión de un núcleo de medias representando a las esferas hijas como puntos de un proceso de Poisson homogéneo con intensidad λ en $\mathbb{R}^d \times [0,1)$. Las primeras d dimensiones dan el centro de la esfera hija y la última es la imagen de su radio por F. Esto es posible porque la proyección a las primeras d dimensiones de un tal proceso $d+1$-dimensional es un proceso de Poisson de parámetro λ, mientras que la componente restante es independiente y uniforme en $[0,1)$, con lo cual tiene distribución F cuando le aplicamos F^{-1}.

Sea $u \in [0,1)$ un valor para la última componente de nuestra representación puntual. Las esferas de radio $F^{-1}(u)$ que intersecan una esfera materna de radio x pueden estar en uno de dos conjuntos. Uno es el de aquellas cuyos centros se ubican dentro de la esfera madre. El otro es el de las que tienen el centro fuera de ella pero a una distancia en $\mathbb{R}^d$ al centro de la madre que está dentro del intervalo $(x, x+F^{-1}(u)]$. Así que en nuestra representación en $d+1$ dimensiones, los puntos que corresponden a hijos de tipo $F^{-1}(u)$ de un individuo de tipo x se encuentran en la esfera d-dimensional que tiene las mismas componentes para el centro que la esfera madre, radio igual a $x+F^{-1}(u)$ y última componente igual a u. Para cualquier $x_1 \in (0,\infty)$, la unión de tales esferas sobre u en el intervalo $[F(0), F(x_1)]$ es un sólido como el que se esquematiza en la figura para $d = 2$.

Figura 4.2.2



El sólido sombreado contiene los puntos que representan esferas que intersecan una esfera de radio X y tienen radio en $[0, X_1]$.

Este sólido tiene volumen igual a

$$\int_{F(0)}^{F(x_1)} v_d(x+F^{-1}(u))^d du = \int_0^{x_1} v_d(x+y)^d dF(y),$$

donde v_d es el volumen de la esfera de radio 1 en $\mathbb{R}^d$. Por lo tanto hemos obtenido el volumen del subconjunto de $\mathbb{R}^{d+1}$ cuyos puntos representan aquellos hijos de un x que tienen última componente $u \in [F(0), F(x_1)]$ o equivalentemente, tipo en $[0, x_1]$. Como nuestro proceso de Poisson es homogéneo de parámetro λ, el número esperado de hijos de un x con tipos en $[0, x_1]$ es

$$E_x Z_1([0,x_1]) = \lambda \int_0^{x_1} v_d(x+y)^d dF(y).$$

Sea $k_d(x,y) = \lambda v_d(x+y)^d$. Si suponemos que la distribución de los radios tiene el $2d$-ésimo momento finito (el contenido de las esferas tiene varianza finita) entonces el operador integral $K_d : L^2([0,\infty), dF) \to L^2([0,\infty), dF)$ definido por k_d es compacto porque k_d es un núcleo

Hilbert-Schmidt. En efecto, $\int_0^\infty \int_0^\infty k_d^2(x,y)dF(x)dF(y) < \infty$. Dado que K_d es además estrictamente positivo, tiene un autovalor dominante $\rho > 0$. Entonces como consecuencia de la fórmula $E_xZ_n([0,\infty)) = \int_0^\infty K_d^{n-1}(k_d(x,y))dF(y)$, se tiene que para cualquier tipo inicial x la sucesión $\{E_xZ_n([0,\infty)) : n = 1,2,...\}$ está acotada si y sólo si el radio espectral ρ es menor o igual que 1.

Resultados de criticalidad

Se verá a continuación que para cualquier dimensión d y cualquier tipo x, tenemos $\rho \leq 1$ si y sólo si $q_x = 1$. Para completar las hipótesis del Teorema 2.2.1 cuando $\rho \leq 1$ sólo falta probar que se satisface la Condición 2.2.1. Fijando $\tilde{x} > 0$, mostraremos primero que existe una cota positiva para la probabilidad de tener todos los hijos con tipos debajo de $\tilde{x}$, o más precisamente, para cualquier $k \geq 1$,

$$\inf_{x\in[0,\infty)} P_x(Z_1([0,\infty)) = Z_1([0,\tilde{x}])|1 \leq Z_1([0,\infty)) \leq k) > b(k)$$

para algún $b(k) > 0$. Por la representación de los hijos como puntos de un proceso de Poisson homogéneo, vemos que dado un número de hijos, cada elección de tipo para un hijo es independiente de los demás y que la probabilidad de que resulte de un tipo menor que $\tilde{x}$ cuando el progenitor es de tipo x, es igual al cociente entre volúmenes

$$\frac{\int_0^{\tilde{x}} v_d(x+y)^d dF(y)}{\int_0^\infty v_d(x+y)^d dF(y)} = \frac{\int_0^{\tilde{x}} \sum_{i=0}^d \binom{d}{i} x^{d-i} y^i dF(y)}{\int_0^\infty \sum_{i=0}^d \binom{d}{i} x^{d-i} y^i dF(y)}$$

$$= \frac{x^d \int_0^{\tilde{x}} dF(y) + \sum_{i=1}^d \binom{d}{i} x^{d-i} \int_0^{\tilde{x}} y^i dF(y)}{x^d + \sum_{i=1}^d \binom{d}{i} x^{d-i} \int_0^\infty y^i dF(y)}.$$

Las integrales son finitas en virtud del supuesto que hemos hecho acerca de los momentos, así que esta probabilidad es una función continua y estrictamente positiva de x. Hacer x tender a ∞ da el valor estrictamente positivo $\int_0^{\tilde{x}} dF(y)$. Luego la función tiene una cota inferior estrictamente positiva $\beta_{\tilde{x}}$ y podemos tomar el $b(k)$ que buscábamos como $\beta_{\tilde{x}}^k$. Finalmente notamos que un individuo cuyo tipo está en $[0,\tilde{x}]$ no tiene hijos con una probabilidad mayor o igual a la de $\tilde{x}$ porque el volumen que contiene la representación de sus hijos es menor o igual que el que le corresponde a $\tilde{x}$. Luego la Condición 2.2.1 se satisface por

$$\inf_{x\in[0,\infty)} P_x(Z_2([0,\infty]) = 0|1 \leq Z_1([0,\infty]) \leq k)$$

$$\geq \inf_{x\in[0,\infty)} P_x(Z_2([0,\infty]) = 0, Z_1([0,\infty)) = Z_1([0,\tilde{x}])|1 \leq Z_1([0,\infty)) \leq k) \bullet$$

$$\bullet\, P_x(Z_1([0,\infty)) = Z_1([0,\tilde{x}])|1 \leq Z_1([0,\infty)) \leq k)$$

$$\geq (P_{\tilde{x}}(Z_1(X) = 0))^k \beta_{\tilde{x}}^k.$$

Si $\rho > 1$, tomando $x = 0$ como el tipo inicial, $E_0Z_i([0,\infty)) > 1$ para alguna generación i. Por el Corolario 3.2.3, tenemos $q_x < 1$ para cualquier $x \in [0,\infty)$ si probamos que 0 es el tipo menor tal como lo pide ese corolario. En efecto, $x \geq y$ implica que x es mayor que y en el sentido

del Capítulo 3. Esta afirmación puede justificarse por acoplamiento, construyendo un proceso que comienza con un individuo de tipo y, conjuntamente con uno que comienza con un x. Primero asignamos hijos al y ubicando puntos con intensidad λ en el sólido de $\mathbb{R}^{d+1}$ que corresponde a la descendencia de y en la primera generación, esto es, la unión sobre $u \in [0,1)$ de las esferas de $\mathbb{R}^d$ con radio $y + F^{-1}(u)$. Este conjunto está contenido en el sólido correspondiente a la descendencia de primera generación si el tipo del progenitor es x porque $y + F^{-1}(u) \leq x + F^{-1}(u)$. Así que podemos asignar a x los mismos hijos que los de y más eventualmente algunos adicionales con el objeto de seguir la distribución de hijos de un x. En las generaciones siguientes, x recibe la misma descendencia que la de y más la descendencia de esos hijos adicionales de la primera generación. Para todo n, esto hace la generación n-ésima en cualquier $A \subset X$, igual o mayor para x que para y.

Prueba de la dominación estocástica de percolación con ramificación

Hemos expuesto el concepto de la dominación de los clusters de percolación con familias de un proceso de ramificación. Aquí lo justificaremos utilizando la representación de las esferas d-dimensionales mediante puntos en $\mathbb{R}^d \times [0,1)$ introducida más arriba.

Queremos probar la dominación estocástica del cluster de percolación de una esfera cualquiera E_0 con el proceso de ramificación que hemos definido y en el cual el individuo inicial es la esfera E_0. Para ambos es λ la intensidad del proceso de Poisson y los radios siguen la misma distribución. Para obtener la dominación estocástica basta definir un acoplamiento de ambos procesos tal que en toda realización el número de elementos del cluster es menor o igual que el del conjunto formado por E_0 y su descendencia en el proceso de ramificación. Notemos que la posibilidad teórica de realizar esta construcción acoplada implica que si en el proceso de ramificación la extinción es casi segura, entonces el cluster de percolación al que pertenence cualquier esfera es finito casi seguramente.

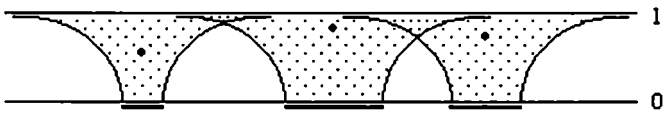
Dada una realización del proceso de percolación representada en $\mathbb{R}^d \times [0,1)$ y en la que E_0 está representada por el punto P_0, construímos la descendencia de E_0 en el proceso de ramificación de la siguiente manera. Las esferas hijas de E_0 son todas las que la intersecan en el proceso de percolación, es decir aquellas representadas por puntos que caen en el sólido V_0, siendo éste el subconjunto de $\mathbb{R}^d \times [0,1)$ donde según nuestra representación se ubican todos los puntos que corresponden a esferas que intersecan E_0. Sea $E_{1,1}$ una de las hijas de E_0 y $V_{1,1}$ la región que corresponde a las esferas que intersecan $E_{1,1}$. Los puntos del proceso de percolación que se encuentren en $V_{1,1}$ serán considerados como hijas de $E_{1,1}$ siempre que no se encuentren también en V_0. En $V_0 \cap V_{1,1}$ generamos un nuevo proceso de Poisson independiente homogéneo con intensidad λ para completar la descendencia de $E_{1,1}$ de manera que siga el proceso de ramificación. En forma análoga determinamos la descendencia de una segunda hija $E_{1,2}$ de E_0. Le asignamos como hijos todos los puntos de la región correspondiente $V_{1,2}$ salvo los que además caen en la región que le corresponde a cualquiera de los individuos para los cuales ya determinamos la descendencia. Lo que eventualmente le quitamos a $V_{1,2}$, es decir los puntos en $V_{1,2} \cap (V_0 \cup V_{1,1})$, se lo restituímos con un proceso de Poisson independiente en $V_{1,2} \cap (V_0 \cup V_{1,1})$. Del mismo modo seguimos con todas la hijas de E_1, y luego con las siguientes generaciones.

En general, cada inviduo recibe una descendencia independiente según la distribución del proceso de ramificación definido más arriba, obtenida con los puntos de la representación de la

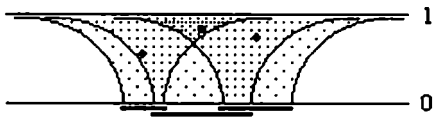
percolación en $\mathbb{R}^d \times [0,1)$. De éstos se sustraen los que ya están asignados como hijos de otro individuo y se suman los que eventualmente habrá que agregar para completar la distribución del propia del proceso de ramificación (del cual conocemos una condición sobre λ para tener extinción casi segura). Como esta construcción de la ramificación incorpora todos los elementos del cluster de percolación de E_0, tenemos el resultado de dominación buscado.

La figura ilustra para $d = 1$ la representación de esferas y las regiones cuyos puntos son la representación de esferas que las intersecan.

Figura 4.2.3



Caso d=1. Tres esferas (unidimensionales) que no se intersecan representadas como puntos en $R \times [0,1)$. Las superficies sombreadas alrededor de cada uno de los tres puntos contienen los puntos que representan las esferas que intersecan a cada una de las tres.



Representación de tres esferas unidimensionales formando un cluster. La familia de ramificación que lo acota se construye generando procesos de Poisson independientes en las intersecciones de las áreas sombreadas.

Segunda Parte:

Simulación exacta de redes con pérdida con objetos no acotados

Capítulo 5

Simulación de clusters de percolación continua

En el capítulo anterior apareció la percolación continua de esferas motivando la definición de procesos de ramificación asociados que permiten obtener conclusiones sobre aquella. En este capítulo describimos la simulación de clusters de percolación continua de esferas. Estas no serán necesariamente uniformemente acotadas. La técnica de construcción de clusters es la expuesta en Tetzlaff (2002).

Recordemos que la percolación continua de esferas estudia las propiedades de los clusters o aglomerados que aparecen al centrar esferas de radio, aleatorio o no, en puntos de un proceso de Poisson homogéneo de intensidad λ. Dada una esfera S se puede definir el cluster al que ella pertenece como el conjunto de esferas cuyos elementos son S y toda esfera que interseca a un elemento del conjunto. Los radios son independientes y siguen la distribución de una variable aleatoria R. El trabajo de Hall (1985) y el libro de Meester y Roy (1996) son las principales referencias sobre el tema. Esferas uniformemente acotadas son aquellas de radio constante o aleatorio R pero tal que existe una cota $c > 0$ tal que $P(R \leq c) = 1$. Las conclusiones que obtendremos para esferas pueden extenderse a otras formas geométricas pero nosotros nos limitaremos a las primeras.

Un objetivo de la teoría de percolación es conocer un valor λ' tal que la condición sobre la intensidad λ del proceso de Poisson

$$\lambda \leq \lambda'$$

es suficiente para asegurar que todo cluster es finito con probabilidad 1. Este es el objetivo que mencionamos en la Sección 4.2 y que en el contexto de simulación significa que un algoritmo que construye un cluster de percolación, concluirá la construcción en tiempo finito con probabilidad 1.

La percolación continua de esferas u otras formas tiene aplicaciones de lo más variadas. Modelos de esta naturaleza se vienen utilizando desde hace casi un siglo para representar por ejemplo cristalización en metales (Kolmogorov, 1937), aglomeración de partículas de polvo (Armitage, 1949), patrones en emulsiones fotográficas (Marchant y Dillon, 1961), microestructura del papel (Corte y Dodson, 1969), sistemas de gotas de agua en el estudio de transiciones de fase líquido-vapor (Widom y Rowlinson, 1970) y formación de estructuras geológicas por sedimentación (Jacod y Joathon, 1971).

Las esferas pueden representar no sólo objetos sino la zona de influencia o entorno de un objeto de menor tamaño o un punto. Así, el radio de la esfera puede ser el alcance de las llamas que provienen de un arbol en un incendio forestal, de manera que un cluster al que pertenece un árbol encendido equivale al conjunto de árboles hacia los cuales el incendio puede propagarse. En relación con esto, se puede citar el estudio de la distribución del calor en un bosque por Diggle (1981).

La aplicación que desarrollaremos nosotros aparecerá en el Capítulo 6, cuando adaptemos los

resultados de simulación de este capítulo al caso de la percolación orientada, con el objetivo final de simular redes con pérdida. En pos de ese objetivo, abordaremos la simulación de los clusters de percolación continua que intersecan un conjunto acotado A, es decir, nos interesa generar todas las esferas que intersecan un conjunto A más todas aquellas que intersecan a una esfera que interseca a A y así sucesivamente hasta obtener todos los aglomerados de esferas que tienen alguna intersección con A.

5.1 Percolación continua de esferas uniformemente acotadas

Simulación de un proceso de Poisson en un conjunto acotado

Para simular un proceso de Poisson homogéneo de intensidad λ en un conjunto $B \subset \mathbb{R}^d$ medible y acotado (en el sentido de estar contenido en un rectángulo acotado), se utilizan dos hechos bien conocidos:

(1) El número de puntos en un tal proceso, sigue la distribución de una variable aleatoria con distribución de Poisson de parámetro $\lambda \bullet v(B)$, donde $v(B)$ es el volumen del conjunto B (su medida de Lebesgue en $\mathbb{R}^d$).

(2) Cada uno de estos puntos se ubica en B con distribución uniforme sobre ese conjunto, independientemente de la ubicación de los demás.

Esto hace fácil la implementación de la simulación cuando B es un rectángulo $C \subset \mathbb{R}^d$, tanto por la posibilidad de cálculo del volumen como porque la distribución uniforme sobre el rectángulo d-dimensional C se consigue con d variables uniformes independientes sobre los lados de C. Para un conjunto B que no sea un rectángulo, se puede simular el proceso de Poisson en un rectángulo finito que lo contenga. De estos puntos, los que quedan dentro de B siguen el proceso de Poisson en B.

Cabe notar que para cualquier valor de λ, el número de puntos que cae en un conjunto de volumen finito es finito con probabilidad 1, por tener distribución de Poisson con parámetro igual a λ multiplicado por ese volumen. Esto garantiza la conclusión en finitos pasos de ciertos procedimientos que llevaremos a cabo en nuestras simulaciones y que utilizan como input una cantidad de puntos igual a la generada con un proceso de Poisson en un volumen finito.

Algoritmo de simulación de clusters de esferas uniformemente acotadas

Sea $A \subset \mathbb{R}^d$ un conjunto acotado. A continuación construiremos los clusters o aglomerados tales que alguna de sus esferas interseca el conjunto A. Llamaremos *cluster de A* a ese conjunto de clusters. Nuestro algoritmo consiste en generar el proceso de Poisson en capas esféricas y concéntricas tales que cada una contiene a la anterior. A cada punto del proceso se le asigna independientemente un radio con el fin de obtener una esfera. Cada nueva capa corresponde a un paso de nuestra construcción del cluster de A y tiene un ancho constante igual a c, la cota para los radios de las esferas. En cada paso se incorporan al cluster todas las esferas que intersecan lo

que ya tenemos construido del mismo. Llamamos la atención sobre el hecho de que tanto los objetos de la percolación como las capas en las cuales las vamos generando, son esferas. Las primeras tienen centros en los puntos del proceso de Poisson y las segundas, que llamaremos $S_0, S_1, \ldots$, son concéntricas y no son esferas del proceso de percolación sino que definen las capas en las que vamos generando el proceso. El algoritmo es el siguiente:

Algoritmo 5.1.1:

(1) Inicializamos un contador de pasos, $n = 0$.
Se define S_0 como la menor esfera que contiene al conjuto A.
Generamos en S_0 el proceso de Poisson de intensidad λ y a cada punto le asignamos independientemente un radio siguiendo la distribución de R, quedando así definidas todas las esferas con centro en S_0, pertenezcan o no al cluster de A.
Se marcan las esferas que intersecan a A. Si no existe esfera alguna que lo hace, el algoritmo concluye aquí.

(2) $n = n + 1$.
Se define S_n como la esfera que tiene el mismo centro que S_{n-1} y radio igual al radio de S_{n-1} más $2c$.
Generamos en $S_n - S_{n-1}$ el proceso de Poisson de intensidad λ y a cada punto se le asigna independientemente un radio siguiendo la distribución de R.

(3) Se recorren todas las esferas que están sin marcar y se marcan todas aquellas que intersecan a una esfera marcada. Esto se repite hasta que un recorrido de las esferas sin marcar no produzca una nueva marca.
Si este procedimiento no aumenta el número de esferas marcadas entonces el algoritmo concluye aquí. En caso contrario, se pasa a (2).

Proposición 5.1.1 Si el Algoritmo 5.1.1 concluye en un número finito de pasos, las esferas que quedan marcadas son las pertenecientes a los clusters que intersecan al conjunto A.

Demostración. Está claro que toda esfera marcada pertenece al cluster de A porque se marcan sólo esferas que intersecan al conjunto A o que intersecan esferas marcadas. Veamos la inclusión recíproca. Si el algoritmo concluye inicialmente (paso 0) es porque ninguna esfera interseca a A y tenemos trivialmente la inclusión deseada. Si concluye en el paso n ($n \geq 1$) es porque ninguna esfera con centro en $S_n - S_{n-1}$ interseca lo que teníamos construido del cluster de A (las esferas marcadas hasta ese momento). Entonces ya no hay esfera que falte incorporar a dicho cluster. En efecto, todas las esferas dentro de S_n ya han sido generadas, ninguna con centro en $S_n - S_{n-1}$ forma parte del cluster y dado que el radio de S_n es mayor en $2c$ que el de S_{n-1}, toda esfera con centro fuera de S_n no pertenece al cluster porque no podrá intersecar a esfera alguna con centro en S_{n-1}.

Proposición 5.1.2. Si todo cluster del proceso de percolación es finito con probabilidad 1, entonces el Algoritmo 5.1.1 concluye con probabilidad 1 en un número finito de pasos.

Demostración. Tenemos que con probabilidad 1 es finito el número de centros de esferas dentro de S_0. Por lo tanto, dada la hipótesis de la proposición, es también finto el número de esferas que componen los clusters que tienen intersección con A, los cuales hemos llamado cluster de A. Como el algoritmo incorpora por lo menos una nueva esfera al cluster de A en cada paso y cuando ya no puede hacerlo termina, tenemos que en un número finito de pasos va a concluir.

Corolario 5.1.1 Sea λ' la intensidad de un proceso de Poisson asociado a un proceso de percolación de esferas tal que para una distribución de radios dada, el proceso de ramificación que domina los clusters de percolación se extingue casi seguramente. Entonces si $\lambda \leq \lambda'$, el Algoritmo 5.1.1 construye el cluster de A en un número finito de pasos con probabilidad 1.

5.2 Percolación continua de esferas no uniformemente acotadas

En este caso no existe una cota c para los radios de las esferas. Una esfera que interseca una región dada puede tener su centro arbitrariamente distante de la misma y por lo tanto no se puede avanzar en la construcción del cluster dando pasos constantes (iguales a $2c$ en el algoritmo precedente). Daremos en esta sección un algoritmo, no tan sencillo como el anterior, pero que genera el conjunto de clusters que intersecan un conjunto A acotado cuando la distribución de los radios no es necesariamente uniformemente acotada.

Esferas del proceso de percolación que intersecan una esfera S dada

Notemos que el problema de construcción de las esferas que intersecan una esfera dada, que llamaremos S, es exactamente la situación que se planteó en la segunda sección del Capítulo 4. Aquí también podemos representar a las esferas hijas como puntos de un proceso de Poisson homogéneo con intensidad λ en $\mathbb{R}^d \times [0,1)$. Las primeras d dimensiones dan el centro de la esfera hija y la última es la imagen de su radio por F, la función de distribución del radio R. Por simplicidad hacemos el supuesto fácilmente generalizable de que F tiene una inversa $F^{-1} : [0,1) \to [0,\infty)$ cuando se la restringe a $[0,\infty)$, como es el caso de la distribución exponencial, por ejemplo.

Como en la Sección 4.2, si nuestra esfera S tiene radio x, obtenemos que los puntos que representan las esferas con cualquier radio positivo que la intersecan, están en el sólido de $\mathbb{R}^{d+1}$ cuyo volumen es

$$\int_0^1 v_d(x + F^{-1}(u))^d du = \int_0^\infty v_d(x+y)^d dF(y) = v_d \sum_{i=0}^{d} \binom{d}{i} x^{d-i} \int_0^\infty y^i dF(y),$$

donde v_d es el volumen de la esfera de radio 1 en $\mathbb{R}^d$. Nótese que el volumen es finito si $E(R^d) < \infty$. En la Sección 4.2 pedimos una condición más fuerte, a saber, $E(R^{2d}) < \infty$. La primera condición garantiza la finitud del número de esferas que intersecan una esfera dada. La segunda es parte de los requerimientos que aseguran, según se vió en la Sección 4.2, que el total de esferas del cluster que construyamos será finito.

A pesar del valor teórico de la representación $d+1$-dimensional, a la hora de implementar la

simulación, aparece el siguiente problema. Bajo el supuesto de que la distribución de los radios no es uniformemente acotada, el sólido de $\mathbb{R}^{d+1}$ donde quisiéramos ubicar puntos del proceso de Poisson que representan las esferas que intersecan la esfera de radio x no es un conjunto acotado, aunque su volumen sea finito. Por este motivo no podemos utilizar el procedimiento que consiste en generar el proceso en un rectángulo que lo contiene. Una manera de resolver esto es generar de manera distinta las esferas que tienen el centro dentro de la esfera S y las que no lo tienen pero intersecan a S, a saber:

(1) Las esferas con centro dentro de S se generan con un proceso de Poisson en S para los centros, y radios independientes que siguen la distribución F.

(2) Para las esferas que tienen centro fuera de S, se determina primero cuántas serán en nuestra simulación. Apelando a nuestra representación en $\mathbb{R}^{d+1}$, sabemos que el número N de ellas tiene distribución de Poisson con parámetro igual a λ multiplicado por el volumen en el que se ubican, es decir

$$\lambda \bullet (\int_0^1 v_d(x + F^{-1}(u))^d du - v_d x^d).$$

Ahora, la representación como puntos de un proceso de Poisson en un subconjunto de $\mathbb{R}^{d+1}$ nos muestra también que cualquiera sea el valor n que haya tomado N, el centro y radio de las n esferas se determinan independientemente para cada una de ellas y con las mismas distribuciones. Hacemos esto para cada esfera con la distribución de la distancia del borde de S al centro de una esfera que la interseca y completamos la extensión del radio con la distribución condicional que corresponda. Por ejemplo con $d = 1$, en el caso de radios con distribución exponencial de parámetro μ, la distancia entre el centro de una esfera (un segmento de $\mathbb{R}^1$) que interseca a S y el borde de S, es exponencial con el mismo parámetro, y la prolongación del radio también es exponencial con parámetro μ. Esto último es una versión del conocido resultado de la paradoja de inspección.

Esta manera de simular las esferas de percolación que intersecan una esfera dada nos permitirá definir un algoritmo de simulación de percolación de esferas no uniformemente acotadas.

Algoritmo de simulación de clusters de esferas no uniformemente acotadas

Generaremos el cluster de un conjunto acotado $A \subset \mathbb{R}^d$ con esferas de un proceso de percolación cuyos radios siguen la distribución de una variable aleatoria R que no es necesariamente uniformemente acotada. Sólo le pedimos a R que tenga momento de orden $2d$ finito, es decir $E(R^{2d}) < \infty$, requisito que apareció en la Sección 4.2, de la cual luego derivaremos una condición sobre λ que asegura finitud de los clusters con probabilidad 1. En este algoritmo también hay una sucesión de capas concéntricas pero a diferencia del algoritmo anterior son de tamaño aleatorio. Dada una capa se construyen no sólo las esferas con centro en ella sino también aquellas con centro fuera de ella pero que intersecan dicha capa. Cada nueva capa se define usando la menor esfera que contiene todo el cluster construido hasta el momento.

Algoritmo 5.2.1:

(1) Inicializamos un contador de pasos, $n = 0$.

Definimos S_0 como la menor esfera que contiene al conjuto A.

Siguiendo el método descripto precedentemente generamos un proceso de esferas de percolación que intersecan S_0. Es decir, primero simulamos en S_0 el proceso de Poisson de intensidad λ y a cada punto le asignamos independientemente un radio siguiendo la distribución de R, quedando así generadas todas las esferas con centro en S_0. A continuación generamos todas las esferas con centro fuera de S_0 pero que tienen intersección con S_0.

Se marcan las esferas que intersecan a A. Si no existe esfera alguna que lo hace, el algoritmo concluye aquí.

(2) $n = n + 1$.

Se define S_n como la menor esfera que tiene el mismo centro que S_{n-1} y contiene todas las esferas marcadas.

Siguiendo el método descripto precedentemente generamos un proceso de esferas de percolación que intersecan S_n. De estas esferas eliminamos aquellas que también tengan intersección con S_{n-1}, dado que en el paso anterior $(n - 1)$ ya quedó generado un conjunto de esferas que cumplen con ese requisito.

(3) Se recorren todas las esferas que están sin marcar y se marcan todas aquellas que intersecan a una esfera marcada. Esto se repite hasta que un recorrido de las esferas sin marcar no produzca una nueva marca.

Si este procedimiento no aumenta el número de esferas marcadas entonces el algoritmo concluye aquí. En caso contrario, se pasa a (2).

Proposición 5.2.1 Si el Algoritmo 5.2.1 concluye en un número finito de pasos, las esferas que quedan marcadas son las pertenecientes a los clusters que intersecan al conjunto A.

Demostración. Como en el caso del Algoritmo 5.1.1, toda esfera marcada pertenece al cluster de A porque se marcan sólo esferas que intersecan al conjunto A o que intersecan esferas marcadas. Veamos la inclusión recíproca. Si el algoritmo concluye en el paso 0 es porque ninguna esfera interseca a A y tenemos trivialmente el resultado. Si concluye en un paso posterior n es porque ninguna esfera construida en ese paso interseca las esferas marcadas hasta ese momento, entonces ya no hay esfera que falte incorporar al cluster de A. En efecto, todas las esferas marcadas están totalmente contenidas en S_n y todas las que intersecan S_n ya han sido generadas. Luego, si ninguna interseca una marcada, ninguna que no hayamos generado aún puede intersecar una esfera marcada y por lo tanto no hay más para incorporar al cluster de A.

La Proposición 5.1.2 se cumple para el Algoritmo 5.2.1 igual que para el Algoritmo 5.1.1 y la demostración es la misma. Lo mismo podemos decir para el Corolario 5.1.1. Damos a continuación los correspondientes enunciados.

Proposición 5.2.2. Si todo cluster del proceso de percolación es finito con probabilidad 1, entonces el Algoritmo 5.2.1 concluye con probabilidad 1 en un número finito de pasos.

Corolario 5.2.1 Sea λ' la intensidad de un proceso de Poisson asociado a un proceso de percolación de esferas no uniformemente acotadas tal que para una distribución de radios dada, el proceso de ramificación que domina los clusters de percolación se extingue casi seguramente. Entonces si $\lambda \leq \lambda'$, el Algoritmo 5.2.1 construye el cluster de A en un número finito de pasos con probabilidad 1.

5.3 Valor de λ que asegura finitud de los clusters

Se vió en la Sección 4.2 que el proceso de ramificación que domina los clusters de la percolación de esferas d-dimensionales, se extingue con probabilidad 1 si y sólo si $\rho \leq 1$, donde ρ es el máximo autovalor del operador $K_d : L^2([0,\infty), dF) \to L^2([0,\infty), dF)$ definido por $K_d f(x) = \int_A \lambda v_d (x+y)^d f(y) dF(y)$ para $f \in L^2([0,\infty), dF)$. Luego, $\rho \leq 1$ es una condición suficiente para que la percolación produzca clusters finitos casi seguramente.

Vamos a obtener un valor λ_c tal que si λ es la intensidad del proceso de Poisson en el que se centran las esferas de percolación, se tenga $\lambda \leq \lambda_c$ si y sólo si $\rho \leq 1$. Ahora bien, los autovalores de K_d no son más que $d+1$ porque el rango de K_d no supera este valor. En efecto, la imagen por K_d de cualquier $f \in L^2([0,\infty), dF)$ es

$$K_d f(x) = \int_A \lambda v_d (x+y)^d f(y) dF(y) = \lambda v_d \sum_{i=0}^{d} \binom{d}{i} x^{d-i} \int_0^\infty f(y) y^i dF(y),$$

es decir, un polinomio de grado no mayor que d. Luego la imagen de $L^2([0,\infty), dF)$ por K_d está contenida en el subespacio generado por los polinomios de grado menor o igual que d, el cual tiene dimensión $d+1$.

Además, planteando la ecuación

$$K_d v = \rho_v v$$

que deben verificar los autovalores y sus respectivos autovectores polinómicos, obtenemos un sistema de $d+1$ ecuaciones polinomiales (no lineales en general) que define implícitamente a los ρ_v y a los coeficientes de los autovectores en función de λ y de los momentos de la distribución F, desde el primero hasta el $2d$-ésimo, que llamamos $m_1, \ldots, m_{2d}$.

Por ejemplo con $d = 2$ consideramos los autovectores de la forma $x^2 + bx + c$. Entonces, como el volumen de la esfera de radio 1 es $v_2 = \pi$ tenemos

$$\lambda\pi \int_0^\infty (x^2 + 2xy + y^2)(y^2 + by + c) dF(y) =$$

$$= \lambda\pi(x^2 m_2 + bx^2 m_1 + cx^2 + 2xm_3 + 2bxm_2 + 2cxm_1 + m_4 + bm_3 + cm_2).$$

Igualamos a $\rho_v(x^2 + bx + c)$ para obtener una ecuación para cada potencia de x y nos queda el sistema no lineal

$$\lambda\pi(m_2 + bm_1 + c) = \rho_v$$

$$\lambda\pi 2(m_3 + bm_2 + cm) = b\rho_v$$

$$\lambda\pi(m_4 + bm_3 + cm_2) = c\rho_v$$

Para $d = 1$ se puede obtener analíticamente una solución del correspondiente sistema. Tenemos en este caso que la esfera de radio unitario es un segmento de longitud 2. Para obtener λ planteamos

$$\lambda 2 \int_0^\infty (x+y)(y+a)dF(y) = \rho_v(x+a),$$

o sea

$$\lambda 2(xm_1 + xa + m_2 + am_1) = \rho_v x + \rho_v a.$$

Igualando coeficientes queda el sistema

$$\lambda 2(m_1 + a) = \rho_v.$$

$$\lambda 2(m_2 + am_1) = \rho_v a.$$

Luego

$$\lambda 2(m_1 + a) = \lambda 2(m_2 + am_1)/a$$

es decir

$$a = \pm\sqrt{m_2}$$

Sustituyendo en $\lambda 2(m_1 + a) = \rho_v$ obtenemos λ como

$$\lambda = \frac{\rho_v}{2(m_1 \pm \sqrt{m_2})}.$$

De las dos soluciones para el par (λ, a) nos interesa la que corresponde al autovalor más grande, el radio espectral ρ. El valor de λ para esta solución es

$$\lambda = \frac{\rho}{2(m_1 + \sqrt{m_2})}.$$

Luego el valor crítico λ_c resulta de reemplazar ρ por 1 y la condición de extinción casi segura queda

$$\lambda \leq \frac{1}{2(m_1 + \sqrt{m_2})},$$

donde m_1 y m_2 son el primero y segundo momento del radio de la esfera unidimensional, es decir la mitad del segmento. Si reemplazamos los momentos por los de la longitud del segmento, entonces desaparece el 2 en el denominador de la expresión anterior como veremos a continuación.

Caso unidimensional, considerando diámetros en lugar de radios

En el caso unidimensional que acabamos de considerar, resulta más sencillo considerar los objetos como segmentos en lugar de esferas y caracterizar su extensión mediante su diámetro, es decir, su longitud. Vamos a obtener la expresión de la cota para λ en función de los primeros dos momentos del diámetro porque en los capítulos siguientes trabajaremos con segmentos.

Lamando μ_1 y μ_2 al primero y segundo momento del diámetro, tenemos las siguientes relaciones con los momentos del radio

$$\mu_1 = 2m_1,$$

$$\mu_2 = 4m_2.$$

Luego, $2(m_1 + \sqrt{m_2}) = \mu_1 + \sqrt{\mu_2}$ y la condición que asegura finitud de los clusters de percolación con probabilidad 1 es

$$\lambda \leq \frac{1}{\mu_1 + \sqrt{\mu_2}}.$$

Capítulo 6

Simulación de clusters de percolación continua orientada

Abordamos aquí la simulación de clusters de percolación continua orientada con respecto a una coordenada espacial. Este es un paso previo a la simulación de procesos de nacimiento y muerte con interacción usando el Algoritmo Backward-Forward para obtener muestras de redes con pérdida, lo cual es el tema del capítulo siguiente.

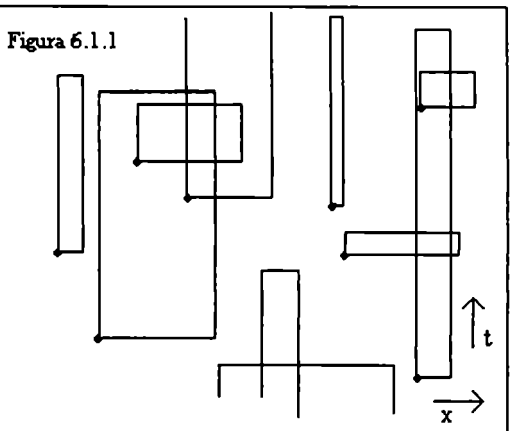
6.1 Percolación orientada de rectángulos en el plano

Veremos aquí la percolación continua orientada de objetos no uniformemente acotados. Estos serán rectángulos en $\mathbb{R}^2$ porque a partir de los clusters de percolación de éstos se podrá obtener la simulación de una red con pérdida en $\mathbb{R}$ en el capítulo próximo.

Considérese un proceso de Poisson homogéneo en $\mathbb{R}^2$ con intensidad λ en el que se centran esferas unidimensionales, es decir segmentos, paralelos al eje horizontal. Los radios son independientes, igualmente distribuídos y no necesariamente uniformemente acotados. Podemos generar un proceso con la misma distribución si en los puntos del proceso de Poisson situamos el comienzo de segmentos horizontales que se extienden hacia la derecha en el plano cartesiano y tienen longitud igual al doble de los radios mencionados. Esta manera equivalente de definir el proceso facilita la simulación.

El eje vertical representa el tiempo. A cada segmento se le asigna un tiempo de vida de manera que quedan definidos cilindros, más precisamente rectángulos, cuyas bases son las esferas (segmentos) y cuyas alturas representan sus respectivos tiempos de vida y se extienden en dirección vertical como lo ilustra la figura.

Figura 6.1.1



Los vértices inferiores izquierdos de los rectángulos siguen un proceso de Poisson. Las alturas tienen distribución exponencial.

Este tiempo de vida es independiente para cada esfera y sigue una distribución exponencial de parámetro 1. El valor de este parámetro es sólo una cuestión de escala, pero la condición de exponencial de la distribución de las alturas responde al hecho de que van a modelizar tiempos

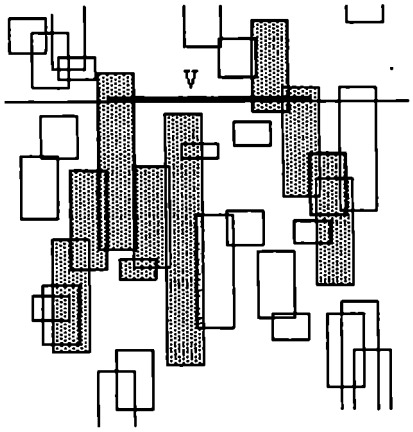
de vida en un proceso markoviano.

Nuestro objetivo es generar el conjunto de clusters de rectángulos que interseca un segmento finito V. Este segmento es llamado *ventana* y consiste en un segmento horizontal incluido en el eje $t = 0$, siendo t la variable correspondiente al eje cartesiano vertical. Los clusters se construyen con orientación, lo cual significa que un rectángulo (cilindro de base unidimensional) C_1 pertenece a un cluster que interseca la ventana si:

(1) C_1 interseca la ventana, o si

(2) C_1 interseca a algún rectángulo C_2 que pertenece al cluster y además C_1 nace antes que C_2.

Figura 6.1.2



Los rectángulos sombreados componen el cluster de la ventana V. En este ejemplo las bases de los rectángulos tienen longitud constante.

El conjunto de clusters que intersecan la ventana V se llamará simplemente *cluster de V*.

Algoritmo de simulación del cluster de V

La simulación sigue la misma idea del Algoritmo 5.2.1 del capítulo anterior, que consiste en definir un volumen o *capa* que contiene la parte ya obtenida del cluster y generar todos los objetos que la intersecan. Este procedimiento se repite hasta en uno de sus pasos ninguno de los nuevos objetos producidos sea alguno que se agregue al cluster. Como los objetos pueden ser arbitrariamente grandes, no podemos, como en el Algoritmo 5.1.1, generar solamente los objetos que tienen su origen (que en este caso es el vértice inferior izquierdo del rectángulo) en la capa que estamos considerando, sino que debemos generar todos los que intersecan la capa.

La primer capa es exactamente la ventana V. Los cilindros que intersecan esta ventana son los primeros elementos del cluster que nos interesa. El menor rectángulo que contiene estos cilindros, define la siguiente capa. A cada paso del algoritmo se construyen los cilindros que intersecan la capa presente lo cual eventualmente agranda el cluster de V. El menor menor rectángulo que contiene la parte del cluster construida hasta el momento define la nueva capa y así continúa la construcción hasta que en algún paso no se agregan nuevos cilindros al cluster de V. Tanto de los rectángulos de percolación como de las capas que se definen a lo largo del algoritmo, interesa solamente su intersección con el semiplano $t \leq 0$. Esto se debe a que nuestra construcción es orientada y procede hacia atrás en el tiempo. De hecho la construcción del

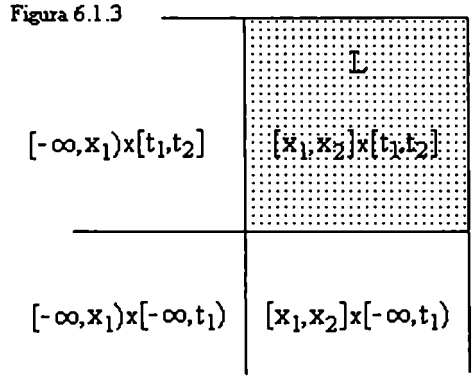
proceso más allá de $t = 0$ es irrelevante para la aplicación que haremos de él para simular la red con pérdida, como se verá más adelante.

Rectángulos de percolación orientada que intersecan una región rectangular L dada

El algoritmo de simulación requerirá tener resuelto el problema de generar todos los rectángulos de percolación orientada que interesecan una dada región rectangular con lados paralelos a los ejes.

Llamamos *origen* o punto donde *nace* un rectángulo de percolación, a su vértice inferior izquierdo. Dado un rectángulo L cuya base se extiende entre x_1 y x_2 ($x_1 < x_2$) y cuyo lado vertical va de t_1 a t_2 ($t_1 < t_2$), los rectángulos del proceso de percolación pueden tener su origen en una de cuatro regiones disjuntas. Estas son el interior de L, el sector a la izquierda de L (o sea $(-\infty,x_1) \times [t_1,t_2]$), el sector abajo de L ($[x_1,x_2] \times (-\infty,t_1)$) y el sector abajo a la izquierda de L ($(-\infty,x_1) \times (-\infty,t_1)$). La unión de éstos es $(-\infty,x_2) \times (-\infty,t_2]$ y cualquier rectángulo con origen en su complemento no puede intersecar a L. Estos sectores se representan en la siguiente figura.

Figura 6.1.3



Sectores en los que tienen su vértice inferior izquierdo los rectángulos que pueden intersecar un rectángulo L.

Utilizando el hecho de que punto de origen, base y altura son independientes, representamos a cada cilindro de percolación mediante un punto de un proceso de Poisson homogéneo de parámetro λ en $\mathbb{R} \times [0,1) \times \mathbb{R} \times [0,1)$. La primera componente es la coordenada en el eje horizontal (espacial) del origen del rectángulo. La segunda es la longitud de la base tras aplicarle su función de distribución que llamamos F, de la cual asumimos que tiene una inversa $F^{-1} : [0,1) \to [0,\infty)$ cuando se la restringe a $[0,\infty)$, lo cual puede generalizarse. La tercera es la coordenada vertical (temporal) del origen del rectángulo. La cuarta es la altura del rectángulo tras aplicarle su función de distribución que es $1 - e^{-x}$ por tratarse de una distribución exponencial de parámetro 1. Para cada uno de los sectores mencionados más arriba, veremos la manera de generar los rectángulos que tienen allí su punto de nacimiento y llegan a intersecar a L.

Rectángulos que nacen en el interior de L (en $[x_1,x_2] \times [t_1,t_2]$): Todo rectángulo que nace en el interior de L interseca a L cualquiera sea la longitud de su base y su altura. La región que contiene los puntos que representan estos rectángulos es entonces

$$[x_1,x_2] \times [0,1) \times [t_1,t_2] \times [0,1).$$

Luego, el número de estos rectángulos tiene distribución de Poisson de parámetro igual a λ multiplicado por el volumen de esta región, es decir $\lambda(x_2 - x_1)(t_2 - t_1)$. Para simularlos se produce un número pseudoaleatorio según esa distribución y se genera esa cantidad de veces un punto de nacimiento, una base y una altura, todos independientes.

Rectángulos que nacen a la izquierda de L (en $(-\infty,x_1) \times [t_1,t_2]$): Los rectángulos con origen en este sector intersecan L cualquiera sea su altura, siempre que su base b sea mayor que la distancia (horizontal) d entre el origen del rectángulo y L. Pero la condición $b > d$ es equivalente a $F(b) > F(d)$ porque supusimos que F es estrictamente creciente en $[0,\infty)$. Luego la región cuyos puntos representan los rectángulos que cumplen las presentes condiciones es

$$\{(x,x') : x \in (-\infty,x_1) \wedge x' > F(x_1 - x)\} \times [t_1,t_2] \times [0,1).$$

El área de $\{(x,x') : x \in (-\infty,x_1), x' > F(x_1 - x)\}$ es igual al de la región de $\mathbb{R}^2$ limitada por $x = 0, y = 1$ y la gráfica de F, cuya superficie es $\int_0^1 F^{-1}(u)du = \int_0^\infty v dF(v) = \mu$, el primer momento de la distribución F. Así, el volumen de la región que nos interesa es $\mu(t_2 - t_1)$ y el número de rectángulos allí representados tiene distribución de Poisson de parámetro $\lambda\mu(t_2 - t_1)$. Para simular estos rectángulos determinamos primero cuántos son, digamos k, siguiendo dicha distribución. Luego asignamos a cada uno independientemente una longitud para la base. Como la base corta el lado izquierdo de L, la generamos como la suma de dos segmentos. El primero de longitud igual a la distancia entre el origen de un rectángulo y L, condicional a que la base de este rectángulo es mayor que esta distancia, porque el rectángulo interseca L. Por ejemplo en el caso de bases exponenciales de media μ, esta distancia es exponencial de media μ. Con este primer segmento determinamos entonces la coordenada en el eje x para cada uno de los k rectángulos. El segundo segmento es la continuación de la base hacia la izquierda, que en el caso exponencial mencionado, también es exponencial de media μ e independiente de la porción anterior. Para finalizar la construcción de cada uno de los k rectángulos nos falta la ubicación de su origen en el eje vertical, que es uniforme en $[t_1,t_2]$, y la altura, que es exponencial de parámetro 1. Ambos valores se determinan con independencia de todos los anteriores.

Rectángulos que nacen abajo de L (en $[x_1,x_2] \times (-\infty,t_1)$): Aquí la situación es análoga a la anterior pero en la dirección vertical en lugar de la horizontal. La región de nuestra representación en $\mathbb{R} \times [0,1) \times \mathbb{R} \times [0,1)$ cuyos puntos representan los rectángulos que nacen en este sector e intersecan L es ahora

$$[x_1,x_2] \times [0,1] \times \{(t,t') : t \in (-\infty,t_1) \wedge t' > 1 - e^{-(t_1-t)}\}.$$

El volumen de esta región es $(x_2 - x_1) \bullet 1$ y el número de rectángulos allí representados tiene distribución de Poisson de parámetro $\lambda(x_2 - x_1)$. Para simular estos rectángulos determinamos primero cuántos son, siguiendo dicha distribución y luego asignamos a cada uno independientemente una longitud para la altura. Como la base en el caso de los rectángulos que nacen a la izquierda de L, dicha altura se compone de dos porciones. La primera será la distancia del punto de nacimiento a la base de L y se generará según una distribución exponencial de parámetro 1. La segunda será el resto de la altura desde t_1 hacia arriba y será independiente de la primera, con la misma distribución. En cuanto a la posición del nacimiento del rectángulo en el eje x, la determinamos mediante la distribución uniforme en $[x_1,x_2]$ y la longitud de la base mediante la distribución F.

Rectángulos que nacen abajo a la izquierda de L (en $(-\infty,x_1) \times (-\infty,t_1)$): Los rectángulos con origen en este sector intersecan L si cumplen simultáneamente los dos requisitos que pedimos en los sectores anteriores. Su base debe ser mayor que la distancia horizontal entre su origen y x_1 mientras que su altura debe ser mayor que la distancia vertical entre el nacimiento y t_1. La región de nuestra representación en $\mathbb{R} \times [0,1) \times \mathbb{R} \times [0,1)$ cuyos puntos representan los rectángulos que nacen en este sector e intersecan L es entonces

$$\{(x,x') : x \in (-\infty,x_1) \wedge x' > F(x_1 - x)\} \times \{(t,t') : t \in (-\infty,t_1) \wedge t' > 1 - e^{-(t_1-t)}\}.$$

El volumen de esta región es $\mu \bullet 1$ y el número de rectángulos allí representados tendrá entonces distribución de Poisson de parámetro $\lambda\mu$. Para simular estos rectángulos determinamos primero cuántos son, siguiendo dicha distribución y luego asignamos a cada uno independientemente una base y una altura como lo hicimos, respectivamente, para los rectángulos que nacen en $(-\infty,x_1) \times [t_1,t_2]$ y en $[x_1,x_2] \times (-\infty,t_1]$. Así queda también automáticamente determinado, para cada rectángulo, su vértice inferior izquierdo, es decir su punto de nacimiento dentro de $(-\infty,x_1) \times (-\infty,t_1)$.

Ahora que tenemos la posibilidad de simular los rectángulos de percolación no uniformemente acotados que intersecan un rectángulo dado, damos el algoritmo que genera el cluster de una ventana V. Notemos que si el rectángulo L tiene altura nula, como es el caso de la ventana V, el interior y el sector a la izquierda de L tienen área nula y sólo tenemos que generar cilindros bajo L y abajo a la izquierda de L.

Algoritmo 6.1.1:

(1) Inicializamos un contador de pasos, $n = 0$.
Se define L_0 como la ventana V.
Generamos y marcamos los rectángulos que intersecan L_0. Si no existe rectángulo alguno que lo hace, el algoritmo concluye aquí.

(2) $n = n + 1$.
Se define L_n como el menor rectángulo que contiene todos los rectángulos marcados.
Generamos todos los rectángulos de percolación que intersecan L_n. De éstos eliminamos aquellos que tengan intersección con L_{n-1}, dado que en el paso anterior $(n-1)$ ya quedó generado un conjunto de rectángulos que cumplen con ese requisito.

(3) Se recorren todos los rectángulos que están sin marcar y se marcan todos aquellos que intersecan algún rectángulo marcado y nacen antes que éste. Esto se repite hasta que un recorrido de los rectángulos sin marcar no produzca una nueva marca.
Si este procedimiento no hace que se marque cilindro alguno, el algoritmo concluye aquí. En caso contrario, se pasa a (2).

La Proposición 5.2.1 y la Proposición 5.2.2 de percolación de esferas no uniformemente acotadas del Capítulo 5 tienen sus versiones análogas para percolación orientada de rectángulos y se demuestran con iguales argumentos.

Proposición 6.1.1 Si el Algoritmo 6.1.1 concluye en un número finito de pasos, los

rectángulos que quedan marcados son los pertenecientes a los clusters que intersecan la ventana V.

Proposición 6.1.2 Si todo cluster del percolación es finito con probabilidad 1, entonces el Algoritmo 6.1.1 concluye con probabilidad 1 en un número finito de pasos.

También tenemos el correspondiente corolario:

Corolario 6.1.1 Sea λ' la intensidad de un proceso de Poisson asociado a un proceso de percolación orientada tal que para una distribución de radios dada, el proceso de ramificación que domina los clusters de percolación se extingue casi seguramente. Entonces si $\lambda \leq \lambda'$, el Algoritmo 6.1.1 construye el cluster de V en un número finito de pasos con probabilidad 1.

6.2 Valor de λ que asegura finitud de los clusters

Para la percolación no orientada de esferas conocemos la manera de acotar los clusters de percolación mediante un proceso de ramificación, según se describió en la Sección 4.2. Al final de la Sección 5.3 hemos desarrollado el caso unidimensional, llegando a una condición suficiente sobre la intensidad λ de puntos de origen de las esferas (segmentos) para que el cluster de percolación sea finito con probabilidad 1. Pedimos $\lambda \leq 1/(\mu_1 + \sqrt{\mu_2})$ donde μ_1 y μ_2 son el primero y segundo momento del diámetro de las esferas, es decir, la longitud de los segmentos.

Ahora queremos una condición análoga para el caso en que los segmentos son la base de rectángulos en $\mathbb{R}^2$ y λ no es la intensidad de un proceso de Poisson en $\mathbb{R}$ sino la de un proceso en $\mathbb{R}^2$, el de los vértices inferiores izquierdos de los rectángulos de percolación orientada. Veremos a continuación que en este caso obtenemos la misma condición.

Los segmentos de percolación no orientada se representaban según la Sección 4.2 como puntos en $\mathbb{R}^1 \times [0,1)$. Con esta representación obtuvimos la superficie cuyos puntos correspondían a los descendientes de un segmento dado que están en cierto rango de longitudes. El área de esta superficie nos proporcionó la expresión del núcleo del proceso de ramificación que domina la percolación. Ahora dominaremos la percolación orientada de rectángulos con un proceso de ramificación en el que los descendientes de un rectángulo dado son los rectángulos que nacen antes que él y tienen una base y una altura suficientes como para intersecarlo. Notemos que estamos considerando como descendientes de un rectángulo a ciertos rectángulos que nacen antes que él. Esto se debe a la orientación que le dimos a la coordenada temporal, la cual es arbitraria salvo que queremos que sea consistente con los conceptos de *backward* y *forward* que veremos en el algoritmo implementado en el capítulo siguiente.

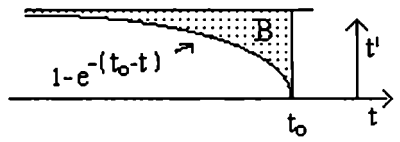
En la representación que hicimos de rectángulos como puntos en $\mathbb{R} \times [0,1) \times \mathbb{R} \times [0,1)$, las dos primeras componentes corresponden a la ubicación del origen del rectángulo en el eje x y a la longitud de su base. Las otras dos corresponden a la ubicación del origen en el eje t y a la altura. Sea A una región de $\mathbb{R} \times [0,1)$ que representa una condición sobre las primeras dos componentes tal que sus puntos representan hijos de una esfera madre (unidimensional) E_0 que tienen longitud dentro de cierto rango. En el caso de rectángulos de percolación, los hijos de un

rectángulo C_0 cuya base es E_0 tienen tiene que cumplir además la condición, independiente de la anterior, de nacer antes del tiempo t_0 de nacimiento de E_0 y tener un tiempo de vida mayor que t_0 menos el instante de su nacimiento. Esta condición se representa con

$$B = \{(t,t') : t \in (-\infty, t_0) \wedge t' > 1 - e^{-(t_0-t)}\}$$

y se grafica en la siguiente figura.

Figura 6.1.4



Como esta región tiene área igual a 1, el número de puntos de un proceso de Poisson en $\mathbb{R}^4$ de intensidad λ en $A \times B$ tiene la misma distribución que el número de puntos de un proceso de Poisson en $\mathbb{R}^2$ de intensidad λ en A. De manera que con los rectángulos llegamos al mismo núcleo del proceso de ramificación que con las esferas unidimensionales y por lo tanto también a la condición

$$\lambda \leq \frac{1}{\mu_1 + \sqrt{\mu_2}}.$$

Capítulo 7

Simulación de redes con pérdida mediante el algoritmo Backward-Forward

7.1 Redes con pérdida

Una red con pérdida es un modelo para un conjunto de recursos de capacidad limitada, relacionados según un ordenamiento espacial (como una red bidimensional de nodos, o un continuo unidimensional de puntos de un cable) que son requeridos en cantidad estocástica por demandas de duración también aleatoria. La teoría de redes con pérdida generaliza la de colas en el sentido que los despachadores dejan de ser necesariamente un conjunto homogéneo de elementos que no tienen relación entre sí. Kelly (1991) es la referencia fundamental sobre el tema. La distribución de probabilidad de los estados es una medida invariante de un proceso de nacimiento y muerte, lo cual es aprovechado para la simulación, como veremos en este capítulo.

El caso continuo unidimensional

El caso continuo unidimensional fue introducido por Kelly (1991) y estudiado por Ferrari y Garcia (1998). Es un modelo para la ocurrencia de llamadas en una red unidimensional, un cable, en la que cada punto correspondería a un despachador. Una llamada involucra a un segmento del cable cuyos extremos son el emisor y el receptor, o dicho de otro modo, ocupa a todo el continuo de despachadores correspondientes a ese segmento. Esta porción de cable está ocupada por la llamada durante un tiempo finito que es el tiempo que dura la llamada o *tiempo de vida*. En dos dimensiones, espacio y tiempo, tenemos un proceso de nacimiento y muerte con interacción que puede graficarse mediante rectángulos en $\mathbb{R}^2$. La interacción se debe a que no puede aparecer una llamada en una porción de cable con su capacidad ocupada. La capacidad del cable es un número entero $c \geq 1$. El proceso posee una única medida invariante y es ergódico si el proceso de percolación asociado produce clusters finitos con probabilidad 1 (Ferrari y Garcia, 1998). Maric (2002) estudió estas redes continuas unidimensionales para el caso en que las llamadas tienen una longitud aleatoria pero uniformemente acotada, obteniendo resultados sobre su simulación. Algunos de ellos se generalizan aquí gracias a nuestras conclusiones sobre procesos de ramificación con tipos no uniformemente acotados.

7.2 El algoritmo Backward-Forward

Simulación exacta

Cuando se simula un proceso estocástico y se desea tomar muestras de los estados más usuales del mismo, se plantea el problema de cuánto tiempo esperar desde el comienzo hasta cuando se toma una muestra. Entonces se deja correr el proceso un tiempo que se cree suficiente como para que la observación del sistema revele valores representativos del comportamiento del sistema. Esto es lo que se suele denominar "esperar que el sistema entre en régimen" o "warming up", pero rara vez se pueden dar criterios objetivos al respecto.

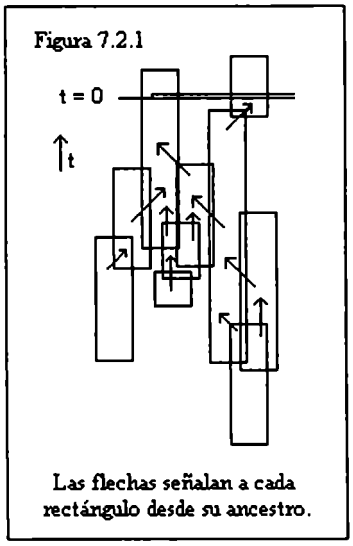
En el contexto de procesos markovianos lo que se se procura muestrear es la distribución invariante o de equilibrio. A partir del trabajo de Propp y Wilson (1996) se desarrollaron métodos que aseguran que la distribución de la que se está muestreando es exactamente la medida invariante. Las aplicaciones son muy variadas y el interés en el tema ha sido alto en los últimos años. El sitio http://dimacs.rutgers.edu/~dbwilson/exact está permanentemente actualizado sobre trabajos en este tema. La *simulación exacta* o *perfecta* apunta a la obtención de muestras de una distribución probabilística siguiendo exactamente esa ley de probabilidad y no sólo una aproximación a la misma. El algoritmo Backward-Forward de simulación de redes con pérdida de Fernández, Ferrari y García se inscribe dentro de esta categoría.

El algoritmo Backward-Forward

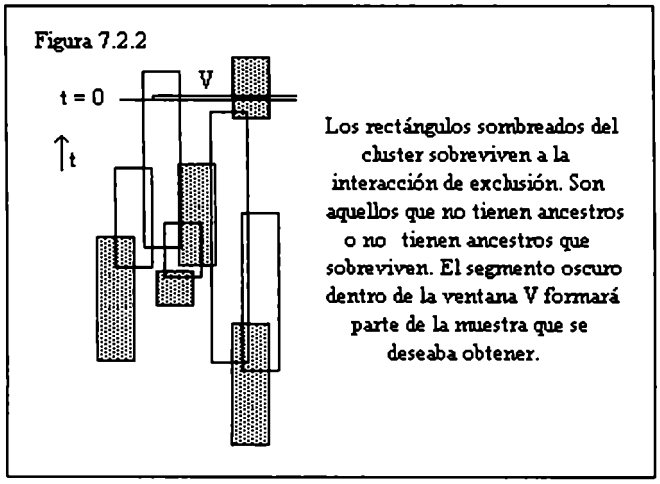
El algoritmo Backward-Forward es una técnica de simulación perfecta y se debe a Fernández, Ferrari y Garcia que lo desarrollan en tres trabajos (1998, 2001, 2002). Se aplica a cualquier medida absolutamente continua respecto de un proceso puntual de Poisson, que sea la medida invariante de un proceso de nacimiento y muerte. Permite simular procesos de interacción de objetos en el espacio euclideo $\mathbb{R}^d$ que se distribuyen al azar salvo que respetan ciertas relaciones de interacción como por ejemplo que dos objetos no pueden intersecarse. El algoritmo consiste en dos etapas: primero la llamada etapa *backward* (hacia atrás) y luego la etapa *forward* (hacia adelante).

Etapa Backward: Se generan hacia atrás en el tiempo los clusters de percolación orientada de cilindros que intersecan una ventana finita en la que se tomará la muestra al final del algoritmo. Estos cilindros tienen como base los objetos de interés y como altura un tiempo de vida exponencial de parámetro 1. Los cilindros generados constituyen un proceso libre, en el sentido de que se ignora todo tipo de interacción, de manera que se pueden intersecar libremente.

Dado un cilindro C, los cilindros que lo intersecan y nacen antes que él se denominan sus *ancestros*. Esta denominación es consistente con la orientación que le damos a la coordenada temporal porque vamos hacia atrás en el tiempo, agregando nuevos ancestros. Notemos que lo que aquí llamamos ancestro es un descendiente en el proceso de ramificación que domina la percolación. Esta aparente contradicción no tiene importancia alguna porque el proceso de ramificación se utiliza para obtener una condición suficiente de factibilidad del algoritmo pero no es ya tomado en cuenta durante la implementación del mismo. La siguiente figura ilustra la relación ancestro-descendiente según la denominación del algoritmo Backward-Forward.



Etapa Forward: Una vez finalizada la construcción de los clusters del proceso de percolación que intersecan la ventana, se eliminan ciertos cilindros desde atrás hacia adelante en el tiempo. Esta poda se realiza siguiendo la regla que corresponde a la interacción que queremos simular. Veamos por ejemplo la interacción de exclusión. Sobreviven todos los cilindros del cluster de percolación que no tienen ancestro alguno y pasan a la categoría de *guardados* (ya nunca serán borrados). Los cilindros intersecados por cilindros guardados son eliminados. Tras esta eliminación puede ser que otros cilindros no tengan más ancestros porque éstos fueron borrados. Al no tener cilindros que los intersequen y hayan nacido antes pasan a guardados y se borran los cilindros que intersecan. Así se procede hasta decidir, para todos los cilindros, si se guardan o se eliminan. El resultado tras la poda es un proceso de nacimiento y muerte con interacción, el proceso de los cilindros que fueron guardados. La intersección de los cilindros guardados con la ventana nos da la muestra de la red con pérdida, que es el objetivo final del algoritmo.



Otras reglas de interacción pueden ser la de la eliminar un cilindro sólo si tiene más de una cierta cantidad de ancestros, o la de decidir para cada cilindro si se lo elimina, independientemente y siguiendo cierta probabilidad pero favoreciendo a aquellos que se intersecan, por ejemplo. Con esta última se conseguiría una interacción atractiva en lugar de repulsiva.

Expliquemos aquí el motivo por el cual es exacta, en el sentido de seguir exactamente la distribución de equilibrio, la muestra obtenida con el algoritmo Backward-Forward desarrollado

por Fernández, Ferrari y Garcia en los trabajos citados, y que nosotros estamos estudiando para el caso de objetos no acotados. Ocurre que sólo los cilindros del cluster de la ventana tienen influencia en la muestra. Ir más allá del cluster en la construcción del proceso de percolación orientada no agrega cilindros que afecten lo que se observa en la ventana. De manera que nuestro resultado es el mismo que el que arroja un proceso de nacimiento y muerte construido desde un pasado arbitrariamente más lejano que el del comienzo del cluster de percolación. Así el algoritmo Backward-Forward nos proporciona la misma muestra que la que se obtendría de un proceso que corrió por tiempo infinito, es decir, exactamente con la distribución invariante o de equilibrio. Supongamos en cambio que se simula un proceso de nacimiento y muerte con interacción con los mismos parámetros pero hacia adelante en el tiempo. Se generan para ello cilindros con nacimientos aleatorios (en los puntos de un proceso de Poisson de parámetro λ) pero impidiendo (en el caso de la interacción de exclusión) el nacimiento de todo aquel que pretende nacer en un punto ocupado por un cilindro que nació anteriormente. Con un estado inicial arbitrario, la muestra que se tome en un instante finito será posiblemente sesgada respecto de la distribución de equilibrio, porque si el estado inicial no sigue esta distribución, la convergencia a la misma sólo se puede asegurar para el tiempo tendiendo a infinito.

Condición suficiente de factibilidad del algoritmo

Vemos que el algoritmo descripto se puede llevar a cabo siempre y cuando el número de cilindros producidos en la etapa backward sea finito. Estos cilindros son los elementos de un número finito de clusters de percolación orientada. Luego, una condición suficiente de factibilidad del algoritmo es que los clusters de percolación puedan ser dominados por un proceso de ramificación que encuentra la extinción con probabilidad 1.

Resumen del esquema de acotación y simulación para esferas interactuantes de $\mathbb{R}^d$

Antes de entrar en la implementación computacional, conviene repasar la relación entre los procesos aludidos hasta aquí. Se tiene *Ramificación* $\geq$ *Percolación orientada* $\geq$ *Nacimiento y muerte con interacción* donde los signos $\geq$ denotan la acotación de un proceso por otro, y donde destacamos que:

• El proceso de de ramificación no se simula, pero su estudio proporciona una cota para el valor de λ que asegura finitud de los clusters de percolación.
• La percolación orientada se simula para obtener un proceso de nacimiento y muerte que es libre en el sentido que los cilindros se intersecan sin interaccionar entre sí,
• El proceso de nacimiento y muerte con interacción resulta de borrar ciertos cilindros del proceso libre que hemos generado. Finalmente, dado que la distribución probabilística de una red con pérdida es la medida invariante de un proceso de nacimiento y muerte con interacción, la sección de este proceso espacio-temporal en tiempo 0 nos proporciona una muestra de la red con pérdida que es el objetivo final de la simulación.

Las dimensiones involucradas para cada proceso son las siguientes:

• En el proceso de ramificación, tenemos tipos en $\mathbb{R}^1$ (radios) que determinan objetos en $\mathbb{R}^d$ (esferas d-dimensionales). Se tiene $d = 1$ en el caso de segmentos como lo son las llamadas en el

cable del modelo de Kelly. Ahora bien, para deducir propiedades del proceso, hemos representado las esferas como puntos en $\mathbb{R}^{d+1}$ (Sección 4.2). En la percolación orientada, los objetos d-dimensionales son las bases de cilindros $d+1$-dimensionales pero, como hemos visto en detalle para $d = 1$ en la Sección 6.2, el proceso de ramificación que domina los clusters de estos cilindros tiene el mismo valor critico para el parámetro λ que el proceso que domina la percolación de esferas con la misma distribución de radios que las bases de los cilindros.

• En el proceso de percolación orientada, tenemos cilindros en $\mathbb{R}^{d+1}$, d dimensiones para la base y una para la altura que representa el tiempo. A su vez, representamos estos cilindros mediante puntos en $\mathbb{R}^{d+3}$ que poseen la siguiente información. (1) La ubicación en el espacio del centro de la esfera, o simplemente un extremo del segmento para esferas unidimensionales, para lo cual se necesitan d dimensiones. (2) La extensión del objeto en el espacio, que para esferas requiere sólo un valor real y que transformamos para que tenga distribución uniforme en $(0,1)$. (3) El instante de nacimiento del cilindro, un valor real. (4) El tiempo de vida del cilindro, exponencial de parámetro 1 y transformado para que tenga distribución uniforme en $(0,1)$

• Al seccionar en el instante de tiempo igual a 0 el proceso de cilindros interactuantes (proceso de nacimiento y muerte con interacción) en $\mathbb{R}^{d+1}$, obtenemos objetos de $\mathbb{R}^d$. Esta es una muestra de la red con pérdida que sigue la medida invariante del proceso de nacimiento y muerte.

7.3 Simulación de llamadas no uniformemente acotadas en un cable infinito

Este trabajo finaliza con la simulación de una red con pérdida unidimensional que modeliza llamadas no uniformemente acotadas en un cable infinito de capacidad $c = 1$. Destacamos que, aunque la ventana en la que obtenemos la muestra es finita, la simulación corresponde a un cable infinito dado que no imponemos condiciones de borde sino observamos una porción finita de un proceso que se extiende en todo $\mathbb{R}$. Esta posibilidad es propia del algoritmo Backward-Forward.

La etapa backward es exactamente la construcción del cluster de percolación orientada de rectángulos en $\mathbb{R}^2$ para una ventana finita, que ya hemos descripto. Para esta construcción conocemos una condición suficiente para asegurar que termina en tiempo finito y por lo tanto, que el algoritmo es factible. La etapa forward es la poda de este cluster, como se describió más arriba. Supondremos que la capacidad del cable es $c = 1$, lo cual es fácilmente generalizable a un valor mayor. La poda consiste en guardar los cilindros que no tienen ancestros y borrar aquellos cilindros que tienen al menos un ancestro que tiene categoría de guardado.

Nuestro programa en C, *BFNA* (por *Backward-Forward No Acotado*, en referencia a los tipos no uniformemente acotados) lleva a cabo la simulación. Cada ejecución del mismo, con una nueva semilla para el generador de números aleatorios, permite obtener una nueva muestra independiente de la red con pérdida en una ventana finita de un cable infinito. Toda la simulación del proceso de percolación hasta la obtención del cluster de la ventana y luego la poda que termina dando la muestra de la red con pérdida, se lleva a cabo mientras se generan salidas numéricas con los datos de los cilindros y salida gráficas. Estas últimas utilizan las subrutinas gráficas del ROM-BIOS de una PC.

Parámetros iniciales

La longitud de la ventana se fija en 5 unidades. Las llamadas son de longitud aleatoria con

distribución exponencial de media *mu* dada por el usuario. Como la distribución es exponencial (lo cual podría modificarse), el programa puede entonces calcular el valor λ_c que es crítico para el proceso de ramificación que domina la percolación orientada usando la fórmula final de la Sección 6.2. El valor crítico obtenido se expone al usuario para que en vista del mismo dé un valor de *lambda* que se usará como parámetro para el proceso de Poisson en $\mathbb{R}$ de los puntos de origen de llamadas. Este es también el parámetro del proceso de Poisson en $\mathbb{R}^2$ de los puntos que son origen de los rectángulos de percolación. Dado que la altura de los cilindros es independiente y la porción desde el nacimiento hasta un tiempo dado tiene media 1, el argumento de la Sección 6.2 indica que si la intensidad es λ en el proceso de los orígenes de los cilindros en $\mathbb{R}^2$, tendrá el mismo valor λ en el proceso de los orígenes de las llamadas en $\mathbb{R}$. Si el usuario da un valor de *lambda* menor o igual que el valor crítico de ramificación, sabe que la ejecución del programa va a concluir con probabilidad 1.

Se pide también una semilla incial para el generador de números pseudoaleatorios. Dos ejecuciones del programa con iguales valores de de *mu*, de *lambda* y de esta semilla dan resultados exactamente iguales.

Principales estructuras de datos: matrices C y L

En la matriz *C* se guardan los datos de los cilindros (en este caso rectángulos) que se van generando. Cada fila corresponde a uno de ellos. Las cinco columnas de la matriz almacenan los siguientes valores.

En la primer columna va un código para el estado del cilindro. El estado de todo cilindro generado por el proceso de percolación es en principio el de *libre*. Un cilindro libre pasa a ser *del cluster* cuando se determina que forma parte del cluster de la ventana. El estado de *borrado* corresponde a los cilindros que son eliminados ya sea por el algoritmo de construcción del cluster de percolación en la etapa Backward (Algoritmo 6.1.1) o por la poda de la etapa Forward. Finalmente, un cilindro pasa a estado de *guardado* (o *kept*) cuando sobrevive a la poda. En la salida gráfica del programa, los cilindros libres aparecen en color fucsia, los borrados en blanco y los del cluster en negro. Tras la poda del cluster, solamente quedan en negro los cilindros guardados.

En las siguientes cuatro columnas de la matriz se almacenan para cada rectángulo un valor en x para el origen de la llamada, uno para su destino (el otro extremo del segmento), un valor en t para el instante de comienzo de la llamada y uno para el final de la misma. Éstos determinan el rectángulo correspondiente a esa llamada y su ubicación. Se generan según el Algoritmo 6.1.1.

En *L* se guardan las *capas* o *layers* que utiliza el Algoritmo 6.1.1 en su construcción del cluster de percolación orientada, de las cuales la primera es la ventana. Cada fila corresponde a uno de estos rectángulos. En las columnas de *L* se almacenan para cada capa, en forma análoga a como se almacenan los cilindros en *C*, los valores en la coordenada x para su lado izquierdo y su lado derecho y los valores en la coordenada t de su lado inferior y de su lado superior.

Etapa Backward

Construye en primer lugar los cilindros de percolación que intersecan la ventana o capa 0 (fila 0 de la matriz L). Todos estos pertenecerán al cluster de la ventana. El menor rectángulo que los contiene (dentro del semiplano de tiempos no positivos) es la capa 1. El programa entra en el ciclo de construcción de capas descripto por el Algoritmo 6.1.1 y con probabilidad 1 termina el cluster de percolación si el valor de *lambda* dado por el ususario se encuentra bajo el valor crítico de ramificación.

Siempre se ignora la parte de los rectángulos que se encuentra en tiempo positivo del plano porque la misma es irrelevante para los fines del algoritmo Backward-Forward.

Etapa Forward

Se toma el cilindro del cluster que nació primero, es decir, que tiene su origen más abajo que todos los demás en el plano espacio-tiempo. Éste pasa a categoría de guardado. Se borran todos los cilindros que intersecan a éste.

Un ciclo repite la operación con los cilindros del cluster mientras quede alguno sin guardar ni borrar. Como el cluster tiene finitos cilindros y en cada iteración se guarda por lo menos uno, la etapa forward concluye en finitos pasos.

Generación de números pseudoaleatorios

La generación de números pseudoaleatorios se realiza con un generador lineal congruencial de período 65536, lo cual se puede implementar de manera eficiente en una sola línea de lenguaje C usando enteros *unsigned int*. Valores con distribución uniforme en un intervalo $[a, b]$ y con distribución exponencial se obtienen por el método de invertir la función de distribución acumulada. Cada valor con distribución de Poisson con un parámetro dado se obtiene restándole 1 al número de variables exponenciales independientes que hay que sumar para superar dicho parámetro.

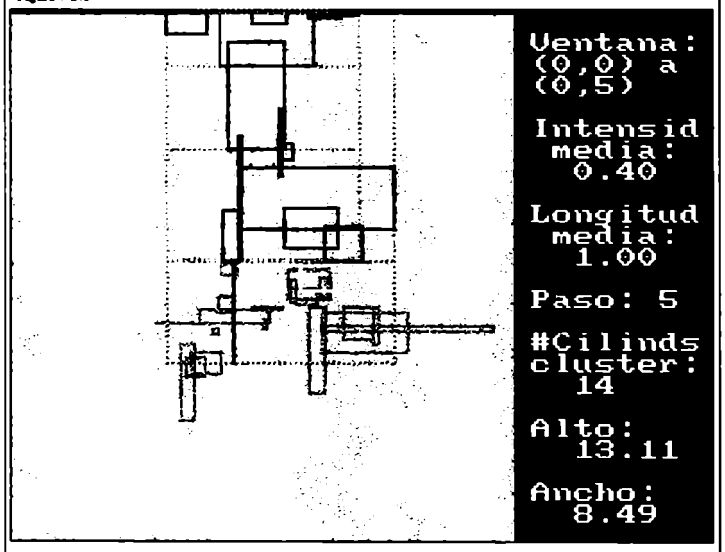
Costo computacional

La etapa backward acota el costo computacional de la simulación. Ésta hace uso del Algoritmo 6.1.1 que tiene un costo acotado por N^3 en su parte (3), donde llamamos N al número total de rectángulos producidos por el algoritmo. En efecto, en dicha parte (3) se recorre cada uno de los rectángulos sin marcar (acotados por N) y para cada uno se verifica cuáles de los demás (también acotados por N) lo intersecan para decidir si el rectángulo sin marcar se incorpora al cluster. Este procedimiento debe iterarse hasta que no quede sin marcar un rectángulo que interseca un rectángulo marcado. No puede haber más de estas iteraciones que el número de rectángulos que pueden llegar a marcarse y por lo tanto están acotadas también por N. Llegamos así al orden de N^3, que se repite para cada capa de la construcción del cluster por parte del Algoritmo 6.1.1. Como cada capa incorpora por lo menos uno de los N cilindros, no hay más capas que el total de cilindros y tenemos entonces un costo total polinómico acotado por N^4. Cabe destacar que N es aleatorio y que una caracterización completa del costo de la simulación requeriría estudiar las propiedades estocásticas de N.

Ejemplo de salidas gráficas del programa BFNA

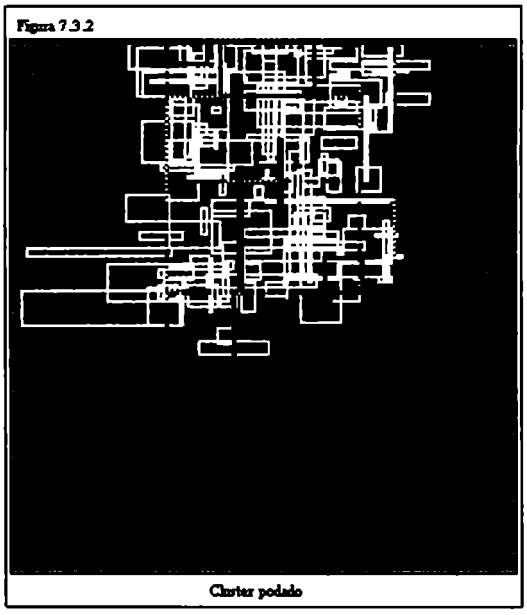
Damos aquí las dos salidas gráficas finales de la etapa backward y de la etapa forward de una ejecución del programa con una media $\mu = 1$ para la longitud de llamadas y $\lambda = 0,4$ para la intensidad del proceso de Poisson ($0,41$ es el valor crítico para $\mu = 1$). El segmento horizontal color fucsia intermitente en el lado superior es la ventana o capa 0. Con cinco capas más queda concluído el cluster cuyos rectángulos constituyentes aparecen en negro. Los límites de las sucesivas capas aparecen en fucsia intermitente. Los rectángulos en blanco (borrados) son los que el programa fue descartando por alguno de los dos motivos de borrado de la etapa backward. Uno es que fueron generados porque intersecan la capa i pero deben ser eliminados porque además intersecan la $i-1$. El otro es porque se determinó que no van a pertenecer al cluster. Obsérvese en la siguiente figura que ninguno de los rectángulos (en color fucsia) construídos en el paso correspondiente a la capa 5 cumple con intersecar algún rectángulo del cluster y nacer antes que él. Por eso el cluster esta concluído y por lo tanto también la etapa backward. No hace falta borrar esos nuevos rectángulos de la capa 5 porque el cluster completo ya está identificado.

Figura 7.3.1



Cluster terminado

Este cluster es sometido a la poda de la etapa forward dando el siguiente resultado.



La intersección con la ventana de los rectángulos que sobrevivieron es la muestra de la red con pérdida correspondiente a esta realización.

Listado del programa BFNA

A continuación damos el código fuente en C del programa BFNA.

```
#include "dos.h"
#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#include "math.h"


/* Definiciones globales */

#define CC 500
#define LL 25
#define ROWS 200
#define COLS 250
// CC cota num de cilindros, LL cota num de layers (capas)
// ROWS y COLS cotas para el Output grafico
// Transformaciones de escala:
// Input = Display numerico de Parametros
// Input (*100) -> Calculos y Valores en matrices C y L
//    (/10+100 en x; /10*-1 en t) -> Valores en matriz O y Display grafico
//    (/100) -> Display numerico de Alto y Ancho de cluster

/* Estructuras de datos globales */

int C[CC][5];
// Filas: CC cilindros
// Cols: 0: estado del cil 1,x: origen llamada (izq) 2,X: destino llam. (der)
//                         3,t: comienzo llamada (abajo) 4,T: fin llam. (arr)
// estados: 0=no hay cilindro definido 1=borrado
// 2=libre 3=disponible (valor sin uso) 4=del cluster 5=guardado tras la poda
int   L[LL][5];
char  O[ROWS][COLS];
int   v=500;         // Longitud de la ventana (=500/100=5)
unsigned int U65536;// Variable que guarda un entero aleatorio
float lambda;        // Intensidad del proceso de Poisson dada por el usuario
float mu;            // Media de la long exponencial de cable, dada por el us.
int   x1=0,X1=0,t1=0,T1=0,x2=0,X2=0,t2=0,T2=0,K,KK; //Rectangulos, Contadores
char  ClusterListo=0, PodaLista=0, IteracionLista=0;//Flags
char  i=0,N=0;       //Contador layers, Contador cilindros cluster

/* Declaracion de subrutinas */

void  Mode(int mode_code);
// Establece el modo video (via ROM-BIOS, ver cap 25, Manual C, H. Schildt)
void  Pallette(int pnum);
// Establece la paleta  0: ro,am.ve,ne 1: bla,cele,fucsia,ne (se usa esta)
void  PaintPoint(int row, int col, char color);
// Dibuja un pixel usando el ROM-BIOS
// (0,0) arriba a la izquierda (como matriz)
void  LoadLO(void);
// Carga en L y en O el cilindro x1,X1,t1,T1
void  LoadCO(char EstBusc1, char EstBusc2, char EstQueQueda, char color);
// Carga o borra en C y en O (fondo=1) el cilindro x1,X1,t1,T1
void  NuevaCapa(void);
// Se calcula la capa i+1. Se actualiza numero N de cilindros cluster. i++
void  OutBox(char color, char interm);
// Pone en matriz O el cil x1X1t1T1 con color (fondo=1) intermitente o no
void  DisplayOutput(void);
// Vuelca la matriz O a la pantalla
char  Inters(void);
// Verifica si 2 rectangulos se intersecan y devuelve 0 o 1
void RemarcaCluster(void);
// Remarca en negro los cilindros del custer (4) y realiza el Display
void  RndInt(void);      // Pone en U65536 un entero al azar entre 0 y 65535
float Exp(float mm);     // Devuelve un real con distribucion Exponencial(mm)
int   Poisson(float ll); // Devuelve un entero con distribucion Poisson(ll)
int   Unif(int a, int b);// Devuelve un entero al azar entre a y b inclusive


/* Main */


void main(void)
{

/* Inicializaciones */

clrscr(); int m,n; // variables contadoras para ciclos
for(m=0; m<CC; m++) {for(n=0; n<5; n++) C[m][n]=0;}
for(m=0; m<LL; m++) {for(n=1; n<5; n++) L[m][n]=0;}
for(m=0;m<ROWS;m++){for(n=0;n<COLS;n++) {O[m][n]=1; O[0][n]=0;}}
//1=celeste 0=negro

clrscr();
printf("\n\nPROGRAMA QUE IMPLEMENTA EL ALGORITMO BACKWARD-FORWARD\n");
printf("PARA GENERAR UNA MUESTRA DE LLAMADAS\n");
printf("EN UNA VENTANA DE LONGITUD %d DE UN CABLE INFINITO\n",v/100);
printf("CADA LLAMADA SE ORIGINA EN UN PUNTO DE UN PROCESO DE POISSON ");
```

```
printf("EN LA RECTA\nOCUPA UNA PORCION DE CABLE ALEATORIA EXPONENCIAL\n");
printf("(POR LO TANTO, NO UNIFORMEMENTE ACOTADA)\n");
printf("Y SI NO SE SUPERPONE CON OTRA TIENEN DURACIÀN EXPONENCIAL ");
printf("DE PARAMETRO 1\n\n");
printf("Media para porcion (exponencial) de cable que ocupan ");
printf("las llamadas (.01 a 1)? "); scanf("%f",&mu);
mu=(mu>=.01)?mu:.01;mu=(mu<=1)?mu:1;
printf("mu=%1.2f da un lambda critico del ",mu);
printf("proceso de ramificacion = %1.2f\n",1/(mu*(1+sqrt(2))));
printf("Lambda para intensidad de llamadas (media) (.01 a 1)? ");
scanf("%f",&lambda);
lambda=(lambda>=.01)?lambda:.01;lambda=(lambda<=1)?lambda:1;
printf("Semilla para el generador pseudoaleatorio (1 a 10000)? ");
scanf("%d",&U65536);
U65536=(U65536>0)?U65536:1;U65536=(U65536<10001)?U65536:10000;
gotoxy(20,25); printf("Presione cualquier tecla para continuar");
getch(); clrscr();

/* Layer i=0 (ventana) */
x1=0; X1=v; t1=0; T1=i; LoadLO();
// Generacion de cilindros que nacen en sector 1 (bajo la ventana):
// Cantidad: Poiss(lambda*longitud ventana/100*1)
K=Poisson(lambda*v/100); KK=K;
// Ubicacion
for(m=0;m<K;m++){
                x1=Unif(0,v);X1=x1+Exp(mu*100);t1=-Exp(100);T1=Exp(100);
                printf("  %d %d %d %d,",x1,X1,t1,T1);
                LoadCO(0,1,4,0);} // 4 de cluster 0=negro
printf("\nCilindros que nacen bajo la ventana: %d\n",K);
getch();
// Generacion de cilindros que nacen en sector 2 (abajo-izq de la ventana):
// Cant: Poiss(lambda*mu*1)
K=Poisson(lambda*mu); KK=KK+K;
// Ubicacion
for(m=0;m<K;m++){
                x1=-Exp(mu*100);X1=Exp(mu*100);t1=-Exp(100);T1=Exp(100);
                printf("  %d %d %d %d,",x1,X1,t1,T1);
                LoadCO(0,1,4,0);} // 4 de cluster 0=negro
printf("\nCilindros que nacen abajo a la izquierda de la ventana: %d\n",K);

// Se calcula capa i (i=1). Si no aumento #cils cluster, ClusterListo=1. i++
if (KK==0) { ClusterListo=1; printf("\n Cluster Listo!\n");}
KK=0; NuevaCapa(); getch(); DisplayOutput();

/* Layer i */

while(ClusterListo==0)
{
// I) Se generan cilindros libres (2) que intersecan capa i

// Generacion de cilindros que nacen en sector 1 (la capa i):
// Cantidad: Poisson(lambda*ancho/100*alto/100)
K=Poisson(lambda*(L[i][2]-L[i][1])/100*(-L[i][3])/100);
// Ubicacion
for(m=0;m<K;m++){ x1=Unif(L[i][1],L[i][2]);X1=x1+Exp(mu*100);
                t1=Unif(L[i][3],L[i][4]);T1=t1+Exp(100);
                printf("  %d %d %d %d,",x1,X1,t1,T1);
                LoadCO(0,1,2,2);} // 2 de libres, 2=fucsia
printf("\nCilindros que nacen en la capa %d: %d\n",i,K);
getch();

// Generacion de cilindros que nacen en sector 2 (izq de la capa i):
// Cantidad: Poisson(lambda*mu*alto/100)
K=Poisson(lambda*mu*(-L[i][3])/100);
// Ubicacion
for(m=0;m<K;m++){ x1=L[i][1]-Exp(mu*100);X1=L[i][1]+Exp(mu*100);
                t1=Unif(L[i][3],L[i][4]);T1=t1+Exp(100);
                printf("  %d %d %d %d,",x1,X1,t1,T1);
                LoadCO(0,1,2,2);} // 2 de libres, 2=fucsia
printf("\nCilindros que nacen a la izquierda de la capa %d: %d\n",i,K);
getch();

// Generacion de cilindros que nacen en sector 3 (bajo la capa i):
// Cantidad: Poisson(lambda*ancho/100*1)
K=Poisson(lambda*(L[i][2]-L[i][1])/100);
// Ubicacion
for(m=0;m<K;m++){ x1=Unif(L[i][1],L[i][2]);X1=x1+Exp(mu*100);
                t1=L[i][3]-Exp(100);T1=L[i][3]+Exp(100);
                printf("  %d %d %d %d,",x1,X1,t1,T1);
                LoadCO(0,1,2,2);} // 2 de libres, 2=fucsia
printf("\nCilindros que nacen bajo la capa %d: %d\n",i,K);
getch();

// Generacion de cilindros que nacen en sector 4 (abajo-izq de la capa i):
// Cantidad: Poisson(lambda*mu*1)
K=Poisson(lambda*mu);
// Ubicacion
for(m=0;m<K;m++){ x1=L[i][1]-Exp(mu*100);X1=L[1][1]+Exp(mu*100);
```

```
                t1=L[i][3]-Exp(100);T1=L[i][3]+Exp(100);
                printf("  %d %d %d %d,",x1,X1,t1,T1);
                LoadCO(0,1,2,2);} // 2 de libres , 2=fucsia
printf("\nCilindros que nacen abajo a la izquierda de la capa %d: %d\n",i,K);
getch(); RemarcaCluster();

// II) Borrado (paso a 1) de los libres (2) que intersecan la capa i-1
x2=L[i-1][1]; X2=L[i-1][2]; t2=L[i-1][3]; T2=L[i-1][4];
K=0; for(m=0;m<CC;m++)
{         if (C[m][0]==0) m=CC;
         else {x1=C[m][1];X1=C[m][2];t1=C[m][3];T1=C[m][4];
               if(C[m][0]==2&&Inters()) {C[m][0]=1; // Pasa a borrado
               K++;printf(" m=%d %d %d %d %d,",m,x1,X1,t1,T1); OutBox(3,0);}
              }
}
printf("\nCils libres que intersecan capa %d pero tamb la %d: %d\n",i,i-1,K);
getch(); RemarcaCluster();

// III) Libres (2) que intersecan cluster pasan a 'del cluster' (4)

K=0;
IteracionLista=0;           // flag en 0
while(IteracionLista==0)
{  IteracionLista=1;        // flag en 1 para que salga si no agrega cilindro
   for(m=0;m<CC;m++)        // recorre los cilindros
   {   if (C[m][0]==0) {KK=m; m=CC;}
       else {x1=C[m][1];X1=C[m][2];t1=C[m][3];T1=C[m][4];
            for(n=0;n<CC;n++)
               {if (C[n][0]==0) n=CC;
               else {x2=C[n][1];X2=C[n][2];t2=C[n][3];T2=C[n][4];
                  if (C[m][0]==2 && C[n][0]==4 && Inters() && (t1<=t2))
                      {C[m][0]=4; printf(" m=%d %d %d %d %d,",m,x1,X1,t1,T1);
                      OutBox(0,0);K++;n=CC;// 0=negro
                      IteracionLista=0;}   // flag en 0, agrego un cilindro
                     }
               }
            }
   }
}// Cierra el while (IteracionLista)
printf("\n Cilindros libres del paso %d que pasaron al cluster: %d\n",i,K);
printf("Cantidad de cilindros usados: %d\n",KK); KK=0;
if (K==0) { ClusterListo=1; printf("\n Cluster Listo!\n");}
getch(); DisplayOutput();

NuevaCapa();

}// Cierra el while (ClusterListo)


/* Poda */

K=0; if (N==0) PodaLista=1;
while(PodaLista==0)
{
// Se pasa a (5) el cilindro que nacio primero de todos los (4)
t2=1; n=-1; for(m=0;m<CC;m++)
              if (C[m][0]==0) m=CC;
              else if (C[m][0]==4 && C[m][3]<t2)  {t2=C[m][3]; n=m;}

if (n!=-1) {C[n][0]=5; x2=C[n][1]; X2=C[n][2]; T2=C[n][4];
           K++; printf("m=%d %d %d %d %d\n",n,x2,X2,t2,T2);}
else PodaLista=1;
// Pasan a (1) todos los cils que intersecan este cil, salvo el (salvo los 5)
// (nada que lo interseca puede vivir (nazca antes o despues) pq el es vivo)
if(PodaLista==0) {for(m=0;m<CC;m++)
                  {if (C[m][0]==0) m=CC;
                   else if (C[m][0]!=5)
                      {x1=C[m][1]; X1=C[m][2]; t1=C[m][3]; T1=C[m][4];
                      if (Inters()) {C[m][0]=1; OutBox(3,0);}}}}// A borrado

}// Cierra el while
printf("Estos %d cilindros quedaron vivos tras la poda",K);
getch(); clrscr(); RemarcaCluster();
getch();

}// Fin main


/* Subrutinas */

/* Display Grafico*/
void DisplayOutput(void)
{
int m,n; // variables para ciclos
Mode(4); Pallette(1);
printf("\n                              Ventana:");
printf("                                 (0,0) a\n");
printf("                                 (0,5)\n\n");
```

```
printf("                                    Intensid");
printf("                                     media:\n");
printf("                                      %1.2f\n\n",lambda);
printf("                                    Longitud");
printf("                                     media:\n");
printf("                                      %1.2f\n\n",mu);
printf("                                    Paso: %d\n\n",i);
printf("                                    #Cilinds");
printf("                                    cluster:");
printf("                                     %3d\n\n",N);
printf("                                    Alto:\n");
printf("                                      %2.2f\n\n",-L[i][3]/100.);
printf("                                    Ancho:\n");
printf("                                      %2.2f",(L[i][2]-L[i][1])/100.);
for(m=0; m<200; m++) {for(n=0; n<250; n++) PaintPoint(m,n,O[m][n]);}
getch(); Mode(2);} /* Return to standard mode */

/* Set video mode */
void Mode(int mode_code)
{
  union REGS r;
  r.h.al = mode_code;
  r.h.ah = 0;
  int86(0x10, &r, &r);
}
/* Set pallette */
void Pallette(int pnum)
{
  union REGS r;
  r.h.bh = 1; /* code for graphic mode 4 */
  r.h.bl = pnum;
  r.h.ah = 11; /* pallette function */
  int86(0x10, &r, &r);
}
/* Draw pixel in x,y with specified color */
void PaintPoint(int row, int col, char color)
{
  union REGS r;
  if(row<0||row>199||col<0||col>319) return; /* return if out of range */
  r.h.ah = 12; /* for writing */
  r.h.al = color; /* color */
  r.x.dx = row; /* row */
  r.x.cx = col; /* column */
  int86(0x10, &r, &r);
}
/* Se calcula la capa i+1. Se actualiza numero N de cilindros cluster. i++ */
void NuevaCapa(void)
{
int m,M=0; x1=L[i][1];X1=L[i][2];t1=L[i][3];T1=0;
for(m=0;m<CC;m++)                                   {
  if (C[m][0]==0) m=CC;
  else if (C[m][0]==4) {x1=(C[m][1]<x1)?C[m][1]:x1;X1=(C[m][2]>X1)?C[m][2]:X1;
              t1=(C[m][3]<t1)?C[m][3]:t1; M++;} }// Se cuentan cils cluster
if (ClusterListo==0) i++; // Aumenta numero capa (y paso) si aumento tama¤o
N=M; LoadLO();            // Va a cargar nueva capa solo con cils cluster (4)
}
/* Carga en L[i][] y en O el cilindro x1,X1,t1,T1 */
void  LoadLO(void)
{
L[i][1]=x1; L[i][2]=X1; L[i][3]=t1; L[i][4]=T1; OutBox(2,1);
}

/* Carga o borra en C y en O (fondo=1) el cilindro x1,X1,t1,T1 */
void LoadCO(char EstBusc1, char EstBusc2, char EstQueQueda, char color)
{
int m; for(m=0;m<CC;m++)                                  {
       if (C[m][0]==EstBusc1 || C[m][0]==EstBusc2) {
          C[m][0]=EstQueQueda;C[m][1]=x1;C[m][2]=X1;C[m][3]=t1;C[m][4]=T1;
          OutBox(color,0); m=CC;                    }     }
}

/* Incorpora a matriz O el rectangulo x1,X1,t1,T1 con color dado (fondo=1)*/
void OutBox(char color, char interm)
{
int m;
x1=x1/10.+.5+100; X1=X1/10.+.5+100; t1=-t1/10.-.5; T1=-T1/10.-.5-10*interm*i;
if(x1>=0&&x1<COLS)  for(m=T1;m<=t1;m=m+1+interm)
// +interm para que los dibujos de layers sean intermitentes
                    if(m>=0&&m<=ROWS)O[m+(interm==0||i)][x1]=color;
// +(interm==0||i) p'q' corra todo el output para abajo, salvo la ventana
// (si no, la tapan mucho los cilindros)
if(X1>=0&&X1<COLS)  for(m=T1;m<=t1;m=m+1+interm)
                    if(m>=0&&m<=ROWS)O[m+(interm==0||i)][X1]=color;
if(T1>=0&&T1<ROWS-1)for(m=x1;m<=X1;m=m+1+interm)
                    if(m>=0&&m<=COLS)O[T1+(interm==0||i)][m]=color;
if(t1>=0&&t1<ROWS-1)for(m=x1;m<=X1;m=m+1+interm)
                    if(m>=0&&m<=COLS)O[t1+(interm==0||i)][m]=color;
}
```

```
/* Verifica si 2 rectangulos se intersecan */
char Inters(void)
{
char TRUE=0;
if( (X1-x1+X2-x2 >= X1*(X1>X2)+X2*(X1<=X2)-(x1*(x1<x2)+x2*(x1>=x2))) &&
    (T1-t1+T2-t2 >= T1*(T1>T2)+T2*(T1<=T2)-(t1*(t1<t2)+t2*(t1>=t2))) )TRUE=1;
return TRUE;
}
/* Remarca capas y los cilindros del custer (4) Realiza el Display */
void RemarcaCluster(void)
{int m; for(m=1;m<=i;m++){x1=L[m][1];X1=L[m][2];t1=L[m][3];T1=0;OutBox(2,1);}
for(m=0;m<CC;m++) if (C[m][0]==0)m=CC;else if(C[m][0]==4||C[m][0]==5)
{x1=C[m][1];X1=C[m][2];t1=C[m][3];T1=C[m][4]; OutBox(0,0);} DisplayOutput();}

/* Entero al azar entre 0 y 65535 */
void RndInt(void)          //  X=aX+c mod m con m=65536, a=10001, c=10001
{U65536=10001*U65536+10001;}  // (c,m)=1, a-1 mult de todo p q'|m, a-1=4*2500

/* Variable exponencial continua*/
float Exp(float mm)
{RndInt(); return -log(U65536/65536.)*mm;}

/* Variable Poisson */
int Poisson(float ll)
{float sum=Exp(1); int k=0; while(sum<ll) {k++; sum=sum+Exp(1);} return k;}

/* Variable uniforme entera en [a,b] */
int Unif(int a, int b)
{RndInt(); return floor(U65536/65536.*(b+1-a)+a);}
```

Referencias

Armitage P. (1949)
An overlap problem arising in particle counting
Biometrika 36, 1949, 257–266.

Athreya K., Ney P. (1972)
Branching processes
Springer, New York

Biggins J.D. (1997)
How fast does a general branching random walk spread?
Classical and modern branching processes (Athreya K. Jagers P. eds.)
Springer, New York

Corte H., Dodson C.T.J. (1969)
Über die Verteilung der Massendichte in Papier. Erster Teil: Theoretische Grundalgen.
Das Papier 23, 1969, 381–394.

Dawson D.A., Wu Y. (1997)
Multilevel multitype branching models of an information system.
Classical and modern branching processes (Athreya K. Jagers P. eds.)
Springer, New York

De Candolle A. (1873)
Histoires de sciences et de savants depuis deux Siècles, Geneva, 1873.

Diggle P.J. (1981)
Binary mosaics and the spatial pattern of heather
Biometrics 37, 1981, 531–539.

Edelstein-Keshet L. (1987)
Mathematical models in biology
Birkhäuser Mathematics Series. Random House, New York.

Fernández R. , Ferrari P.A. , Garcia N.L. (1998)
Measures on contour, polymer or animal models. A probabilistic approach.
Markov Processes and Related Fields 4, 1998, no. 4, 479–497.

Fernández R. , Ferrari P.A. , Garcia N.L. (2001)
Loss network representation of Peierls contours
Ann. Probab. 29, 2001, no. 2, 902–937.

Fernández R. , Ferrari P.A. , Garcia N.L. (2002)
Perfect simulation for interacting point processes, loss networks and Ising models.
Stoch. Process. Appl. 102, 2002, no. 1, 63–88.

Ferrari P.A., Garcia N. (1998)
One-dimensional loss networks and conditioned M/G/∞ queues.
J. Appl. Probab. 35, no. 4, 1998, 963–975.

Galton F. (1873)
Problem 4001, Educational Times, 1 April, p. 17, 1873

Guttorp P. (1991)
Statistical inference for branching processes
John Wiley and Sons, New York.

Hall P. (1985)
On continuum percolation
Ann. Prob. 13, 1985, 1250–1266

Hansen P.E. (1986)
Leslie matrix models: a mathematical survey
Papers on mathematical ecology, 1, 1–139, DM, 1986-2, K.Marx Univ. Econom., Budapest

Harris T. (1963)
The theory of branching processes.
Spinger, Berlin

Jacod J., Joathon P. (1971)
Use of random genetic models in the study of sedimentary processes
J. Assoc. Math. Geol. 3, 1971, 265–279.

Karlin S. (1966)
A first course in stochastic processes
Academic, New York.

Kelly F.P. (1991)
Loss networks
The Annals of Applied Probability 1 (3), 1991, 319–378.

Kolmogorov A.N. (1937)
Statistical theory of cristalization of metals
Bull. Acad. Sci. USSR Mat. Ser. 1, 1937, 75–83.

Kolmogorov A.N., Dmitriev N.A. (1947)
Branching stochastic processes.
Dokl. Akad. Nauk SSSR 56, 1947, 5–8.

Lelsie P.H. (1945)
On the use of matrices in certain population mathematics.
Biometrika 33, 1945, 183–212.

Marchant J.C., Dillon P.L.P. (1961)
Correlation between random-dot samples and the photographic emulsion
J. Opt. Soc. Amer., 51, 1961, 641–644.

Maric N. (2002)
Simulaçao perfeita para redes com perdas.
Tese de mestrado. Universidade Estadual de Campinas, Brasil

Meester R., Roy R. (1996)
Continuum percolation
Cambridge University Press

Mode C.J. (1971)
Multitype Branching Processes. Theory and Applications.
American Elsevier, New York.

Pollard J.H. (1966)
On the use of the direct matrix product in analysing certain stochastic population models
Biometrika 53, 1966, 397–415.

Propp J.G., Wilson D.B. (1996)
Exact sampling with coupled Markov chains and applications to statistical mechanics
Proceedings of the Seventh International Conference on Random Structures and Algorithms
(Atlanta, GA, 1995) Vol. 9, 223–252.

Ross S.M. (1983)
Stochastic Processes
John Wiley and Sons, New York.

Tetzlaff G.T. (2002)
Simulación de clusters de percolación continua
VIII Congreso Argentino de Ciencias de la Computación
Universidad de Buenos Aires, Argentina, octubre 2002.

Tetzlaff G.T. (2004)
Criticality in unbounded-types branching processes
http://www.arxiv.org/PS_cache/math/pdf/0402/0402156.pdf (Arxiv.org e-print archive)
Enviado a Stochastic Processes and their Applications

Watson H.W. (1873)
Solution to problem 4001, Educational Times, 1 August, pp. 115-116, 1873.

Widom J.S., Rowlinson B. (1970)
New model for the study of liquid-vapor phase transitions
J. Chem. Phys. 52, 1970, 1670–1684.

Woodroofe M. (1975)
Probability with applications
McGraw Hill Kogakusha, Tokio.