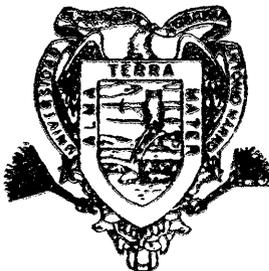


PROBLEMAS DE ENUMERACION EN ESTADISTICA
NO PARAMETRICA

GERARDO SANCHEZ MARTINEZ

T E S I S

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS
EN ESTADISTICA EXPERIMENTAL

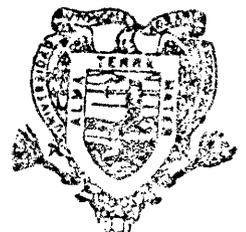


Universidad Autónoma Agraria
Antonio Narro

PROGRAMA DE GRADUADOS

Buenavista, Saltillo, Coah.

JUNIO DE 1999



BIBLIOTECA
EGIDIO G. REBONATO
BANCO DE TESIS
U.A.A.A.N.

11043

UNIVERSIDAD AUTÓNOMA AGRARIA

ANTONIO NARRO

SUBDIRECCIÓN DE POSTGRADO

Problemas de Enumeración en Estadística No Paramétrica

TESIS

por

GERARDO SÁNCHEZ MARTÍNEZ

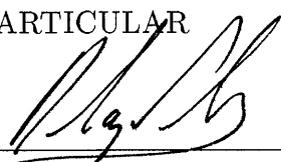
Elaborada bajo la supervisión del Comité Particular de Asesoría y aprobada
como requisito parcial para optar al grado de

MAESTRO EN CIENCIAS

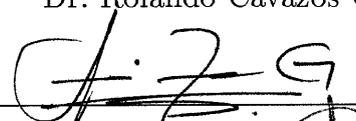
en Estadística Experimental

COMITÉ PARTICULAR

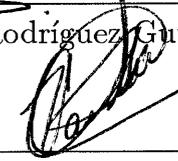
Asesor Principal:

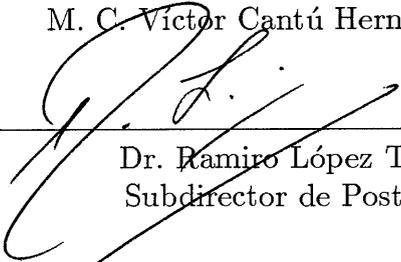

Dr. Rolando Cavazos Cadena

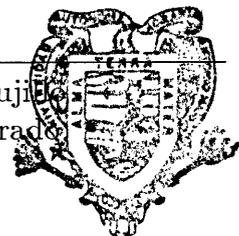
Asesor:


M. C. Luis Rodríguez Gutiérrez

Asesor:


M. C. Víctor Cantú Hernández


Dr. Ramiro López Trujillo
Subdirector de Postgrado



Buenavista, Saltillo, Coahuila. Junio de 1999

BIBLIOTECA
EGIDIO G. REBONATO
BANCO DE TESIS
U.A.A.A.N.

Agradecimientos

Deseo expresar mi agradecimiento a:

- Primeramente a **Dios** por el don de la vida y por abrirme siempre una ventana cuando se han cerrado las puertas.
- A la **Universidad Autónoma Agraria “Antonio Narro”** por brindarme la oportunidad de superarme.
- En forma especial al **Dr. Rolando Cavazos Cadena** , por sus consejos, paciencia, orientación y su valiosa ayuda en la elaboración de la tesis.
- A los sinodales **M.C. Luis Rodríguez Gutiérrez** y **M.C. Víctor Cantú Hernández** por sus comentarios y revisión de este trabajo.
- A todos los maestros del Departamento de **Estadística** por su apoyo.
- A mis compañeros de estudio y a mis amigos.

Dedicatoria

- A MIS PADRES

Genoveva Martínez y con especial cariño y gran admiración a quien fue ejemplo y guía en mi vida, a un gran hombre y persona, **Ismael Sánchez Galván**.

- A MIS HERMANOS

Con especial afecto a **Sergio Sánchez Martínez** por su apoyo incondicional.

COMPENDIO

Problemas de Enumeración en Estadística No Paramétrica

por

GERARDO SÁNCHEZ MARTÍNEZ

TESIS

MAESTRÍA EN CIENCIAS
en Estadística Experimental

UNIVERSIDAD AUTÓNOMA AGRARIA
ANTONIO NARRO

Buenavista, Saltillo, Coahuila, México. Junio de 1999

Rolando Cavazos Cadena — Asesor

Palabras Clave: Algoritmo de Listado de Permutaciones, Listado de Subconjuntos, Relación de Orden Total, Orden Lexicográfico, Estadísticos U , Estadísticos de Rango, Implementación Computacional.

Dado el conjunto de los primeros N enteros positivos, este trabajo trata sobre el problema de producir una lista de todas sus permutaciones o subconjuntos de tamaño r . La motivación para estudiar este tema proviene de dos vertientes: su

importancia en la aplicación de métodos no paramétricos basados en los denominados estadísticos U , y el análisis de la bondad con que una distribución asintótica aproxima a la distribución actual de un estadístico de rango lineal. Para abordar los problemas de enumeración, se desarrolla una teoría general que establece la equivalencia entre el listado de un conjunto y la definición de una relación de orden total en el mismo, resultado que permite formular un algoritmo general de enumeración. Después de identificar a las familias de permutaciones y subconjuntos con vectores r -dimensionales, la solución a los problemas analizados se obtiene aplicando la teoría general, y considerando al espacio Euclideo de dimensión r como un conjunto lexicográficamente ordenado. Además, los procedimientos de listado se implementan computacionalmente.

ABSTRACT

Enumeration Problems in Nonparametric Statistics

by

GERARDO SÁNCHEZ MARTÍNEZ

THESIS

MASTER of SCIENCE

Experimental Statistics

UNIVERSIDAD AUTÓNOMA AGRARIA

ANTONIO NARRO

Buenavista, Saltillo, Coahuila, México. June, 1999

Rolando Cavazos Cadena — Advisor

Key Words: Algorithm for Listing Permutations, Listing Subsets of a Fixed Size, Total Order Relations, Lexicographical Order, U -Statistics, Linear Rank Statistics, Computational Implementation.

Given a set consisting of the first N positive integers, this work concerns the problem of listing all its permutations or subsets of a given size r . The motivation behind this work stems from the theory of nonparametric statistics, specially

from the application of U -statistics as unbiased estimators, and the use of linear rank statistics in hypothesis testing. The listing problems are approached via a general theory that establishes the equivalence between producing a list a set \mathcal{A} , and the introduction of a complete (total) order in \mathcal{A} . This result is used to formulate a general listing procedure which, after identifying subsets and permutations with vectors in a Euclidean space endowed with the lexicographical order, renders an algorithm to solve the listing problems. The computational implementation of the results is illustrated.

CONTENIDO

	Página
El Plan de Trabajo	1
Introducción	1
Objetivos	3
La Organización	4
Estimación No Paramétrica	6
Introducción	6
El Contexto	7
Funciones Estimables	8
Estadísticos U	10
Cómputo de un Estadístico U	16
Conclusión	19
Teoría General Sobre Listado de un Conjunto	20
Introducción	20
El Ángulo Estadístico	22
Relaciones de Orden	24
El Sucesor de un Elemento	29
El Procedimiento General	33
El Orden Lexicográfico	37
Conclusión	40
Permutaciones de Longitud Máxima	42
Introducción	42
La Perspectiva Estadística	44
Las Permutaciones Mínima y Máxima	45
El Índice de Ruptura	47

	Página
Caracterización del Sucesor	50
Cálculo del Sucesor de una Permutación	51
Implementación del Algoritmo	59
Conclusión	65
Generación de Subconjuntos	66
Introducción	66
El Contexto Estadístico	68
Los Subconjuntos Máximo y Mínimo	69
Sucesores e Índices de Ruptura	72
Caracterización del Sucesor	75
Cálculo del Sucesor de un Conjunto	76
Implementación del Algoritmo	85
Conclusión	90
Retrospectiva	92
Literatura Citada	94

Capítulo 1

El Plan de Trabajo

El propósito de este capítulo es describir el contenido de esta tesis, plantear los objetivos que se pretenden alcanzar, y la manera en que se ha organizado la presentación del trabajo desarrollado.

Introducción

Este trabajo trata sobre un problema de gran importancia para la aplicación de técnicas estadísticas en un entorno no paramétrico, a saber, la generación de una lista de todas las permutaciones y de todos los subconjuntos de tamaño r de una población finita de números enteros. La relevancia de esta problema proviene de su estrecha relación con dos cuestiones fundamentales: (i) La evaluación de estadísticos del denominado tipo U , los cuales son utilizados para construir estimadores insesgados que tiene propiedades de optimalidad, y (ii) la determinación de la distribución exacta de estadísticos de rango lineal para tamaños ‘pequeños’ de muestra; aunque en los Capítulos 2 y 5 se hará una presentación breve sobre esta ideas, una exposición detallada puede encontrarse en Randles y Wolfe (1981), o Meddis(1984).

Para clarificar la importancia de los problemas que se considerarán a continuación, es conveniente describir, así sea brevemente, la forma que un estadístico de tipo U adopta, y las dificultades que se enfrentan en las aplicaciones para evaluarlo: Cada estadístico de tipo U tiene asociada una función $h(x_1, x_2, \dots, x_r)$ de r variables. Cuando se observa una muestra de tamaño $n > r$, la determinación del estadístico requiere los siguientes pasos: (i) Para cada sucesión (s_1, s_2, \dots, s_r) , con $1 \leq s_1 < s_2 < \dots < s_r \leq n$, calcule $h(X_{s_1}, X_{s_2}, \dots, X_{s_r})$, y (ii) el estadístico

U asociado a h se obtiene promediando todas las cantidades encontradas en el paso anterior.

Como ya se ha mencionado, los estadísticos que se obtienen de esta forma tienen propiedades de optimalidad como estimadores puntuales, aspecto que será abordado con cierta profundidad en el Capítulo 2. Por otro lado, es claro que la utilidad práctica de un estimador depende de que se pueda calcular el valor que asume para cada conjunto de datos disponibles, y a primera vista puede parecer que, en una situación, concreta, este aspecto no debe significar un problema sustancial. Sin embargo, un momento de reflexión sobre el procedimiento delineado para construir estadísticos de tipo U , muestra un aspecto que puede pasar desapercibido, y que constituye un obstáculo para su evaluación, a saber, deben encontrarse todas las sucesiones (s_1, s_2, \dots, s_r) de enteros que satisfacen $1 \leq s_1 < s_2 < \dots < s_r \leq n$, las cuales se conocen como *permutaciones de tamaño r de n objetos*, y tienen la siguiente peculiaridad: forman un conjunto ‘enorme’ aún para tamaños de muestra relativamente ‘pequeños’ (Even,1973, Liu,1978). Por ejemplo, para $n = 50$ y $r = 2$, hay 2450 permutaciones de tamaño dos, las cuales deben listarse si acaso se desea determinar el valor asumido por el estadístico U correspondiente a una función de $r = 2$ variables. Más aún, en el caso que se discute, la varianza del estadístico U debe estimarse para construir intervalos de confianza correspondientes al parámetro bajo consideración, y esta última requiere calcular un estadístico U cuya función h depende de cuatro ($= 2r$) variables, de manera que para estimar dicha varianza se requiere encontrar todas las permutaciones de tamaño cuatro de los primeros 50 enteros positivos; el aspecto notable es que hay 5527200 de tales permutaciones. Desde un punto de vista general, hay un caso en el cual puede evitarse la tarea de calcular todas la permutaciones de tamaño r de los primeros n enteros positivos; esta es la situación que ocurre cuando la función $h(x_1, x_2, \dots, x_r)$ de la cual se obtiene un estadístico U es *simétrica* (Randles y Wolfe (1981)). En este caso, en vez de determinar permutaciones de tamaño r

para calcular el estadístico, se requiere encontrar todos los *subconjuntos* de r elementos de $\{1, 2, \dots, n\}$; si $n = 50$ y $r = 2$ deben determinarse explícitamente 1225 subconjuntos de tamaño dos para evaluar el U estadístico, y 230300 subconjuntos de tamaño cuatro para estimar su varianza; esta última cifra, aunque menor que los más de cinco millones y medio de permutaciones de tamaño cuatro, continúa siendo una cifra ‘impresionante’.

El enorme número de permutaciones o subconjuntos que deben encontrarse para evaluar un U -estadístico, pone de manifiesto la necesidad de disponer de un método para generar permutaciones o subconjuntos de una manera ‘ordenada’, esto es, sin omisiones, ni repeticiones.

Objetivos

Como se señaló en la sección precedente, el problema de generar una lista completa de permutaciones o de subconjuntos de tamaño r de una población finita, tiene enorme importancia en la implementación práctica de métodos no paramétricos basados en estadísticos de tipo U , lo cual proporciona una fuerte motivación para estudiar el problema de enumerar todas las permutaciones y subconjuntos de tamaño determinado que pueden extraerse de una población finita. Una forma de generar las permutaciones ha sido propuesta en Johnson (1963), donde se utiliza la idea de transponer elementos adyacentes para generar todas las permutaciones de tamaño n de la población $\{1, 2, \dots, n\}$. En esta tesis, la intención es desarrollar una teoría general de listado de conjuntos finitos, y enmarcar la generación de permutaciones y subconjuntos dentro de ese contexto; además del interés propio de este enfoque, las consideraciones anteriores muestran que los resultados que se obtengan tienen repercusiones importantes en la implementación de métodos no paramétricos en casos concretos.

Los objetivos que esta tesis se plantea son los siguientes:

- *Establecer* la equivalencia entre el problema de enumerar un conjunto finito, y

la formulación de un orden completo en el conjunto.

Este propósito es el primer paso hacia la elaboración de una teoría general de listado de conjuntos. La siguiente etapa es utilizar una ordenación completa como vehículo para generar de una lista.

- *Formular* un algoritmo general de listado de un conjunto finito en términos de un orden completo.

Después de introducir el orden lexicográfico en \mathbb{R}^r , y de identificar permutaciones y subconjuntos con tipos especiales de vectores, se buscará utilizar dicho orden para alcanzar el propósito fundamental de este trabajo:

- *Diseñar* un procedimiento para listar todas las permutaciones o subconjuntos de tamaño r de una población finita.

La Organización

Para alcanzar los objetivos propuestos, el trabajo subsecuente se ha organizado de la siguiente manera: En el Capítulo 2 se introducen las nociones de funciones estimables y de estadísticos del tipo U , estableciendo la optimalidad de éstos últimos como estimadores puntuales, y analizando de manera cuidadosa la construcción de intervalos de confianza para funciones estimables; la argumentación muestra la relevancia que los problemas de enumeración de permutaciones y subconjuntos tiene en las aplicaciones.

En el Capítulo 3 se considera el problema general de listado desde un punto de vista abstracto, estableciendo la equivalencia entre la introducción de un orden total y la enumeración de los elementos de un conjunto. El resultado teórico fundamental, es la formulación de un algoritmo general de listado de un conjunto finito en términos de un orden total. Posteriormente, en el Capítulo 4 se estudia el problema de construir una lista de todas las permutaciones del conjunto $\mathcal{P} = \{1, 2, \dots, N\}$; el principal resultado se establece como un algoritmo que implementa, para la familia de las permutaciones lexicográficamente ordenada,

el procedimiento general obtenido en el Capítulo 3. Siguiendo un enfoque similar, en el Capítulo 5 se formula un algoritmo que genera todos los subconjuntos de tamaño r de la población \mathcal{P} . Para cada algoritmo, la exposición incluye una codificación en lenguaje de alto nivel.

Finalmente, la exposición concluye en el Capítulo 5 con algunos comentarios breves.

Capítulo 2

Estimación No Paramétrica

Este capítulo trata sobre la construcción de estimadores insesgados en un contexto no paramétrico. Se introducen las ideas de función estimable y de estadístico U , enfatizando la importancia que ésta última noción tiene en la formulación de intervalos de confianza y de pruebas de hipótesis. La presentación pone en relieve la importancia que la enumeración de una familia de subconjuntos tiene para la implementación de métodos no paramétricos de inferencia

Introducción

El propósito de este capítulo es presentar el entorno estadístico que da origen a los problemas que son el tema de este trabajo, a saber, la enumeración de familias de subconjuntos y permutaciones. La exposición gira alrededor de la idea de *estadísticos* U , los cuales se emplean como estimadores en un contexto no paramétrico (Lehmann (1975), Randles y Wolfe (1981), Hollander y Wolfe (1981)). La exposición destaca la importancia de ese tipo de estadísticos, y enfatiza las dificultades que surgen al implementar métodos de inferencia basados en los mismos, las cuales provienen de la enorme carga computacional que se requiere en una implementación práctica.

El material ha sido organizado de la siguiente manera: En la Sección 2 se describe, a grandes rasgos, el contexto de trabajo básico en un marco no paramétrico, mientras que la Sección 3 se refiere a la idea de función estimable, noción que generaliza un concepto ampliamente utilizado en el estudio de modelos lineales. Posteriormente, los estadísticos U se introducen en la Sección 4 como un medio de construir estimadores que ‘utilizan toda la información en la muestra’;

se establece la optimalidad de dichos estimadores, en el sentido de que tienen varianza uniformemente mínima dentro de la clase de estimadores insesgados, y se analiza la teoría asintótica que permite establecer intervalos de confianza para una función estimable en términos de estadísticos U . En la Sección 5 se discute la determinación de dicho intervalo de confianza, prestando especial atención a la enorme cantidad de permutaciones o subconjuntos que deben determinarse, hecho que proporciona la motivación para diseñar un método que permita generarlos de manera organizada. Finalmente, la exposición concluye en la Sección 6 delineando la estrategia que se seguirá en los siguientes capítulos.

El Contexto

Un punto de partida común en un problema estadístico es una serie de observaciones X_1, X_2, \dots, X_n , las cuales, bajo ‘la hipótesis nula’, son independientes e idénticamente distribuidas. El rasgo característico en el enfoque no paramétrico, es que los supuestos sobre la distribución común de las variables X_i son bastante débiles, en el sentido de no se asume forma específica alguna para la función de distribución—denotada por F —de la cual provienen las observaciones. Para clarificar esta idea, suponga que X_1, X_2, \dots, X_n son los tiempos de vida útil de n artículos. En un contexto no paramétrico, se podría suponer, por ejemplo, que la función de distribución F es continua, y que esta cocentrada en el intervalo $(0, \infty)$, mientras que en el enfoque paramétrico usual, tal vez podría considerarse adecuado suponer que la distribución común de las observaciones es exponencial con parámetro $\lambda > 0$, la cual corresponde a la función de distribución $F_\lambda(x) = (1 - e^{-\lambda x})I_{(0, \infty)}(x)$, y el problema de inferencia en este caso se reduce a estimar, construir intervalos de confianza, o probar hipótesis acerca de λ .

Para introducir las ideas de este capítulo, se supondrá que

$$X_1, X_2, \dots, X_n \quad \text{es una muestra de la función de distribución } F, \quad (2.1)$$

donde $F \in \mathcal{F}$, y \mathcal{F} es una familia ‘amplia’ de funciones de distribución, cuyos miembros no siguen una forma específica determinada. Note que (2.1) equivale a

$$P[X_1 \leq x_1, X_2 \leq x_2, \dots, X_n \leq x_n] = F(x_1)F(x_2) \cdots F(x_n), \quad F \in \mathcal{F},$$

y a partir de esta igualdad se desprende que si $r \leq n$,

$$(X_1, X_2, \dots, X_r) \stackrel{d}{=} (X_{s_1}, X_{s_2}, \dots, X_{s_r}), \quad (2.2)$$

siempre que s_1, s_2, \dots, s_r sean enteros diferentes entre 1 y n , donde el símbolo ‘ $\stackrel{d}{=}$ ’ significa que los vectores involucrados poseen la misma distribución.

Funciones Estimables

En esta sección se introduce la noción de función estimable en el contexto de estadística no paramétrica, la cual es una generalización del concepto de *función lineal estimable* que desempeña un papel central en el análisis de datos cuando éstos se suponen generados de acuerdo a un modelo lineal (Graybill, 1985, Searle, 1979). Por ejemplo, bajo la hipótesis de normalidad, en el estudio de experimentos diseñados, los instrumentos para comparar las consecuencias de utilizar cada uno de los tratamientos involucrados son los contrastes entre efectos principales, los cuales son funciones lineales y estimables de los parámetros del modelo (Fisher y McDonald, 1978). De hecho, la teoría de los modelos lineales es, en gran parte, el análisis de las funciones lineales estimables, y de las propiedades que poseen sus estimadores y los métodos de inferencia asociados a los mismos. La idea de función estimable en un contexto no paramétrico, la cual se introduce a continuación, es una extensión importante de la noción familiar en el análisis de experimentos.

Definición 3.1. [Randles y Wolfe (1981).] Sea \mathcal{F} una familia de distribuciones. Una función $\gamma: \mathcal{F} \rightarrow \mathbb{R}$ es estimable respecto a \mathcal{F} , si existe una función $h: \mathbb{R}^n \rightarrow \mathbb{R}$ con la siguiente propiedad: Si X_1, X_2, \dots, X_r es una muestra de la distribución $F \in \mathcal{F}$, entonces

$$E_F[h(X_1, X_2, \dots, X_r)] = \gamma(F). \quad (3.1)$$

Así, una función es estimable respecto a \mathcal{F} , siempre y cuando pueda expresarse como el valor esperado de alguna función de r variables aleatorias que constituyen una muestra de $F \in \mathcal{F}$. El mínimo de los números r para los cuales es posible encontrar una función $h(\cdot)$ tal que (3.1) se satisfaga es el *grado de* $\gamma(\cdot)$, y la función $h(\cdot)$ en (3.1) es un *núcleo* de γ . A continuación se ilustra esta idea en algunos casos particulares; un ejemplo interesante desde el punto de vista económico se presenta en el Capítulo 5.

Ejemplos 3.1. (a) Sea $\gamma(F)$ el valor esperado de la distribución F . En este caso,

$$\gamma(\cdot) = E_F[X_1]$$

de manera que la función ‘esperanza de la distribución’ es estimable sin importar cual sea la familia \mathcal{F} , siempre que ésta consista de distribuciones con primer momento finito. Note que el núcleo de $\gamma(\cdot)$ es $h(x) = x$, la función identidad en \mathbb{R} .

(b) Suponga que $\gamma(F)$ es la varianza de la distribución F , esto es,

$$\gamma(F) = E_F[X_1^2] - (E_F[X_1])^2.$$

A continuación se verá esta función es estimable. Para comprobar esta característica, considere una muestra de tamaño dos de la distribución $F \in \mathcal{F}$. En este caso, $E_F[X_1] = E_F[X_2]$, pues X_1 y X_2 tienen la misma distribución, y entonces

$$(E_F[X_1])^2 = E_F[X_1]E_F[X_2] = E_F[X_1X_2],$$

pues la esperanza de un producto de variables aleatorias independientes es el producto de las esperanzas (Mood *et al.*, 1985, Dudewicz y Mishra, 1988). Por lo tanto,

$$\gamma(F) = E_F[X_1^2] - E_F[X_1X_2] = E_F[X_1^2 - X_1X_2],$$

y entonces $\gamma(F)$ se expresa como el valor esperado de la función $h(X_1, X_2) = X_1^2 - X_1X_2$, esto es, $\gamma(F)$ es estimable con respecto a la familia \mathcal{F} , cuando ésta consiste

de distribuciones con segundo momento finito; el núcleo de $\gamma(F)$ es $h(x_1, x_2) = x_1^2 - x_1x_2$, y el grado de $\gamma(F)$ en el caso presente es dos, siempre que la familia de distribuciones \mathcal{F} sea ‘lo suficientemente amplia’; para detalles sobre este aspecto, vea Randles y Wolfe (1981).

(c) Defina $\gamma(F)$ mediante

$$\gamma(F) = E_F[Y_{(n)} - Y_{(1)}],$$

donde $Y_{(n)}$ y $Y_{(1)}$ son el máximo y el mínimo, respectivamente, de una sucesión de variables aleatorias X_1, X_2, \dots, X_n obtenida de la distribución $F \in \mathcal{F}$. A partir de la expresión que define a $\gamma(F)$, es claro que $\gamma(\cdot)$ es estimable, y que el núcleo correspondiente es $h(X_1, X_2, \dots, X_n) = Y_{(n)} - Y_{(1)}$. Esta función $\gamma(F)$ es el rango esperado de una muestra de tamaño n , y es importante en procedimientos aplicados en el área de control estadístico de calidad (Saucedo, 1999). \square

Una razón fundamental por la cual las funciones estimables son importantes desde el punto de vista no paramétrico, es que admiten estimadores insesgados de varianza uniformemente mínima, aspecto que se analiza en la siguiente sección.

Estadísticos U

El problema de encontrar ‘buenos’ estimadores para una función estimable ha sido extensivamente estudiado en la literatura, y el análisis sobre este tema se remonta, por lo menos, a Hoeffding (1948). Antes de abordar esta cuestión desde un punto de vista formal, se discutirá de manera intuitiva el enfoque que se emplea.

Discusión. Suponga que $\gamma(F)$ es una función estimable, por ejemplo, de grado dos. De acuerdo a la Definición 3.1, esto significa que para toda función de distribución F en \mathcal{F} , se tiene que $\gamma(F) = E_F[h(X_1, X_2)]$, donde $h(\cdot, \cdot)$ es el núcleo de $\gamma(\cdot)$. El problema que se enfrenta cuando se dispone de una muestra de tamaño n ,

es construir un estimador de $\gamma(F)$ que utilice ‘toda la información’ contenida en la muestra. Para propósitos de ilustración, suponga que se dispone de una muestra de tamaño $n = 3$ proveniente de la distribución $F \in \mathcal{F}$. En este caso, como se observó en la Sección 2, se tiene que cada pareja (X_i, X_j) con $i \neq j$ tiene la misma distribución que (X_1, X_2) , de manera que

$$\gamma(F) = E_F[h(X_1, X_2)] = E_F[h(X_i, X_j)], \quad 1 \leq i \neq j \leq 3. \quad (4.1)$$

Esto significa que, además de $h(X_1, X_2)$, la muestra permite construir otros estimadores insesgados de $\gamma(F)$. Por ejemplo, $h(X_2, X_1)$ y $h(X_3, X_2)$, son estimadores insesgados de $\gamma(F)$. En el caso presente, la muestra de tamaño tres permite construir seis estimadores insesgados de $\gamma(F)$, los cuales son

$$\begin{array}{cccc} h(X_1, X_2), & h(X_2, X_1), & h(X_1, X_3), & h(X_3, X_1) \\ & h(X_2, X_3), & h(X_3, X_2). & \end{array}$$

Una manera de construir un estimador insesgado de $\gamma(F)$ que involucre a todas las observaciones X_1, X_2, \dots, X_n de manera simultánea, es promediar los estimadores insesgados anteriores, los cuales utilizan sólo dos observaciones a la vez. Al proceder de esta forma, se obtiene

$$\hat{\gamma} = \frac{h(X_1, X_2) + h(X_2, X_1) + h(X_1, X_3) + h(X_3, X_1) + h(X_2, X_3) + h(X_3, X_2)}{6},$$

o en notación más compacta,

$$\hat{\gamma} = \frac{1}{6} \sum_{(i,j): 1 \leq i \neq j \leq 3} h(X_i, X_j).$$

Es conveniente resumir la ruta seguida para obtener el lado derecho de esta igualdad:

Inicialmente, se cuenta con una muestra de tamaño $n = 3$: X_1, X_2, X_3

Se obtienen todas las parejas (i, j) , con $1 \leq i \neq j \leq 3$;

Después se calcula $h(X_i, X_j)$ para cada pareja (i, j) encontrada anteriormente.

Finalmente, se determina el promedio de las cantidades calculadas en la etapa precedente.

Esta construcción se generaliza a continuación.

Definición 4.1. Sea $h(x_1, x_2, \dots, x_r)$ una función de r variables, y suponga que X_1, X_2, \dots, X_n es una sucesión de n variables aleatorias. En este caso, el estadístico de tipo U asociado a la función h , se define mediante

$$U_h = U_h(X_1, X_2, \dots, X_n) = \frac{1}{(n)_r} \sum_S h(X_{s_1}, X_{s_2}, \dots, X_{s_r}), \quad (4.2)$$

donde la suma se extiende sobre la clase de todas las sucesiones $S = (s_1, s_2, \dots, s_r)$ de enteros que satisfacen $s_i \neq s_j$ si $i \neq j$, y $1 \leq s_i \leq n$ para todo i

Expresada en terminología más común, la sumatoria en (4.2) se extiende sobre todas las *permutaciones de tamaño r del conjunto $\{1, 2, \dots, n\}$* , cuyo número es

$$(n)_r = n(n-1) \cdots (n-r+1); \quad (4.3)$$

vea, por ejemplo, Dudewicz y Mishra (1988).

Observación 4.1. El proceso de obtener el estadístico de tipo U correspondiente a la función h y a la sucesión X_1, X_2, \dots, X_n , puede expresarse como sigue:

- (i) Obtenga las permutaciones de tamaño r del conjunto $\{1, 2, \dots, n\}$,
- (ii) Para cada una de las permutaciones $S = (s_1, s_2, \dots, s_r)$ determinadas en el paso anterior, calcule $h(X_{s_1}, X_{s_2}, \dots, X_{s_r})$, y
- (iii) La evaluación de U_h se efectúa promediando las cantidades encontradas en el paso previo.

La importancia de los estadísticos de tipo U se debe al siguiente resultado general.

Teorema 4.1. [Hoeffding (1948), Randles y Wolfe (1981).] Suponga que $\gamma(F)$ es una función estimable cuando F varía en la familia \mathcal{F} , y sea $h(x_1, x_2, \dots, x_r)$ un núcleo para $\gamma(\cdot)$, esto es,

$$\gamma(F) = E_F[h(X_1, X_2, \dots, X_r)], \quad F \in \mathcal{F}.$$

En este caso,

(i) El estadístico $U_h(X_1, X_2, \dots, X_n)$ es un estimador insesgado de $\gamma(F)$.

Más aún,

(ii) Si la variable $h(X_1, X_2, \dots, X_r)$ tiene segundo momento finito, y la familia \mathcal{F} contiene a todas las distribuciones continuas, entonces U_h es el estimador insesgado de $\gamma(F)$ con varianza uniformemente mínima.

Demostración. (i) Como se mencionó en la Sección 2, cuando la sucesión de números enteros s_1, s_2, \dots, s_r forman una permutación de tamaño r del conjunto $\{1, 2, \dots, n\}$, los vectores (X_1, X_2, \dots, X_r) y $(X_{s_1}, X_{s_2}, \dots, X_{s_r})$ tienen la misma distribución, y por lo tanto, $h(X_1, X_2, \dots, X_r)$ y $h(X_{s_1}, X_{s_2}, \dots, X_{s_r})$ son variables aleatorias idénticamente distribuidas. Por lo tanto, para cada $F \in \mathcal{F}$,

$$\gamma(F) = E_F[h(X_{s_1}, X_{s_2}, \dots, X_{s_r})] = E_F[h(X_1, X_2, \dots, X_r)],$$

de donde se desprende que

$$\begin{aligned} E_F[U_h] &= E_F \left[\frac{1}{\binom{n}{r}} \sum_S h(X_{s_1}, X_{s_2}, \dots, X_{s_r}) \right] \\ &= \frac{1}{\binom{n}{r}} E_F \left[\sum_S h(X_{s_1}, X_{s_2}, \dots, X_{s_r}) \right] \\ &= \frac{1}{\binom{n}{r}} \sum_S E_F [h(X_{s_1}, X_{s_2}, \dots, X_{s_r})] \\ &= \frac{1}{\binom{n}{r}} \sum_S \gamma(F). \end{aligned}$$

En la última sumatoria, el índice S varía sobre la clase de todas las permutaciones de tamaño r del conjunto $\{1, 2, \dots, n\}$, cuyo número es $(n)_r$. Por lo tanto,

$$E_F[U_h] = \frac{1}{(n)_r} [(n)_r \gamma(F)] = \gamma(F),$$

lo cual significa que U_h es un estimador insesgado de $\gamma(F)$.

(ii) Se dará un bosquejo del argumento que conduce al resultado. Primero, observe que $U_h = U_h(X_1, X_2, \dots, X_n)$ no se altera si se intercambia el orden en que aparecen las variables X_i , lo cual implica que

$$U_h \text{ es función de } \mathbf{Y} = (Y_{(1)}, Y_{(2)}, \dots, Y_{(n)}),$$

donde \mathbf{Y} denota la vector de los estadísticos de orden correspondientes a la muestra X_1, X_2, \dots, X_n . Además, debido a que \mathcal{F} contienen a todas las distribuciones continuas, \mathbf{Y} es un estadístico suficiente y completo respecto a \mathcal{F} (Lehmann, 1971). Por lo tanto, U_h es un estimador insesgado de $\gamma(F)$, el cual es una función del estadístico suficiente y completo \mathbf{Y} , y en estas circunstancias, el teorema de Lehmann–Scheffè implica que U_h tiene varianza uniformemente mínima dentro de la clase de estimadores insesgados de $\gamma(F)$ (Lehmann (1971). Mood *et al.*, 1985).
□

Es claro que la distribución exacta de un estadístico U es difícil de determinar, fundamentalmente, debida a que en un contexto no paramétrico las distribuciones que pueden pertenecer a \mathcal{F} poseen características sumamente diversas. Por esta razón, los problemas de prueba de hipótesis y de construcción de intervalos de confianza para una función estimable, se abordan por medio del siguiente resultado asintótico.

Teorema 4.2. [Hoeffding (1948), Serffling (1980), Randles y Wolfe (1981).] Sea X_1, X_2, \dots, X_n una muestra de una distribución perteneciente a la familia \mathcal{F} , y

suponga que $\gamma(\cdot)$ es una función estimable de grado r con núcleo $h(x_1, x_2, \dots, x_r)$. Defina $\zeta = \zeta(F)$ mediante

$$\zeta(F) = E_F[h(X_1, X_2, \dots, X_r)h(X_1, X_{r+1}, \dots, X_{2r-1})] - \gamma^2. \quad (4.4)$$

Con esta notación, la distribución límite de $\sqrt{n}(U_h - \gamma)$ conforme n crece a ∞ es normal con media nula y varianza $r^2\zeta$, esto es,

$$\sqrt{n}(U_h - \gamma) \xrightarrow{\mathcal{D}} \mathcal{N}(0, r^2\zeta).$$

Una demostración de este resultado, la cual se obtiene a través de argumentos de aproximación de U_h por medio de sumas de variables aleatorias independientes e idénticamente distribuidas, puede encontrarse en Randles y Wolfe (1981). La aplicación de este resultado para establecer intervalos de confianza para γ requiere estimar ζ definida en (4.4), y el aspecto relevante es que ζ es una función estimable. Para verificar esta afirmación, note que

$$\begin{aligned} \gamma(F)^2 &= E_F[h(X_1, \dots, X_r)]^2 \\ &= E_F[h(X_1, \dots, X_r)]E_F[h(X_{r+1}, \dots, X_{2r})] \\ &= E_F[h(X_1, \dots, X_r)h(X_{r+1}, \dots, X_{2r})]; \end{aligned}$$

para obtener la primera igualdad se utilizó que (X_1, \dots, X_r) y (X_{r+1}, \dots, X_{2r}) tienen la misma distribución y son independientes, por lo tanto $h(X_1, \dots, X_r)$ y $h(X_{r+1}, \dots, X_{2r})$ también tienen estas propiedades, mientras que la tercera igualdad utilizó la independencia de $h(X_1, \dots, X_r)$ y $h(X_{r+1}, \dots, X_{2r})$ para concluir que el producto de sus esperanzas es la esperanza de su producto. Combinando esta igualdad con (4.4) se obtiene

$$\begin{aligned} \zeta(F) &= E_F[h(X_1, X_2, \dots, X_r)h(X_1, X_{r+1}, \dots, X_{2r-1})] \\ &\quad - E_F[h(X_1, \dots, X_r)h(X_{r+1}, \dots, X_{2r})] \end{aligned}$$

y por lo tanto,

$$\begin{aligned} \zeta(F) &= E_F[h(X_1, \dots, X_r)h(X_1, X_{r+1}, \dots, X_{2r-1}) - h(X_1, \dots, X_r)h(X_{r+1}, \dots, X_{2r})] \\ &= E_F[H(X_1, \dots, X_r, X_{r+1}, \dots, X_{2r-1}, X_{2r})] \end{aligned} \quad (4.5)$$

Esta última igualdad muestra que $\zeta(F)$ es una función estimable, pues se expresa como el valor esperado de una función $H(X_1, X_2, \dots, X_{2r})$. Por lo tanto, ζ puede estimarse mediante el estadístico de tipo U asociado a H :

$$\hat{\zeta} = \frac{1}{(n)_{2r}} \sum_S H(X_{s_1}, X_{s_2}, \dots, X_{s_{2r}}) = U_H. \quad (4.6)$$

Utilizando este estimador, el Teorema 4.2 implica el siguiente corolario.

Corolario 4.1. Dado $\alpha \in (0, 1)$, sea $z_{\alpha/2}$ el percentil superior de orden α de la distribución normal estándar. Con esta notación el intervalo aleatorio

$$\left[\hat{\gamma} - z_{\alpha/2} \frac{r}{\sqrt{n}} \hat{\zeta}, \hat{\gamma} + z_{\alpha/2} \frac{r}{\sqrt{n}} \hat{\zeta} \right] \quad (4.7)$$

es un intervalo de confianza para $\gamma(F)$ con nivel asintótico $1 - \alpha$, esto es, para todo $F \in \mathcal{F}$,

$$\lim_{n \rightarrow \infty} P_F \left[\gamma \in \left[\hat{\gamma} - z_{\alpha/2} r \sqrt{\frac{\hat{\zeta}}{\sqrt{n}}}, \hat{\gamma} + z_{\alpha/2} r \sqrt{\frac{\hat{\zeta}}{\sqrt{n}}} \right] \right] = 1 - \alpha.$$

Cómputo de un Estadístico U

El resultado establecido en el Corolario 4.1 es extremadamente importante. En efecto, sin suponer forma específica alguna para la distribución de la que provienen los datos, la fórmula (4.7) permite construir un intervalo de confianza para $\gamma = \gamma(F)$, el cual, si se desea o se requiere, puede utilizarse para probar hipótesis sobre γ . Sin embargo, hay un aspecto que puede pasar, en principio, desapercibido en la expresión (4.7), y sobre el cual se discutirá a continuación: La carga computacional para evaluar el intervalo de confianza (4.7) es, en general, enorme, aún en el caso de que el tamaño de la muestra sea relativamente ‘pequeño’.

La discusión inicia observando que, para encontrar el intervalo de confianza (4.7), es necesario calcular dos estadísticos de tipo U , a saber, U_h y U_H , los cuales son los estimadores de γ y ζ , respectivamente (vea (4.2) y (4.6)), y como punto

de partida, suponiendo, por ejemplo, que se tienen $n = 100$ datos que constituyen los valores observados de una muestra de 100 variables aleatorias obtenidas de una distribución $F \in \mathcal{F}$. En este caso, si $\gamma(F)$ es una función estimable de grado dos, el cómputo de U_h requiere evaluar $h(X_{s_1}, X_{s_2})$ para cada una de las permutaciones (s_1, s_2) de tamaño dos del conjunto $\{1, 2, 3, \dots, 100\}$. Así, para lograr calcular estas cantidades, se requiere, ante todo, generar

$$(100)_2 = 100(99) = 9900 \quad \text{permutaciones de tamaño dos.}$$

Por otro lado, ζ es una función estimable de grado $2r = 4$, y su evaluación requiere encontrar, explícitamente,

$$(100)_4 = 100(99)(98)(97) = 94109400 \quad \text{permutaciones de tamaño cuatro.}$$

Al reflexionar sobre estas cifras, se pone en evidencia la enorme importancia de disponer de un procedimiento que permita generar todas las permutaciones de tamaño r de un conjunto de enteros. La generación debe ser ‘ordenada’, en el sentido de que cada permutación debe ser generada una sólo vez, sin posibilidad de incurrir en ‘repeticiones accidentales’.

Existe una manera de ‘disminuir’, en cierto sentido, la carga computacional que significa la evaluación de los estadísticos de tipo U involucrados en la construcción del intervalo de confianza (4.7), la cual involucra utilizar un núcleo adecuado en (4.1).

Definición 5.1. Sea $\gamma = \gamma(F)$ una función estimable de grado r . Un núcleo simétrico para γ , es una función $h^*(x_1, x_2, \dots, x_r)$ que satisface las siguientes condiciones:

$$\gamma(F) = E_F[h^*(X_1, X_2, \dots, X_r)], \quad F \in \mathcal{F},$$

y para toda permutación a_1, a_2, \dots, a_r de $\{1, 2, \dots, r\}$,

$$h^*(x_1, x_2, \dots, x_r) = h^*(x_{a_1}, x_{a_2}, \dots, x_{a_r}). \quad (5.1)$$

Dado un núcleo arbitrario h para γ , siempre puede construirse un núcleo simétrico h^* a través de la siguiente fórmula:

$$h^*(x_1, x_2, \dots, x_r) = \frac{1}{r!} \sum_{a_1, a_2, \dots, a_r} h(x_{a_1}, x_{a_2}, \dots, x_{a_r}). \quad (5.2)$$

Cuando $h^*(\cdot)$ es un núcleo simétrico, su valor no se altera si se cambia el orden de las variables x_1, x_2, \dots, x_r en las que se evalúa a h^* . Luego, al construir el estadístico U asociado a h^* , se reducen algunos términos, lo que conduce a una sumatoria con menos sumandos:

$$U_{h^*} = \frac{1}{\binom{n}{r}} \sum_{\{s_1, s_2, \dots, s_r\}} h^*(x_{s_1}, x_{s_2}, \dots, x_{s_r}), \quad (5.3)$$

donde la suma es sobre todos los *subconjuntos* de tamaño r cuyos elementos pertenecen a la ‘población’ $\{1, 2, \dots, n\}$; note que sin importar cuál es el núcleo de γ que se utilice, se obtendrá siempre el mismo estimador, a saber, el estimador insesgado de varianza uniformemente mínima para γ . Utilizando (5.3), la determinación del estadístico de tipo U que estima a γ , requiere de determinar cada subconjunto $\{s_1, s_2, \dots, s_r\}$ de tamaño r de $\{1, 2, \dots, n\}$, y posteriormente evaluar cada término $h^*(x_{s_1}, x_{s_2}, \dots, x_{s_r})$. Por ejemplo, en el caso discutido anteriormente, en el que γ tiene grado dos y se dispone de 100 observaciones, el cálculo del estimador $\hat{\gamma}$ por medio de un estadístico U asociado a un núcleo simétrico requiere encontrar

$$\binom{100}{2} = 4950 \quad \text{subconjuntos de tamaño dos,}$$

mientras que el cómputo del estimador $\hat{\zeta}$ por medio de un núcleo simétrico implica la determinación de

$$\binom{100}{4} = 3921225 \quad \text{subconjuntos de tamaño cuatro.}$$

Estas cifras continúan siendo enormes, y ponen en evidencia la importancia de listar (enumerar) los conjuntos de r elementos cuyas componentes pertenecen a $\{1, 2, \dots, n\}$.

Conclusión

La discusión que se ha llevado a cabo en este capítulo, muestra la importancia que el problema de listar (enumerar) familias de permutaciones o subconjuntos tiene para la aplicación efectiva de métodos estadísticos no paramétricos, y proporcionan la motivación para emprender el desarrollo de este trabajo.

El trayecto que se seguirá a partir de ahora puede describirse, a grandes rasgos, como sigue:

- Primero se analizará la relación entre los problemas de listar y *ordenar* un conjunto, trabajo que permitirá identificar el aspecto esencial de los problemas de enumeración que son el objeto de estudio de esta tesis;
- Utilizando la teoría desarrollada se diseñara un procedimiento para enumerar todas las permutaciones de tamaño N de la población

$$\{1, 2, \dots, N\}, \quad (6.1)$$

y

- La experiencia acumulada permitirá construir un algoritmo para listar todos los subconjuntos de tamaño $r \leq N$ de la población (6.1)

Suponga ahora que se desea enumerar todas las permutaciones de tamaño $r \leq N$ del conjunto $\{1, 2, \dots, N\}$. Esto se puede lograr combinando los dos procedimientos descritos anteriormente de manera iterativa:

Se genera un subconjunto S de tamaño r

Se generan todas la permutaciones de tamaño r de S , el cual tiene r elementos.

Al repetir este proceso para cada subconjunto S de la población (6.1), se obtienen todas las permutaciones de tamaño r de la población $\{1, 2, \dots, N\}$.

Capítulo 3

Teoría General Sobre Listado de un Conjunto

El objetivo de este capítulo es presentar un enfoque general al problema de producir una lista de todos los miembros de un conjunto. La intención principal de la exposición es mostrar la equivalencia entre el problema de enumeración considerado, y la construcción de una relación de orden en el conjunto de interés. A partir de este resultado se formula un algoritmo general de listado, el cual se aplicará en el siguiente capítulo a los problemas de enumeración de permutaciones y familias de subconjuntos, los cuales son de gran importancia para el cálculo de estimadores construidos utilizando métodos no paramétricos.

Introducción

En este capítulo se analiza el siguiente problema general de enumeración o listado: *Dado un conjunto finito \mathcal{A} , producir una lista en la que cada elemento de \mathcal{A} aparezca sólo una vez.*

A primera vista, este parece ser un problema simple; después de todo, una manera de especificar un conjunto es precisamente proporcionando una lista de todos sus elementos. Por ejemplo, al escribir $\mathcal{A} = \{3, 6, 9, 12, 15\}$, se están enumerando los miembros de \mathcal{A} . Sin embargo, existe otra forma de determinar un conjunto, la cual se denomina *método de comprensión o regla*, de acuerdo a la cual los elementos del conjunto se distinguen mediante una propiedad que los caracteriza; vea por ejemplo, Suppes (1971), o Dolciani *et al.*, (1986). Por ejemplo, si \mathcal{B} es el conjunto de todos los enteros positivos menores a 17 que son múltiplos de tres, se puede establecer con toda seguridad que (i) $2 \notin \mathcal{B}$, pues a pesar de que dos es un entero positivo y menor a 17, dos no es un múltiplo de tres, (ii) $21 \notin \mathcal{B}$,

pues aunque 21 si es un entero positivo múltiplo de tres, 21 no es menor a 17, y (c) $9 \in \mathcal{B}$, pues nueve es un entero positivo que es tanto múltiplo de tres como menor a 17. El aspecto fundamental en esta discusión es que la propiedad que se estipula para los miembros de \mathcal{B} permite decidir, sin ambigüedad alguna, si un objeto determinado pertenece o no a dicho conjunto.

Aún cuando originalmente se especifique un conjunto mediante una caracterización de sus miembros, frecuentemente es necesario disponer de una lista de sus elementos, esto es, debe utilizarse la propiedad que distingue a los objetos que constituyen \mathcal{B} para producir una lista de todos sus miembros—equivalentemente, para enumerar todos los elementos de \mathcal{B} . En el caso especificado en el párrafo precedente, no es difícil mostrar que $\mathcal{B} = \{3, 6, 9, 12, 15\}$, es decir \mathcal{B} coincide con \mathcal{A} , de manera que en estas circunstancias es sencillo pasar de la caracterización del conjunto \mathcal{B} por el método de comprensión, a la descripción de \mathcal{B} mediante el listado o enumeración de sus miembros. Sin embargo, en general la situación es más compleja. Considere, por ejemplo, el conjunto \mathcal{C} que consiste de todos los dígitos (enteros entre cero y nueve, inclusive), que aparecen por lo menos cuatro ocasiones dentro de las primeras veinte cifras de la parte fraccionaria en la expansión decimal del número π . En este caso, \mathcal{C} consta de a lo más cinco miembros; producir una lista de \mathcal{C} , sin embargo, requiere de que se tenga la expansión de π hasta 20 cifras significativa, y es necesario disponer de algoritmos especiales para este fin. Este ejemplo muestra que, a pesar de que se sepa que un conjunto tiene pocos miembros—a lo más cinco en el caso de \mathcal{C} —generar una lista de sus elementos no es, en todos los casos, un problema simple; más bien, con frecuencia reclama un análisis esmerado y, tal vez, complejo. En el caso del conjunto \mathcal{C} , puede mostrarse que consiste de un único elemento, a saber, cinco (Papadopoulos (1997)).

El propósito general de este capítulo es *diseñar un procedimiento para enumerar los miembros de un conjunto finito arbitrario*, y para alcanzar este objetivo, la exposición ha sido organizada de la siguiente manera: En la Sección 2

se discute la relevancia del problema de listado en el campo de la estadística no paramétrica, tratando de ubicar el problema general abordado desde una óptica que muestre su relevancia. Posteriormente, en la Sección 3 se discute la relación entre el problema general de listado, y la construcción de un orden total en el conjunto de interés, y se introducen las nociones de elemento máximo y mínimo en un conjunto ordenado. La Sección 4 trata sobre la idea de sucesor de un elemento, la cual es una noción central para formular, en la Sección 5, el resultado principal de este capítulo, a saber, el algoritmo general de listado. Después de establecer este procedimiento, en la Sección 6 se introduce una relación de orden total entre vectores de \mathbb{R}^p , a saber, el denominado orden lexicográfico (Munkres, 1975), y la exposición concluye con algunos comentarios breves en la Sección 7.

El Ángulo Estadístico

Antes de emprender la tarea de diseñar un método para enumerar los miembros de un conjunto finito arbitrario, es conveniente enfatizar, una vez más, la razón por la cual este problema es de gran importancia en el análisis estadístico. El punto de partida, es la observación de que enumerar (listar) los elementos de un conjunto finito puede ser un problema complejo por dos razones:

- (i) Es difícil determinar *explícitamente* los elementos del conjunto; por ejemplo, este es el caso para el conjunto \mathcal{C} considerado en la Sección 1.
- (ii) El conjunto de interés tiene un número grande de elementos.

En el análisis estadístico, particularmente al emplear métodos de estimación no paramétrica, la dificultad para enumerar los miembros de un conjunto de interés se origina, frecuentemente, por la segunda de las causas mencionadas. Por ejemplo, considere las siguientes tres clases de objetos construidos a partir de una población \mathcal{P} que consta de N objetos, los cuales, sin pérdida de generalidad alguna, se supone que son los primeros N enteros positivos, i.e.,

$$\mathcal{P} = \{1, 2, \dots, N\}. \tag{2.1}$$

- La clase $\mathcal{P}(N, N)$, la cual consiste de todas las permutaciones de tamaño N de la población \mathcal{P} :

$$\mathcal{P}(N, N) = \{(a_1, a_2, \dots, a_n) \mid a_i \in \mathcal{P}, i = 1, 2, \dots, n, a_i \neq a_j, 1 \leq i \neq j \leq N\}. \quad (2.2)$$

- $\mathcal{P}(r, N)$, la familia de todas las permutaciones de tamaño r de la población \mathcal{P} , donde $r < N$, esto es,

$$\mathcal{P}(r, N) = \{(a_1, a_2, \dots, a_n) \mid a_i \in \mathcal{P}, i = 1, 2, \dots, r, a_i \neq a_j, 1 \leq i \neq j \leq r\} \quad (2.3)$$

y, finalmente,

- Dado un número $r < N$, sea $\mathcal{S}(r, N)$, la clase de todos los subconjuntos de tamaño r de la población \mathcal{P} :

$$\mathcal{S}(r, N) = \{S \mid S \subset \mathcal{P}, \text{ y } S \text{ consiste de } r \text{ miembros}\}. \quad (2.4)$$

Cada una de estas clases es de gran importancia; las dos últimas tienen que ver directamente con la evaluación de estimadores del tipo U considerados en el capítulo anterior, mientras que la familia $\mathcal{P}(N, N)$ es de importancia al realizar experimentos de simulación computacional, por ejemplo, para comparar la distribución exacta de un estadístico ‘de distribución libre’, con su distribución asintótica (Meddis, 1984). En los tres casos precedentes, la dificultad para listar los miembros de cada clase proviene de que el número de sus miembros crece muy rápidamente a medida que se incrementa el tamaño de la población. Por ejemplo, aunque para $N = 4$ y $r = 2$, $\mathcal{S}(r, N) = \mathcal{S}(2, 4)$ tiene seis miembros que podrían enumerarse directamente, cuando $N = 50$ y $r = 2$, $\mathcal{S}(2, 50)$ consta de 1225 miembros. La motivación detrás de este capítulo es desarrollar un método que permita listar conjuntos con un número grande de elementos, particularmente, los tres recientemente mencionados. Sin embargo, para alcanzar este propósito, es conveniente analizar el problema de listado desde una perspectiva general (abstracta)

que permita descubrir los aspectos esenciales del problema de enumeración, para posteriormente aplicar los resultados a los casos relevantes en el análisis estadístico.

Relaciones de Orden

La observación clave para abordar el problema de listar un conjunto finito \mathcal{A} , es que cuando se produce una lista de sus miembros, de inmediato se induce un 'orden' en el conjunto, donde los elementos que se escriben primero son 'menores' que aquellos que aparecen después. Para aclarar esta idea, considere el conjunto

$$\mathcal{A} = \{A, *, 0, !\}. \quad (3.1)$$

Suponga ahora que se pide producir una lista de los miembros de \mathcal{A} . Ante este requerimiento, se podría responder con

$$!, A, *, 0 \quad (3.2)$$

En este caso, si se convienen en que un elemento es menor que cualquiera de los que están a su derecha en la lista, se tendría que

$$\begin{aligned} '!' \text{ es menor que } 'A', \quad '!' \text{ es menor que } '*', \quad \text{y} \quad '!' \text{ es menor que } '0' \\ 'A' \text{ es menor que } '*', \quad 'A' \text{ es menor que } '0', \quad \text{y} \quad '*' \text{ es menor que } '0' \end{aligned} \quad (3.3)$$

Esta 'relación de orden' fue inducida por el listado (3.2), y es claro que al formar otra lista de los miembros de \mathcal{A} , la relación de orden se alterará. Por ejemplo, si en vez de (3.2) se enumeran los miembros de \mathcal{A} como

$$A, !, *, 0$$

entonces la primera afirmación en (3.3) debe cambiarse a ' A ' es menor que '!'. La discusión puede extenderse a cualquier conjunto, en el sentido de que al disponer de una lista de sus miembros, puede definirse una relación de orden. El propósito de esta sección es establecer la proposición inversa: Si se dispone de una relación de

orden adecuada en un conjunto finito \mathcal{A} , entonces, dicha relación puede utilizarse para generar un listado de \mathcal{A} , resultado se demostrará posteriormente en el Teorema 3.1. Por el momento, es oportuno definir, de manera precisa, cual es la idea de relación de orden que se utilizará.

Definición 3.1. [Munkres (1975), Suppes (1971).] Sea \mathcal{A} un conjunto finito, y considere una relación, denotada mediante ' \prec ' definida en el conjunto. Esta relación es un *orden completo* en \mathcal{A} si satisface los siguientes requerimientos:

(i) [Transitividad.] Para $a, b, c \in \mathcal{A}$,

$$a \prec b \quad \text{y} \quad b \prec c \quad \implies \quad a \prec c.$$

(ii) [Completez.] Para todos los miembros $a, b \in \mathcal{A}$, se tiene que exactamente una de las siguientes alternativas es válida.

$$a \prec b, \quad a = b, \quad b \prec a.$$

En esta definición, la condición (i) es bastante natural desde un punto de vista intuitivo, mientras que el requerimiento (ii) impone la condición de que dos elementos distintos de \mathcal{A} sean *comparables*, i.e., si $a \neq b$, debe tenerse que $a \prec b$, o bien $b \prec a$. Es costumbre leer la expresión $a \prec b$ como ' a precede a b ', en lugar de ' a es menor a b '; el propósito es no confundir la relación de orden recientemente introducida, con el orden usual en los números reales cuando $\mathcal{A} \subset \mathbb{R}$. Por otro lado, el hecho de que a preceda a b , esto es, $a \prec b$, también se escribe como $b \succ a$, esto es,

$$a \prec b \iff b \succ a. \tag{3.4}$$

La expresión $b \succ a$ se lee como ' b sucede a a ', o ' b es posterior a a '; similarmente ' $a \prec b$ ' con frecuencia se expresa verbalmente como ' a es anterior a b '.

Ejemplo 3.1. Sea $\mathcal{A} = \mathcal{A} = \{A, *, 0, !\}$, el conjunto considerado al inicio de esta sección. La relación (3.3) puede expresarse, con la notación recientemente

introducida, como

$$! \prec A, \quad ! \prec *, \quad ! \prec 0, \quad A \prec *, \quad A \prec 0, \quad \text{y} \quad * \prec 0. \quad (3.5)$$

Con esta definición de la relación de precedencia, no es difícil verificar que las propiedades de transitividad y completez se satisfacen. Por lo tanto (3.5) determina un orden completo en \mathcal{A} . \square

Dado un conjunto finito \mathcal{A} dotado con un orden completo ' \prec ', existen dos elementos especiales, a saber, el elemento mínimo y el miembro máximo.

Definición 3.2. Sea \mathcal{A} un conjunto finito dotado con un orden completo ' \prec '. En este caso,

(i) $\mathbf{m} \in \mathcal{A}$ es un elemento mínimo si para todo $\mathbf{a} \in \mathcal{A}$ con $\mathbf{a} \neq \mathbf{m}$, se tiene que $\mathbf{m} \prec \mathbf{a}$.

Similarmente,

(ii) $\mathbf{M} \in \mathcal{A}$ es un elemento máximo si siempre que $\mathbf{a} \in \mathcal{A}$ sea tal que $\mathbf{a} \neq \mathbf{M}$, entonces $\mathbf{a} \prec \mathbf{M}$.

Ejemplo 3.2. En el Ejemplo 3.1, ' $!$ ' es un mínimo del conjunto \mathcal{A} . En efecto, si $\mathbf{a} \in \mathcal{A}$ es tal que $\mathbf{a} \neq !$, entonces \mathbf{a} es alguno de los miembros A , $*$, o 0 , y en este caso, las primeras tres 'desigualdades' en (3.5) implican que $! \prec \mathbf{a}$. Por otro lado, 0 es un máximo de \mathcal{A} , pues si $\mathbf{a} \in \mathcal{A}$ es diferente de 0 , entonces \mathbf{a} es alguno de los miembros, $!$, A , o $*$, y en este caso, las dos últimas 'desigualdades en (3.5) conjuntamente con la tercera, muestran que $\mathbf{a} \prec 0$. \square

El siguiente resultado muestra que cualquier conjunto finito con un orden completo posee exactamente un elemento mínimo y un máximo.

Teorema 3.1. Sea \mathcal{A} un conjunto finito y no vacío dotado con un orden completo. En este caso,

(i) \mathcal{A} posee un elemento mínimo, Más aún,

(ii) \mathcal{A} tiene exactamente un elemento mínimo.

Similarmente,

(iii) \mathcal{A} posee un elemento Máximo, y

(iv) El elemento máximo de \mathcal{A} es único.

Demostración. (i) El argumento es por inducción en el número de elementos de \mathcal{A} , el cual se denota por n . Primero observe que si \mathcal{A} tiene $n = 1$ elementos, entonces la afirmación es clara. Suponga que si un conjunto dotado con un orden completo tiene k miembros, entonces tiene un elemento mínimo, y sea \mathcal{A} un conjunto de $n = k + 1$ elementos. En estas circunstancias, se verificará que \mathcal{A} posee un elemento mínimo. Con esta finalidad, seleccione un elemento cualquiera, $\mathbf{a}^* \in \mathcal{A}$, y defina conjunto $\tilde{\mathcal{A}}$ mediante

$$\tilde{\mathcal{A}} = \mathcal{A} \setminus \{\mathbf{a}^*\}. \quad (3.6)$$

En este caso, $\tilde{\mathcal{A}}$ es un conjunto con k elementos, y por lo tanto tiene un miembro mínimo, por la hipótesis de inducción. Sea $\tilde{\mathbf{m}} \in \tilde{\mathcal{A}}$ un miembro mínimo de $\tilde{\mathcal{A}}$, i.e.,

$$\mathbf{a} \in \tilde{\mathcal{A}} \quad \text{y} \quad \mathbf{a} \neq \tilde{\mathbf{m}} \implies \tilde{\mathbf{m}} \prec \mathbf{a}. \quad (3.7)$$

Para demostrar que el conjunto original tiene un mínimo, primero observe que $\tilde{\mathbf{m}} \in \tilde{\mathcal{A}}$, y por lo tanto $\tilde{\mathbf{m}} \neq \mathbf{a}^*$ (vea (3.6)). Utilizando que dos elementos distintos de \mathcal{A} son comparables, por la Definición 3.1, esto implica que alguna de las siguientes dos alternativas ocurre:

$$\mathbf{a}^* \prec \tilde{\mathbf{m}}, \quad \text{o} \quad \tilde{\mathbf{m}} \prec \mathbf{a}^*.$$

Caso 1: $\mathbf{a}^* \prec \tilde{\mathbf{m}}$. En esta situación, se verificará que \mathbf{a}^* es un elemento mínimo de \mathcal{A} , i.e.,

$$\mathbf{a}^* \prec \mathbf{a}, \quad \text{si } \mathbf{a} \in \mathcal{A} \text{ y } \mathbf{a} \neq \mathbf{a}^*. \quad (3.8)$$

En efecto, si $\mathbf{a} \in \mathcal{A}$ es diferente de \mathbf{a}^* , entonces $\mathbf{a} \in \tilde{\mathcal{A}}$ (vea (3.6)). Por lo tanto, $\mathbf{a} = \tilde{\mathbf{m}}$, o bien $\tilde{\mathbf{m}} \prec \mathbf{a}$; bajo la primera alternativa se tiene que $\mathbf{a}^* \prec \mathbf{a}$, pues se supone que $\mathbf{a}^* \prec \tilde{\mathbf{m}}$ y $\tilde{\mathbf{m}} = \mathbf{a}$, mientras que si la segunda posibilidad ocurre, se

obtiene que $\mathbf{a}^* \prec \tilde{\mathbf{m}}$ y $\tilde{\mathbf{m}} \prec \mathbf{a}$, de tal forma que usando la transitividad de la relación de orden, se obtiene $\mathbf{a}^* \prec \mathbf{a}$. Este argumento muestra que la implicación (3.8) es válida, y por lo tanto, \mathbf{a}^* es un elemento mínimo de \mathcal{A} .

Caso 2: $\tilde{\mathbf{m}} \prec \mathbf{a}^*$. En estas circunstancias, $\tilde{\mathbf{m}}$ es un miembro mínimo de \mathcal{A} . En efecto, seleccione $\mathbf{a} \in \mathcal{A}$ tal que $\mathbf{a} \neq \tilde{\mathbf{m}}$ y observe que existen dos alternativas posibles:

(a) $\mathbf{a} = \mathbf{a}^*$, circunstancia en la que se tiene

$$\tilde{\mathbf{m}} \prec \mathbf{a}, \quad (3.9)$$

pues $\mathbf{a} = \mathbf{a}^*$, y

(b) si $\mathbf{a} \neq \mathbf{a}^*$, entonces $\mathbf{a} \in \tilde{\mathcal{A}}$ (vea (3.6)), y como $\tilde{\mathbf{m}}$ es un mínimo de $\tilde{\mathcal{A}}$, se tiene que (3.9) también ocurre cuando $\mathbf{a} \neq \tilde{\mathbf{m}}$.

En resumen: Suponiendo que un conjunto con k miembros siempre tiene un elemento mínimo, se ha mostrado que cualquier conjunto con $k+1$ elementos también posee un mínimo, completando el argumento que establece la parte (i) por el método de inducción.

(ii) Suponiendo que \mathbf{m} y \mathbf{m}_1 son dos elementos mínimos de \mathcal{A} , se demostrará que $\mathbf{m} = \mathbf{m}_1$. Primero, observe que

$$\mathbf{a} \in \mathcal{A} \quad \text{y} \quad \mathbf{a} \neq \mathbf{m} \implies \mathbf{m} \prec \mathbf{a}, \quad (3.10)$$

(vea la Definición 3.2), y similarmente,

$$\mathbf{a} \in \mathcal{A} \quad \text{y} \quad \mathbf{a} \neq \mathbf{m}_1 \implies \mathbf{m}_1 \prec \mathbf{a}, \quad (3.11)$$

Suponga ahora que $\mathbf{m} \neq \mathbf{m}_1$. En este caso, apartir de (3.10) con $\mathbf{a} = \mathbf{m}_1$ se desprende que

$$\mathbf{m} \prec \mathbf{m}_1$$

mientras que utilizando (3.11) con $\mathbf{a} = \mathbf{m}$, se obtiene que

$$\mathbf{m}_1 \prec \mathbf{m}.$$

Sin embargo, las dos últimas ‘desigualdades desplegadas ocurren de forma simultánea, contradiciendo la condición de completez en la Definición 3.1. Esta contradicción muestra que el supuesto del cual se origina, i.e., $\mathbf{m} \neq \mathbf{m}_1$, es falso, de manera que $\mathbf{m} = \mathbf{m}_1$, completando la demostración de la parte (ii). El resto del teorema puede demostrarse mediante argumentos análogos a los utilizados para establecer las partes (i) y (ii). \square

El Sucesor de un Elemento

En esta sección se presenta una importante idea para la formulación de un algoritmo general de listado de los miembros de un conjunto. En forma simple, esta noción es la de ‘elemento siguiente’, o ‘sucesor inmediato’ a un miembro de un conjunto ordenado.

Definición 4.1. Sea \mathcal{A} un conjunto finito dotado de un orden completo. Dado $\mathbf{a} \in \mathcal{A}$, sea $\mathbf{b} \in \mathcal{A}$ un elemento que satisface las siguientes condiciones:

$$\mathbf{a} \prec \mathbf{b}, \tag{4.1}$$

y

$$\text{Si } \mathbf{c} \in \mathcal{A} \text{ satisface } \mathbf{a} \prec \mathbf{c}, \text{ entonces } \mathbf{b} \preceq \mathbf{c}. \tag{4.2}$$

En este caso, \mathbf{b} se denomina *un sucesor de \mathbf{a}* ; note que en (4.2), $\mathbf{a} \preceq \mathbf{b}$ es una forma abreviada de la expresión $\mathbf{a} \prec \mathbf{b}$ o $\mathbf{a} = \mathbf{b}$:

$$\mathbf{a} \preceq \mathbf{b} \iff \mathbf{a} \prec \mathbf{b} \text{ o } \mathbf{a} = \mathbf{b}.$$

Para entender esta idea, suponga que los elementos de \mathcal{A} se colocan en un línea horizontal, de manera que si $\mathbf{x}, \mathbf{y} \in \mathcal{A}$ son tales que $\mathbf{x} \prec \mathbf{y}$, entonces \mathbf{x} se encuentra a la izquierda de \mathbf{y} . Suponga ahora que se recorren los miembros de \mathcal{A} de izquierda a derecha; en este caso, un sucesor de \mathbf{a} es un elemento $\mathbf{b} \in \mathcal{A}$ que se ubica a la derecha de \mathcal{A} con la propiedad de que al moverse de \mathbf{a} hacia \mathbf{b} no

se encontrará elemento alguno de \mathcal{A} 'en el camino'. El siguiente teorema muestra que, con la excepción del máximo, todos los miembros de \mathcal{A} tienen un sucesor, y que éste es único.

Teorema 4.1. Sea \mathcal{A} un conjunto finito dotado con un orden completo, y suponga que $\mathbf{a} \in \mathcal{A}$ es diferente del máximo elemento de \mathcal{A} . En este caso, \mathbf{a} posee un sucesor, el cual es único.

Demostración. Considere el conjunto

$$\mathcal{S}(\mathbf{a}) = \{\mathbf{c} \in \mathcal{A} \mid \mathbf{a} \prec \mathbf{c}\}, \quad (4.1)$$

esto es, $\mathcal{S}(\mathbf{a})$ contiene a todos los miembros de \mathcal{A} que suceden al elemento \mathbf{a} . Debido a que $\mathbf{a} \neq \mathbf{M}$, se tiene que $\mathbf{a} \prec \mathbf{M}$, de tal forma que $\mathbf{M} \in \mathcal{S}(\mathbf{a})$. Por lo tanto, $\mathcal{S}(\mathbf{a})$ es un conjunto no vacío, el cual posee un único elemento mínimo, por el Teorema 3.1, el cual se denotará mediante \mathbf{b} . Puesto que $\mathbf{b} \in \mathcal{S}(\mathbf{a})$, se tiene que

$$\mathbf{a} \prec \mathbf{b};$$

vea (4.1). Ahora considere un elemento arbitrario $\mathbf{c} \in \mathcal{A}$ tal que $\mathbf{a} \prec \mathbf{c}$. En este caso, se tiene que $\mathbf{c} \in \mathcal{S}(\mathbf{a})$, de manera que $\mathbf{b} \preceq \mathbf{c}$, pues \mathbf{b} es el mínimo de los elementos de $\mathcal{S}(\mathbf{a})$. Esta discusión puede resumirse como sigue: Se ha demostrado que $\mathbf{a} \prec \mathbf{b}$, y que $\mathbf{b} \preceq \mathbf{c}$ para todo \mathbf{c} que satisface $\mathbf{a} \prec \mathbf{c}$. Por lo tanto, \mathbf{b} es un sucesor de \mathbf{a} . Para concluir, suponga que \mathbf{b} y \mathbf{b}_1 son dos sucesores de \mathbf{a} . En este caso,

$$\mathbf{a} \prec \mathbf{b} \quad \text{y} \quad \mathbf{a} \prec \mathbf{c} \implies \mathbf{b} \preceq \mathbf{c}. \quad (4.2)$$

Similarmente,

$$\mathbf{a} \prec \mathbf{b}_1 \quad \text{y} \quad \mathbf{a} \prec \mathbf{c} \implies \mathbf{b}_1 \preceq \mathbf{c}. \quad (4.3)$$

Usando la implicación en (4.2) con $\mathbf{c} = \mathbf{b}_1$, se desprende que

$$\mathbf{b} \preceq \mathbf{b}_1,$$

mientras que a partir de la implicación en (4.3) con $\mathbf{c} = \mathbf{b}$, se obtiene

$$\mathbf{b}_1 \preceq \mathbf{b},$$

y por lo tanto, si $\mathbf{b} \neq \mathbf{b}_1$, la dos últimas relaciones despelegadas implican que $\mathbf{b} \prec \mathbf{b}_1$ y $\mathbf{b}_1 \prec \mathbf{b}$ ocurren simultáneamente, lo cual se opone a la propiedad de completez en la Definición de orden total. Por lo tanto, $\mathbf{b} = \mathbf{b}_1$, estableciendo la unicidad del sucesor de \mathcal{A} . \square

Definición 4.1. Suponga que \mathcal{A} es un conjunto finito que consiste de más de un elemento. Defina la función

$$\text{Suc}: \mathcal{A} \setminus \{\mathbf{M}\} \rightarrow \mathcal{A}$$

mediante

$$\text{Suc}(\mathbf{a}) = \text{sucesor de } \mathbf{a}, \quad \mathbf{a} \in \mathcal{A} \setminus \{\mathbf{M}\}.$$

Ejemplo 4.1 Considere el conjunto \mathcal{A} introducido en el Ejemplo 3.1, esto es, $\mathcal{A} = \mathcal{A} = \{A, *, 0, !\}$, en el cual la relación de orden completo está dada por

$$! \prec A, \quad ! \prec *, \quad ! \prec 0, \quad A \prec *, \quad A \prec 0, \quad \text{y} \quad * \prec 0. \quad (4.4)$$

Como se mencionó anteriormente, $\mathbf{M} = 0$ es el máximo de \mathcal{A} . El sucesor de cada elemento de \mathcal{A} diferente de 0 está dado, de acuerdo a (4.4), por

$$\text{Suc}(!) = A, \quad \text{Suc}(A) = *, \quad \text{Suc}(*) = 0. \quad (4.5)$$

\square

Como consecuencia del Teorema 4.1 y de la definición de la función **Suc**, se desprende el siguiente Corolario.

Corolario 4.1. Dado un conjunto finito con más de un elemento, la función **Suc** es uno a uno. Más aún, **Suc** es una biyección entre $\mathcal{A} \setminus \{\mathbf{M}\}$ y $\mathcal{A} \setminus \{\mathbf{m}\}$.

Demostración. Primero se verificará que **Suc** es un función uno a uno, esto es, que la siguiente afirmación es válida:

$$\mathbf{Suc}(\mathbf{a}) = \mathbf{Suc}(\tilde{\mathbf{a}}) \implies \mathbf{a} = \tilde{\mathbf{a}}. \quad (4.6)$$

Para verificar esta afirmación se utilizará el método de contradicción. Con este fin, denote mediante **b** al valor común de **Suc**(**a**) y **Suc**($\tilde{\mathbf{a}}$), esto es,

$$\mathbf{b} = \mathbf{Suc}(\mathbf{a}) = \mathbf{Suc}(\tilde{\mathbf{a}}),$$

y suponga que $\mathbf{a} \neq \tilde{\mathbf{a}}$. En este caso, alguna de las dos alternativas ocurre:

$$\mathbf{a} \prec \tilde{\mathbf{a}}, \quad \text{o} \quad \tilde{\mathbf{a}} \prec \mathbf{a},$$

y sin pérdida de generalidad alguna, se supondrá que la primera de estas posibilidades es válida. En este caso, se tiene que

$$\mathbf{a} \prec \tilde{\mathbf{a}} \prec \mathbf{Suc}(\tilde{\mathbf{a}}) = \mathbf{b},$$

donde la segunda relación de precedencia se debe a la definición de sucesor de $\tilde{\mathbf{a}}$. Sin embargo, **b** es también el sucesor de **a**, de manera que la anterior relación desplegada implica que

$$\mathbf{a} \prec \tilde{\mathbf{a}} \prec \mathbf{Suc}(\mathbf{a}) = \mathbf{b},$$

y entonces $\tilde{\mathbf{a}}$ sucede a **a**, pero es anterior a **Suc**(**a**) = **b**, lo cual se opone al hecho de que **b** es el sucesor de **a**. Esto muestra que el supuesto $\mathbf{a} \neq \tilde{\mathbf{a}}$ conduce a una contradicción, y por lo tanto la implicación (4.6) es válida. Para concluir, recuerde que el dominio de la función **Suc** es $\mathcal{A} \setminus \{\mathbf{M}\}$. Además, **m**, el elemento mínimo de \mathcal{A} no puede ser el sucesor de ningún miembro de \mathcal{A} pues, por la Definición 3.2(i), **m** no sucede a ningún elemento de \mathcal{A} . Por lo tanto, **Suc** toma valores en $\mathcal{A} \setminus \{\mathbf{m}\}$, es decir,

Suc transforma cada miembro de $\mathcal{A} \setminus \{\mathbf{M}\}$ en un elemento de $\mathcal{A} \setminus \{\mathbf{m}\}$;

Puesto que ambos conjuntos en este enunciado tienen el mismo número de elementos y la función **Suc** es uno a uno, se desprende que **Suc** transforma $\mathcal{A} \setminus \{\mathbf{M}\}$ sobre $\mathcal{A} \setminus \{\mathbf{m}\}$, esto es, **Suc** es una biyección entre ambos conjuntos. \square

Ejemplo 4.2 De nueva cuenta, sea $\mathcal{A} = \mathcal{A} = \{A, *, 0, !\}$ dotado con el orden descrito en (4.4). Como ya se ha determinado con anterioridad, en este caso, $\mathbf{m} = !$ es el elemento mínimo de \mathcal{A} , mientras que $\mathbf{M} = 0$ es el miembro máximo. En este caso,

$$\mathcal{A} \setminus \{\mathbf{M}\} = \{A, *, !\}, \quad \text{y} \quad \mathcal{A} \setminus \{\mathbf{m}\} = \{A, *, 0\}.$$

Note que de acuerdo a (4.5),

$$\mathbf{Suc}(!) = A$$

$$\mathbf{Suc}(A) = *$$

$$\mathbf{Suc}(*) = 0,$$

de manera que **Suc** establece una biyección entre $\{A, *, !\}$ y $\{A, *, 0\}$, en concordancia con el Corolario 4.1. \square

El Procedimiento General

Después de la teoría desarrollada en las secciones precedentes, ha llegado el momento de presentar el resultado central de este capítulo, a saber, un procedimiento que permite enumerar, o *listar* todos los elementos de un conjunto finito una sólo vez. La premisa fundamental es que se ha definido un orden completo en el conjunto de interés, y la idea detrás del algoritmo es la siguiente: Inicie la lista con el elemento mínimo del conjunto, y vaya agregando el sucesor del elemento que se ha anotado más recientemente, deteniendo el procedimiento cuando el último elemento incorporado no tenga sucesor, lo cual puede ocurrir sólo cuando dicho término es el máximo de los miembros del conjunto (vea el Teorema (4.1)).

Algoritmo General de Listado.

Datos: (i) Un conjunto finito \mathcal{A} con dos o más elementos, dotado con un orden completo.

(ii) Se conoce el elemento mínimo de \mathcal{A} , denotado por \mathbf{m} , así como la función **Suc**, esto es, se puede determinar el sucesor de cualquier elemento que no sea el máximo, y además se puede detectar si un elemento dado no tiene sucesor, de tal forma que dicho elemento es el máximo de \mathcal{A}

Fase 1: [Inicialización.] Defina $k = 1$, $\mathbf{a}_1 = \mathbf{m}$, el elemento mínimo de \mathcal{A}

Fase 2: [Fase de Prueba.] Suponga que se tiene la lista (posiblemente) parcial de elementos de \mathcal{A} :

$$\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k.$$

(a) Si \mathbf{a}_k tiene sucesor, vaya a la Fase 3.

(b) Si \mathbf{a}_k **no** tiene sucesor, vaya a la Fase 4.

Fase 3: [Fase de Iteración.] (a) Incremente k en una unidad y defina \mathbf{a}_{k+1} como el sucesor de \mathbf{a}_k , esto es,

$$k \leftarrow k + 1, \quad \mathbf{a}_{k+1} = \mathbf{Suc}(\mathbf{a}_k).$$

(b) Retorne a la Fase 2.

Fase 4: [Fin del Algoritmo.] La lista completa de los miembros de \mathcal{A} es

$$\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k.$$

□

Antes de demostrar la validez del algoritmo general de listado, se ilustrará su aplicación en el caso que ha sido utilizado anteriormente para ilustrar las ideas introducidas en este capítulo

Ejemplo 5.1. Como antes, sea $\mathcal{A} = \{A, *, 0, !\}$, dotado con la relación de orden (4.4). Como se verificó en el Ejemplo 3.2, el elemento mínimo de \mathcal{A} es $\mathbf{m} = !$ y la función **Suc** está determinada en (4.5); vea el Ejemplo 4.5. La aplicación del algoritmo de listado es como sigue:

Fase 1: $k = 1$, $\mathbf{a}_1 = !$, pues el elemento mínimo de \mathcal{A} es ‘!’

Fase 2: En este momento se tiene la lista parcial

!

Puesto que $\mathbf{a}_1 = !$ tiene sucesor, la ruta es hacia la Fase 3.

Fase 3: (a) Se incrementa k en 1 y se define $\mathbf{a}_2 = \mathbf{Suc}(!) = A$. Después de hacer estos cambios, se tiene

$$k = 2, \quad \text{y} \quad \mathbf{a}_1 = !, \quad \mathbf{a}_2 = A.$$

Fase 2: En este momento $k = 2$ y la lista parcial es

!, A

Puesto que $\mathbf{a}_2 = A$ tiene sucesor, la ruta es hacia la Fase 3.

Fase 3: (a) Se incrementa k en 1 y se define $\mathbf{a}_3 = \mathbf{Suc}(A) = *$. Después de hacer estos cambios, la situación se resume en

$$k = 3, \quad \text{y} \quad \mathbf{a}_1 = !, \quad \mathbf{a}_2 = A, \quad \mathbf{a}_3 = *.$$

(b) Se Retorna a la Fase 2.

Fase 2: Ahora el más reciente de los elementos incorporados a la lista es $\mathbf{a}_3 = *$.

Puesto que \mathbf{a}_3 tiene sucesor, la ruta es hacia la Fase 3.

Fase 3: (a) Se incrementa $k = 3$ en 1 y se define $\mathbf{a}_4 = \mathbf{Suc}(*) = 0$. Después de hacer estos cambios, la situación se resume en

$$k = 4, \quad \text{y} \quad \mathbf{a}_1 = !, \quad \mathbf{a}_2 = A, \quad \mathbf{a}_3 = *, \quad \mathbf{a}_4 = 0.$$

(b) Se Retorna a la Fase 2.

Fase 2: El último elemento agregado a la lista es $\mathbf{a}_4 = 0$ y no tiene sucesor. Por lo tanto, la ruta es hacia la Fase 4.

Fase 4: El Algoritmo concluye: \mathcal{A} tiene $k = 4$ elementos, y la lista de los elementos de \mathcal{A} generada por el algoritmo es $!, A, *, 0$. \square

Teorema 5.1. Dado un conjunto finito \mathcal{A} dotado con un orden completo, el algoritmo general de listado produce una enumeración de todos sus elementos.

Demostración. El argumento es por inducción en el número de elementos de \mathcal{A} . Se demostrará que

$$\begin{aligned} &\text{si } \mathcal{A} \text{ tiene } m \text{ elementos, entonces el algoritmo termina} \\ &\text{después de } m \text{ visitas a la Fase 2.} \end{aligned} \tag{5.1}$$

Esta afirmación es claramente válida para $m = 2$. Suponga que (5.1) ocurre cuando un conjunto \mathcal{A} dotado con un orden completo tiene $m = n \geq 2$ elementos, y sea \mathcal{A} un conjunto con $n + 1$ miembros, entre los cuales se ha construido un orden total. Si \mathbf{M} es el máximo de \mathcal{A} defina el nuevo conjunto

$$\tilde{\mathcal{A}} = \mathcal{A} \setminus \{\mathbf{M}\},$$

el cual tiene n miembros (uno menos que el número de elementos de \mathcal{A}). Por lo tanto, aplicando el algoritmo al conjunto $\tilde{\mathcal{A}}$ la hipótesis de inducción implica que, después de n visitas a la etapa 2, se habrá completado una lista de todos los miembros de $\tilde{\mathcal{A}}$. De hecho, en la visita número n , se encontrará que \mathbf{a}_n no tiene sucesor en $\tilde{\mathcal{A}}$, y la bifurcación será a la Fase 4. Sin embargo, esto último ocurre si el interés se centra en $\tilde{\mathcal{A}}$. Si lo que se busca es listar los miembros del conjunto original \mathcal{A} , se detectará que \mathbf{a}_n , si tiene sucesor (el cual es \mathbf{M}) y la bifurcación será hacia el Fase 3, en el cual se pasará de $k = n$ a $k = n + 1$ y se definirá $\mathbf{a}_{n+1} = \mathbf{Suc}(\mathbf{a}_n) = \mathbf{M}$. Después de este paso, se retornará a la Fase 2 haciendo la visita $n + 1$, en la cual se detectará que \mathbf{M} no tiene sucesor, y el flujo se dirigirá hacia la Fase 4, finalizando la lista de los miembros del conjunto original con $n + 1$ pasos por la Fase 2. Este argumento verifica que (5.1) es válido con $m = n + 1$, y concluye el argumento de inducción. \square

El Orden Lexicográfico

Cuando un conjunto finito \mathcal{A} está dotado con un orden total (completo), el algoritmo propuesto en la sección anterior produce una lista de todos los miembros. De hecho, combinando el Teorema 5.1 con los comentarios plasmados en la Sección 2, es claro que los problemas de *generar una lista de los miembros de un conjunto finito* y de *definir una relación de orden total* entre sus miembros, son problemas equivalentes.

En las aplicaciones, la utilización del Teorema 5.1 para generar listados, depende de que se tenga a la mano una relación de orden completa. Los ejemplos presentados hasta ahora han tenido una finalidad ilustrativa, y han dejado a un lado la consideración de los conjuntos más complejos—la clase de permutaciones y la familia de subconjuntos—que son de interés primordial. Para dichos conjuntos, el problema de generar una lista de sus miembros tiene gran importancia en la estadística no paramétrica; sin embargo, dichos conjuntos se ubican dentro de \mathbb{R}^p para algún entero $p \geq 2$, y es necesario introducir una relación de orden en este espacio multidimensional.

Definición 6.1. La relación de orden lexicográfico en \mathbb{R}^p está determinada de la siguiente manera: Si $\mathbf{a} = (a_1, a_2, \dots, a_p)$ y $\mathbf{b} = (b_1, b_2, \dots, b_p)$ son dos miembros arbitrarios de \mathbb{R}^p , entonces

$$\mathbf{a} \prec \mathbf{b} \quad \text{si y sólo si existe un entero } k \leq p \text{ tal que}$$

$$a_i = b_i \quad \text{para } 1 \leq i < k, \text{ y } a_k < b_k.$$

Verbalmente, esta relación puede describirse como sigue: Para comparar dos miembros $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, recorra \mathbf{a} y \mathbf{b} de izquierda a derecha simultáneamente, hasta encontrar la primera posición en que las componentes correspondientes de \mathbf{a} y \mathbf{b} difieran. En este caso, se declara que $\mathbf{a} \prec \mathbf{b}$ si en dicha posición la componente de \mathbf{b} es mayor que la de \mathbf{a} , mientras que $\mathbf{b} \prec \mathbf{a}$ en otro caso; por supuesto, si todas las componentes de \mathbf{a} y de \mathbf{b} coinciden, se tiene que $\mathbf{a} = \mathbf{b}$. Por ejemplo, considere

el caso $p = 5$ y sean

$$\mathbf{a} = (5, 5, 4, 7, 10), \quad \mathbf{b} = (5, 5, 4, 10, 11).$$

En este caso, al recorrer \mathbf{a} y \mathbf{b} de izquierda a derecha, se observa que difieren en la posición número $k = 4$, y que la cuarta componente de \mathbf{b} es 10, la cual es mayor a la cuarta componente de \mathbf{a} (la cual es 7):

$$a_i = b_i, \quad 1 \leq i < 4, \quad a_4 < b_4.$$

De acuerdo a la Definición 6.1, se desprende que $\mathbf{a} \prec \mathbf{b}$. El siguiente teorema establece que la relación introducida en la Definición 6.1 es, efectivamente, una relación de orden total.

Teorema 6.1. La ordenación lexicográfica en \mathbb{R}^p es completa (total) en el sentido de la Definición 3.1.

Demostración. Se verificará que el orden lexicográfico tiene las propiedades de transitividad y completez:

(i) [Transitividad.] Suponga que $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^p$ son tales que

$$\mathbf{a} \prec \mathbf{b}, \quad \text{y} \quad \mathbf{b} \prec \mathbf{c}. \tag{6.1}$$

En este caso, existen enteros k_1 y k_2 entre 1 y p tales que

$$a_i = b_i, \quad 1 \leq i < k_1, \quad a_{k_1} < b_{k_1}, \quad \text{y} \quad b_i = c_i, \quad 1 \leq i < k_2, \quad b_{k_2} < c_{k_2}. \tag{6.2}$$

Considere ahora los siguientes casos:

(a) $k = k_1 = k_2$. En estas circunstancias, (6.2) implica que $a_i = c_i$ siempre que $1 \leq i < k$, mientras que $a_k = a_{k_1} < b_{k_1} = b_{k_2} < c_{k_2} = c_k$, esto es, en la primera posición para la cual las componentes de \mathbf{a} y \mathbf{c} difieren, la componente \mathbf{c} es mayor.

Por lo tanto,

$$\mathbf{a} \prec \mathbf{c}.$$

(b) $k_1 < k_2$. En este caso, defina $k = k_1$. Utilizando (6.2), se tiene que

$$a_i = b_i, \quad 1 \leq i < k, \quad \text{y} \quad a_k < b_k.$$

Además, debido a que $k < k_2$, se tiene que $b_i = c_i$, $1 \leq i \leq k$, Por lo tanto, la última relación desplegada implica que

$$a_i = c_i, \quad 1 \leq i < k, \quad \text{y} \quad a_k < c_k,$$

de tal forma que $\mathbf{a} \prec \mathbf{c}$.

(c) $k_1 > k_2$. En este caso, defina $k = k_2$. De nueva cuenta, utilizando (6.2), se obtiene, puesto que $k_1 > k = k_2$, que

$$a_i = b_i, \quad 1 \leq i \leq k,$$

mientras que

$$b_i = c_i, \quad 1 \leq i < k, \quad b_k < c_k.$$

Por lo tanto, a partir de estas relaciones desplegadas se obtiene

$$a_i = b_i, \quad 1 \leq i < k, \quad a_k = b_k < c_k,$$

de tal forma que, $\mathbf{a} \prec \mathbf{c}$. Este argumento ha mostrado que, en cualquier circunstancia, se tiene que $\mathbf{a} \prec \mathbf{c}$ siempre que (6.1) sea válida, esto es,

$$\mathbf{a} \prec \mathbf{b}, \quad \text{y} \quad \mathbf{b} \prec \mathbf{c} \implies \mathbf{a} \prec \mathbf{c},$$

estableciendo la propiedad de transitividad.

(ii) [Completez.] Dados dos vectores $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, debe demostrarse que exactamente una de las siguientes alternativas ocurre:

$$\mathbf{a} \prec \mathbf{b}, \quad \mathbf{a} = \mathbf{b}, \quad \text{o} \quad \mathbf{b} \prec \mathbf{a}. \quad (6.3)$$

Suponga que $\mathbf{a} \neq \mathbf{b}$. En este caso, \mathbf{a} y \mathbf{b} difieren en alguna componente; defina k como el primer entero en el que las componente correspondientes de \mathbf{a} y \mathbf{b} son distintas, esto es,

$$a_i = b_i, \quad 1 \leq i < k, \quad a_k \neq b_k.$$

Si $a_k < b_k$, la Definición 6.1 implica que $\mathbf{a} \prec \mathbf{b}$, mientras que si $a_k > b_k$, entonces $\mathbf{b} \prec \mathbf{a}$. Por lo tanto, se ha mostrado que si $\mathbf{a} \neq \mathbf{b}$, entonces exactamente una de las otras dos relaciones en (6.3) ocurre; esto establece que el orden lexicográfico es completo, y concluye la demostración del teorema. \square

Conclusión

En este capítulo se ha estudiado el problema general de listado (enumeración) para un conjunto finito \mathcal{A} . La presentación puso de manifiesto la equivalencia entre este problema y la construcción de una relación de orden total entre los miembros de \mathcal{A} , y se formuló un *algoritmo general de listado*, el cual permite enumerar todos los miembros de un conjunto finito arbitrario, tan pronto como se haya definido una relación de orden total entre sus miembros. Sin embargo, debe señalarse que la aplicación efectiva de dicho procedimiento depende de las siguientes condiciones

- (i) Construir un orden total en el conjunto de interés;
- (ii) Determinar el elemento mínimo del conjunto;
- (iii) Especificar completamente la función **Suc**, la cual transforma a un miembro de \mathcal{A} que no sea el máximo, en su sucesor.

El punto (i) fue considerado en la Sección 6, donde se introdujo el orden lexicográfico en \mathbb{R}^p , y se demostró que es un orden total, de manera que un conjunto contenido en \mathbb{R}^p , siempre puede considerarse dotado del orden lexicográfico. Sin embargo, los aspectos (ii) y (iii) son específicos del conjunto \mathcal{A} bajo consideración, esto es, la dificultad de determinar el mínimo de \mathcal{A} y la función **Suc** dependen de las

características de cada conjunto contenido en \mathbb{R}^p , e implican un análisis detallado cuya complejidad depende del caso específico analizado. En los siguientes capítulos se utilizará la teoría desarrollada para establecer algoritmos que permiten enumerar familias de permutaciones y clases de subconjuntos, problemas que, como ya se ha comentado, son de gran interés en la aplicación de métodos estadísticos no paramétricos.

Capítulo 4

Permutaciones de Longitud Máxima

Este capítulo trata sobre el problema de listar todas las permutaciones de una población finita para la cual se dispone de una enumeración. El punto de partida para abordar este problema es identificar a la población de interés con los primeros N enteros positivos, de manera que la clase de permutaciones está contenida en \mathbb{R}^N , espacio en el cual el orden lexicográfico es total. Utilizando este hecho, el problema se analiza a través del procedimiento general de listado que se formuló en el capítulo anterior, arribando, finalmente, a un algoritmo de listado con estructura simple. La exposición enfatiza la relevancia que el problema de generar la familia de permutaciones tiene en el análisis estadístico

Introducción

Sea \mathcal{P} un conjunto finito dado, para el cual se dispone de una manera de enumerar sus miembros, de tal forma que, sin pérdida de generalidad, puede suponerse que

$$\mathcal{P} = \{1, 2, \dots, N - 1, N\}, \quad (1.1)$$

donde N es un entero positivo. El desarrollo subsecuente supone que $N \geq 2$, pues una población con un sólo elemento no tiene interés en el problema que se estudiará. Dada la población \mathcal{P} en (1.1), el correspondiente conjunto de permutaciones de tamaño (máximo) N se denota mediante $\mathcal{P}(N, N)$, y se define a través de

$$\mathcal{P}(N, N) = \{\mathbf{a} = (a_1, a_2, \dots, a_N), a_i \in \mathcal{P}, \text{ para todo } i, a_i \neq a_j, \text{ cuando } i \neq j.\} \quad (1.2)$$

Verbalmente, los miembros de $\mathcal{P}(N, N)$ son arreglos que se obtienen colocando los elementos de la población uno tras otro, sin repeticiones y en el orden que se desee.

Desde luego, $\mathcal{P}(N, N)$ tiene $N!$ miembros, pero contar los elementos de un conjunto es un problema muy distinto a generarlos todos mediante un procedimiento sistemático, problema que es precisamente el tema de este capítulo. La idea fundamental para el análisis, es la observación de que $\mathcal{P}(N, N)$ consiste de vectores de dimensión N , esto es, $\mathcal{P}(N, N) \subset \mathbb{R}^N$, y que éste último espacio está dotado con un orden total, a saber, el orden lexicográfico. Por lo tanto, es posible utilizar la teoría general desarrollada en el Capítulo 3 para generar un método de listado de todos los miembros de $\mathcal{P}(N, N)$. Para alcanzar este objetivo, hay dos problemas fundamentales que deben abordarse: (i) La determinación del elemento mínimo de $\mathcal{P}(N, N)$, y (ii) Diseñar un algoritmo que permita obtener el sucesor de cualquier miembro de $\mathcal{P}(N, N)$ que no sea el elemento máximo, aspectos a los que se dedica la mayor parte del esfuerzo desplegado en este capítulo. Por otro lado, existe, por lo menos, otra forma de obtener una lista de todas las permutaciones; vea por ejemplo, Even (1973), o Hall (1971), donde se describe un algoritmo basado en transposiciones de elementos adyacentes, el cual fue propuesto por Johnson (1963). Sin embargo, el enfoque seguido en esta tesis, tiene el rasgo distintivo de seguir una ruta bien delineada dentro de una teoría general, que puede utilizarse para listar cualquier conjunto totalmente ordenado, a condición de que se entienda suficientemente el orden definido, en el sentido de que pueda determinarse el elemento mínimo del conjunto así como la función sucesor.

La exposición ha sido organizada de la siguiente manera: En la Sección 2 se ubica al problema de listar las permutaciones de una población dentro de un contexto estadístico, enfatizando el papel que $\mathcal{P}(N, N)$ desempeña dentro de los métodos no paramétricos, mientras que en la Sección 3 se determina el elemento mínimo de $\mathcal{P}(N, N)$, iniciando el trayecto hacia la generación de todas las permutaciones de la población \mathcal{P} . En la Sección 4 se introduce la idea de *índice de ruptura* de una pareja de permutaciones, noción que desempeña un importante papel en la caracterización del sucesor de un miembro de $\mathcal{P}(N, N)$ presentada en

la Sección 5. En la Sección 6 se utilizan los resultados establecidos para formular un algoritmo que permite listar todos los miembros de $\mathcal{P}(N, N)$, el cual puede codificarse de manera simple, y dicho procedimiento se implementa en la Sección 7. Después de establecer el resultado principal, la presentación concluye en la Sección 8 con algunos comentarios breves.

La Perspectiva Estadística

Una razón fundamental para considerar el problema de listar la clase $\mathcal{P}(N, N)$, es su ocurrencia constante en el área de estadística no paramétrica, área en la que el siguiente resultado desempeña un papel central.

Lema 2.1. [Randles y Wolfe (1981), Meddis (1984).] Sean X_1, X_2, \dots, X_N son variables aleatorias independientes con una densidad común, y denote mediante R_i al rango de X_i cuando los datos muestrales se colocan en orden ascendente. En este caso, el vector $\mathbf{R} = (R_1, R_2, \dots, R_N)$ tiene distribución uniforme en el conjunto $\mathcal{P}(N, N)$, esto es, para cada $\mathbf{r} = (r_1, r_2, \dots, r_N) \in \mathbb{R}^N$

$$P[\mathbf{R} = \mathbf{r}] = P[R_1 = r_1, R_2 = r_2, \dots, R_N = r_N] = \frac{1}{N!}.$$

Este resultado es fundamental para construir, por ejemplo, métodos no paramétricos para probar hipótesis. Por ejemplo, considere un estadístico de la forma

$$S = \sum_{i=1}^N a(i)c(R_i),$$

donde $a(\cdot)$ y $c(\cdot)$ son dos funciones dadas. En este caso, S se denomina *estadístico de rango lineal*, y por el Lema 2.1, su distribución no depende cual es la densidad común de las variables X_i ; esto significa que S posee la propiedad de ser *de distribución libre*. Sin embargo, en general la determinación de la distribución exacta de S no es una tarea simple, razón por la cual se ha estudiado la distribución límite de S . Los resultados en esta dirección adoptan la siguiente forma (Meddis, 1984):

Si S^* es una estandarización de S , entonces la distribución de S^* se aproxima a una *distribución fija* \mathcal{D} conforme N aumenta:

$$S^* \xrightarrow{\mathcal{D}} \mathcal{D} \quad \text{conforme } N \rightarrow \infty; \quad (2.1)$$

esta distribución límite \mathcal{D} es normal, pero esta característica no desempeña papel alguno en esta discusión. La implicación práctica de (2.1), es que si el tamaño de la población—i.e., N —es ‘grande’, entonces una probabilidad que involucra a S puede aproximarse mediante la distribución \mathcal{D} , hecho que inmediatamente conduce a dos preguntas: (i) ¿Qué tan buena es la aproximación si N es ‘pequeño’? (ii) ¿Cuál es la bondad de la aproximación de la distribución de S mediante \mathcal{D} ? Para responder estas preguntas, es necesario conocer la distribución exacta de S , y una manera de determinarla, es calculando el valor que S asume en cada permutación de $\mathcal{P}(N, N)$, lo cual requiere, en primera instancia, producir una lista de todos los miembros de esa familia. Por lo tanto,

Generar una lista de las permutaciones de la población \mathcal{P} , es un ingrediente básico para analizar el problema de la bondad con que \mathcal{D} aproxima a la distribución de S^* .

Estas consideraciones ponen de manifiesto que el problema de generar una lista de la familia $\mathcal{P}(N, N)$, además de ser un problema interesante en si mismo, tiene repercusiones en el análisis de cuestiones fundamentales en la disciplina estadística, lo cual proporciona una fuerte motivación para el trabajo que se desarrolla en este capítulo.

Las Permutaciones Mínima y Máxima

Esta sección es el punto de partida en la ruta hacia la generación de un listado de todas las permutaciones de la población \mathcal{P} . De acuerdo al procedimiento delineado en la Sección 5 del Capítulo 3, el primer paso en esta dirección es determinar el elemento mínimo de $\mathcal{P}(N, N)$, tarea que se realiza a continuación.

Proposición 3.1. (i) El elemento mínimo de $\mathcal{P}(N, N)$ dotado con el orden lexicográfico es

$$\mathbf{m} = (1, 2, 3, \dots, N).$$

Similaremente,

(ii) El elemento máximo de $\mathcal{P}(N, N)$ is

$$\mathbf{M} = (N, N - 1, N - 2, \dots, 1).$$

Demostración. (i) Dado un elemento $\mathbf{a} \in \mathcal{P}(N, N)$ se verificará que

$$\mathbf{a} \neq \mathbf{m} \implies \mathbf{m} \prec \mathbf{a}.$$

Para comprobar esta afirmación, suponga que $\mathbf{a} = (a_1, a_2, \dots, a_N)$ es diferente de \mathbf{m} , de manera que $a_s \neq m_s = s$ para algún entero s . Sea k la primera posición en que difieren las componente de \mathbf{a} y \mathbf{m} . esto es,

$$a_i = m_i = i \quad \text{si } i < k, \text{ mientras que } \quad a_k \neq m_k = k. \quad (3.1)$$

Puesto que la inclusión $a_k \in \{1, 2, 3, \dots, N\} \setminus \{a_1, a_2, \dots, a_{k-1}\}$ siempre es válida, usando que $a_k \neq m_k = k$ se desprende que

$$\begin{aligned} a_k &\in \{1, 2, 3, \dots, N\} \setminus (\{a_1, a_2, \dots, a_{k-1}\} \cup \{m_k\}) \\ &= \{1, 2, 3, \dots, N\} \setminus (\{m_1, m_2, \dots, m_{k-1}\} \cup \{m_k\}) \\ &= \{1, 2, 3, \dots, N\} \setminus (\{1, 2, \dots, k-1\} \cup \{k\}) \\ &= \{k+1, \dots, N\} \end{aligned}$$

y entonces $a_k > k = m_k$; por lo tanto, $\mathbf{m} \prec \mathbf{a}$.

(ii) Se mostrará que para todo $\mathbf{a} \in \mathcal{P}(N, N)$

$$\mathbf{a} \neq \mathbf{M} \implies \mathbf{a} \prec \mathbf{M}.$$

Para verificar esta implicación, note que cuando $\mathbf{a} \neq \mathbf{M}$ existe un entero k tal que

$$a_i = M_i \quad \text{para } i < k, \text{ y } \quad a_k \neq M_k. \quad (5.2)$$

Recordando que la inclusión $a_k \in \{1, 2, 3, \dots, N\} \setminus \{a_1, a_2, \dots, a_{k-1}\}$ siempre es válida, (5.2) implica que

$$\begin{aligned} a_k &\in \{1, 2, 3, \dots, N\} \setminus (\{a_1, a_2, \dots, a_{k-1}\} \cup \{M_k\}) \\ &\in \{1, 2, 3, \dots, N\} \setminus (\{M_1, M_2, \dots, M_{k-1}\} \cup \{M_k\}) \\ &= \{1, 2, 3, \dots, N\} \setminus (\{N, N-1, \dots, N-k+2\} \cup \{N-k+1\}) \\ &= \{1, 2, \dots, N-k\} \end{aligned}$$

y entonces $a_k < N-k+1 = M_k$; puesto que $a_i = M_i$ para $1 \leq i < k$, se desprende que $\mathbf{a} \prec \mathbf{M}$. □

El Índice de Ruptura

El siguiente paso para implementar el algoritmo de listado de la clase $\mathcal{P}(N, N)$, es determinar el sucesor de cada permutación deferente de \mathbf{M} . El análisis para lograr este objetivo es más elaborado, y se hará en varias etapas. Primero, es conveniente introducir una idea simple que desempeña un papel importante para caracterizar al sucesor de una permutación.

Definición 4.1. Si \mathbf{a} y \mathbf{b} son dos miembros diferentes de la clase $\mathcal{P}(N, N)$, el *índice de ruptura* asociado a la pareja (\mathbf{a}, \mathbf{b}) , es el primer entero para el cual las componentes correspondientes de \mathbf{a} y \mathbf{b} difieren, esto es, el índice de ruptura es el entero k que satisface

$$a_i = b_i, \quad 1 \leq i < k, \quad a_k \neq b_k.$$

Para ejemplificar esta noción, suponga que $N = 4$. En este caso, si $\mathbf{a} = (3, 2, 1, 4)$ y $\mathbf{b} = (3, 2, 4, 1)$, el índice de ruptura correspondiente a este par de permutaciones es $k = 3$; en efecto, $a_1 = b_1 = 3$ y $a_2 = b_2 = 2$, pero $a_3 = 1 \neq 4 = b_3$, de donde se desprende que la primera coordenada en que difieren \mathbf{a} y \mathbf{b} es la tercera. El objetivo específico de esta sección es determinar condiciones bajo las

cuales puede encontrarse una permutación \mathbf{b} que suceda a \mathbf{a} , y tal que el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) sea k . El primer paso hacia esta meta es la siguiente observación.

Proposición 4.1. Si \mathbf{a} y \mathbf{b} son dos permutaciones diferentes en $\mathcal{P}(N, N)$, entonces, el índice de ruptura de \mathbf{a} y \mathbf{b} es menor que N .

Demostración. Sean $\mathbf{a}, \mathbf{b} \in \mathcal{P}(N, N)$ dos permutaciones distintas. En este caso, puesto que $\{a_1, a_2, \dots, a_N\}$ es el conjunto $\{1, 2, \dots, N\}$, se tiene que

$$\{a_N\} = \{1, 2, \dots, N\} \setminus \{a_1, a_2, \dots, a_{N-1}\}.$$

Similarmente,

$$\{b_N\} = \{1, 2, \dots, N\} \setminus \{b_1, b_2, \dots, b_{N-1}\}.$$

Por lo tanto $a_i = b_i$, $i = 1, 2, 3, \dots, N - 1$ implica que $\{a_N\} = \{b_N\}$, y entonces $a_N = b_N$. Este argumento muestra que el primer índice en el cual dos permutaciones difieren no puede ser N , i.e., el índice de ruptura de dos permutaciones distintas es, invariablemente, menor que N . \square

El siguiente paso para determinar el sucesor de un miembro de $\mathcal{P}(N, N) \setminus \{\mathbf{M}\}$, es estudiar las permutaciones que suceden a cierta permutación fija \mathbf{a} , y cuyo índice de ruptura con \mathbf{a} es un entero determinado. Con esta finalidad, es conveniente introducir la siguiente notación.

Definición 4.2. Sea $\mathbf{a} \in \mathcal{P}(N, N)$ una permutación fija, la cual se supone diferente de \mathbf{M} . Para cada entero positivo $k < N$, la clase $\mathcal{C}_k(\mathbf{a})$ consiste de todas las permutaciones \mathbf{b} que satisfacen

$$\mathbf{a} < \mathbf{b} \quad \text{y el índice de ruptura de } \mathbf{a} \text{ y } \mathbf{b} \text{ es } k.$$

La siguiente proposición establece condiciones necesarias y suficientes para que una clase $\mathcal{C}_k(\mathbf{a})$ sea no vacía.

Proposición 4.2. Para un entero $k < N$ y una permutación $\mathbf{a} \in \mathcal{P}(N, N)$, las siguientes afirmaciones (a) y (b) son equivalente:

(a) La clase $\mathcal{C}_k(\mathbf{a})$ es no vacía;

(b) Para algún entero s que satisface $k < s \leq N$, se tiene que

$$a_s > a_k.$$

En palabras, la Proposición 4.2 establece que la clase $\mathcal{C}_k(\mathbf{a})$ es no vacía sólo cuando al recorrer la permutación \mathbf{a} de izquierda a derecha, se encuentra una una componente mayor que a_k después de la k -ésima posición.

Demostración. (a) \Rightarrow (b). Suponga que $\mathcal{C}_k(\mathbf{a})$ es diferente del conjunto vacío, y seleccione $\mathbf{b} \in \mathcal{C}_k(\mathbf{a})$. Por la definición de $\mathcal{C}_k(\mathbf{a})$ se tiene que $\mathbf{a} \prec \mathbf{b}$ y la primera componente en que \mathbf{a} y \mathbf{b} difieren es la k -ésima, es decir,

$$a_i = b_i, \quad i < k, \quad \text{y} \quad a_k < b_k \tag{4.1}$$

En este caso,

$$\begin{aligned} b_k &\in \{1, 2, \dots, N\} \setminus \{b_1, b_2, \dots, b_{k-1}\} \\ &= \{1, 2, \dots, N\} \setminus \{a_1, a_2, \dots, a_{k-1}\} \\ &= \{a_k, a_{k+1}, \dots, a_N\} \end{aligned}$$

y usando que $b_k > a_k$, se desprende que

$$b_k \in \{a_{k+1}, \dots, a_N\},$$

así que $b_k = a_s$ para algún entero $s > k$. Entonces, (4.1) implica que $a_s > a_k$ para cierto entero $s > k$.

(b) \Rightarrow (a). Suponga que $a_s > a_k$ para algún entero $s \in \{k+1, \dots, N\}$, y defina $\mathbf{b} \in \mathcal{P}(N, N)$ como sigue:

$$\mathbf{b} = (a_1, a_2, \dots, a_{k-1}, a_s, a_{k+1}, \dots, a_{s-1}, a_k, a_{s+1}, \dots, a_N),$$

i.e, \mathbf{b} se obtiene a partir de \mathbf{a} intercambiando los elementos s y k . Por lo tanto,

$$b_i = a_i \quad i < k, \quad b_k = a_s > a_k,$$

lo cual significa que $\mathbf{b} \succ \mathbf{a}$, y que el índice de ruptura de \mathbf{a} y \mathbf{b} es k , esto es, $\mathbf{b} \in \mathcal{C}_k(\mathbf{a})$. \square

Caracterización del Sucesor

En esta sección se combinan las Proposiciones 4.1 y 4.2 para obtener una caracterización del sucesor de una permutación $\mathbf{a} \in \mathcal{P}(N, N)$ cuando $\mathbf{a} \neq \mathbf{M}$. El resultado principal se presenta en el Teorema 5.1 establecido más adelante, el cual requiere el siguiente resultado preliminar.

Lema 5.1. Sean \mathbf{b} y \mathbf{d} dos permutaciones tales que $\mathbf{b} \succ \mathbf{a}$ y $\mathbf{d} \succ \mathbf{a}$. Si $\mathbf{d} \in \mathcal{C}_{k_1}(\mathbf{a})$ y $\mathbf{b} \in \mathcal{C}_k(\mathbf{a})$ donde $k_1 > k$, entonces

$$\mathbf{b} \succ \mathbf{d}$$

Verbalmente, este lema establece que si dos sucesiones \mathbf{b} y \mathbf{c} suceden a una permutación \mathbf{a} , entonces *la menor* de \mathbf{b} y \mathbf{c} es la permutación cuyo índice de ruptura con \mathbf{a} es *mayor*.

Demostración del Lema 5.1. Sean \mathbf{b} y \mathbf{c} dos permutaciones, donde $\mathbf{b} \in \mathcal{C}_k(\mathbf{a})$ y $\mathbf{d} \in \mathcal{C}_{k_1}(\mathbf{a})$ y $k < k_1$. En este caso,

$$b_i = a_i, \quad i < k, \quad a_k < b_k,$$

y

$$d_i = a_i, \quad i < k_1, \quad a_{k_1} < d_{k_1}.$$

Puesto que $k < k_1$, se sigue que

$$b_i = d_i, \quad i < k, \quad b_k > d_k,$$

de manera que $\mathbf{b} \succ \mathbf{d}$. □

Teorema 5.1. [El sucesor de una permutación.] Sea $\mathbf{a} \in \mathcal{P}(N, N) \setminus \{\mathbf{M}\}$ una permutación arbitraria, y defina el entero k^* y la permutación \mathbf{a}^* mediante

$$k^* = \max\{k \mid 1 \leq k \leq N, \text{ y } \mathcal{C}_k(\mathbf{a}) \neq \emptyset\}, \quad (5.1)$$

y

$$\mathbf{a}^* = \min \mathcal{C}_{k^*}(\mathbf{a}). \quad (5.2)$$

En este caso, \mathbf{a}^* es el sucesor de \mathbf{a} , esto es,

$$\mathbf{a} \prec \mathbf{a}^*, \text{ y } [\mathbf{a} \prec \mathbf{b} \Rightarrow \mathbf{a}^* \preceq \mathbf{b}].$$

Demostración. Primero observe que $\mathbf{a}^* \in \mathcal{C}_{k^*}(\mathbf{a})$ implica que $\mathbf{a} \prec \mathbf{a}^*$, pues $\mathcal{C}_{k^*}(\mathbf{a})$ consiste sólo de permutaciones que suceden a \mathbf{a} . Sea $\mathbf{b} \in \mathcal{P}(N, N)$ una permutación tal que $\mathbf{a} \prec \mathbf{b}$. En este caso, $\mathbf{b} \in \mathcal{C}_k(\mathbf{a})$ para algún entero k . En particular, $\mathcal{C}_k(\mathbf{a}) \neq \emptyset$, y entonces $k \leq k^*$ (vea (5.1)). Considere ahora los dos siguientes casos exhaustivos:

Caso 1: $k < k^*$.

En esta situación usando que $\mathbf{a}^* \in \mathcal{C}_{k^*}(\mathbf{a})$, el Lema 5.1 implica que $\mathbf{a}^* \prec \mathbf{b}$.

Caso 2: $k = k^*$.

En estas circunstancias, $\mathbf{b} \in \mathcal{C}_{k^*}(\mathbf{a})$, y entonces $\mathbf{a}^* \preceq \mathbf{b}$, puesto que \mathbf{a}^* es el elemento mínimo de $\mathcal{C}_{k^*}(\mathbf{a})$; vea (5.2).

En resumen, se ha mostrado que $\mathbf{a} \prec \mathbf{a}^*$, y que $\mathbf{a}^* \preceq \mathbf{b}$ siempre que $\mathbf{a} \prec \mathbf{b}$.

Por lo tanto, \mathbf{a}^* es el sucesor de \mathbf{a} . □

Cálculo del Sucesor de una Permutación

El uso efectivo de la caracterización establecida en el Teorema 5.1 para el sucesor de una permutación determinada en $\mathcal{P}(N, N) \setminus \{\mathbf{M}\}$, requiere un método

‘práctico’ para determinar k^* en (5.1), así como \mathbf{a}^* , el elemento mínimo de la clase $\mathcal{C}_{k^*}(\mathbf{a})$ (vea (5.2)), problemas que se consideran en esta sección. Primero se aborda la determinación de k^* .

Proposición 6.1. Sea $\mathbf{a} \neq \mathcal{P}(N, N)$ una permutación dada. El siguiente procedimiento permite determinar si $\mathbf{a} = \mathbf{M}$ o si $\mathbf{a} \prec \mathbf{M}$, y en este último caso, se obtiene el valor de k^* en (5.1).

Datos: Una permutación \mathbf{a} de la población $\mathcal{P} = \{1, 2, \dots, N\}$.

Inicio: Defina $K = N - 1$.

[Bloque Iterativo]

Repetir Mientras $K \geq 1$

Si $a_K > a_{K+1}$ Entonces Disminuya K en una Unidad

Si $a_K < a_{K+1}$ Entonces Salga del Ciclo

Fin del Ciclo Repetir

Fin del Procedimiento:

Si $K = 0$ entonces $\mathbf{a} = \mathbf{M}$

Si $K > 0$ entonces $\mathbf{a} \prec \mathbf{M}$ y $k^* = K$.

Demostración. Suponga que al concluir el procedimiento se tiene que $K = 0$. La única forma en que esto puede ocurrir es que en los diversos pasos por el ciclo del algoritmo se disminuya K en una unidad, lo cual sólo acontece si las siguientes desigualdades ocurren:

$$\underbrace{a_{N-1} > a_N}_{K=N-1}, \quad \underbrace{a_{N-2} > a_{N-1}}_{K=N-2}, \quad \dots, \quad \underbrace{a_2 > a_3}_{K=2}, \quad \underbrace{a_1 > a_2}_{K=1}. \quad (6.1)$$

En el extremo izquierdo de esta relación aparece la desigualdad $a_{N-1} > a_N$ la cual se verifica al entrar al ciclo por primera vez, cuando $K = N - 1$. Esto conduce a

disminuir K en una unidad, para producir $K = N - 2$ de manera que al volver a entrar al ciclo se verifica que $a_{N-2} > a_{N-1}$, y en consecuencia se disminuye K en una unidad obteniendo $K = N - 3$. Este proceso continua hasta que se verifica la desigualdad $a_1 > a_2$ cuando $K = 1$, disminuyendo de nueva cuenta el valor de K , el cual alcanza en ese momento el valor $K = 0$. Al intentar entrar otra vez al ciclo se encuentra que la condición de entrada—la cual es $K \geq 1$ —no se satisface, por lo que el flujo brinca el ciclo y llega al final del procedimiento. En resumen, la discusión muestra que si $K = 0$, todas las desigualdades en (6.1) ocurren, lo cual significa que

$$a_1 > a_2 > a_3 > \cdots > a_{N-2} > a_{N-1} > a_N.$$

Como la única forma de colocar los elementos de \mathcal{P} en orden decreciente es en el orden $N, N-1, N-2, \dots, 3, 2, 1$, se concluye que $\mathbf{a} = \mathbf{M}$ es el máximo del conjunto \mathcal{P} . Para concluir, suponga que el valor de K al final del procedimiento es positivo. Esto significa que la salida del ciclo no se origina en que la condición de entrada se viole, sino en el hecho de que la desigualdad $a_K > a_{K+1}$ no se satisface. Puesto que $a_K \neq a_{K+1}$, se tiene que

$$a_K < a_{K+1}.$$

Además, si la salida del ciclo ocurre para este valor de K significa que las desigualdades

$$a_{N-1} > a_N, \quad a_{N-2} > a_{N-1}, \quad a_{K+1} > a_{K+2}$$

se verificaron, de manera que

$$a_K < a_{K+1} > a_{K+2} > \cdots > a_{N-2} > a_{N-1} > a_N, \quad (6.2)$$

lo cual muestra que K es la posición máxima para la cual puede encontrarse una componente mayor a su derecha. Por lo tanto, si $K > 0$, entonces $K = k^*$, completando la demostración de la proposición. \square

Ejemplo 6.1. Considere el caso $N = 6$, y sea

$$\mathbf{a} = (3, 1, 4, 6, 5, 2).$$

El procedimiento de la Proposición 6.1 funciona como sigue aplicado a esta permutación:

Inicialmente, $K = 5$ ($= N - 1$).

Al intentar entrar al ciclo se prueba la condición $K \geq 1$, la cual se satisface y se produce la entrada al ciclo.

Dentro del ciclo se comparan $a_K = a_5 = 5$ y $a_{K+1} = a_6 = 2$.

Como $a_5 > a_6$ se disminuye K en 1 (ahora $K = 4$)

Se va a la entrada del ciclo.

Al intentar entrar al ciclo, se comprueba si la condición $K \geq 1$ es válida; como $K = 4$ se entra al ciclo

Dentro del ciclo se comparan $a_K = a_4 = 6$ y $a_{K+1} = a_5 = 5$.

Como $a_4 > a_5$ se disminuye K en 1 (ahora $K = 3$)

Se va a la entrada del ciclo.

Al intentar entrar al ciclo, se comprueba la condición $K \geq 1$; como $K = 3$ se entra al ciclo

Dentro del ciclo se comparan $a_K = a_3 = 4$ y $a_{K+1} = a_4 = 6$.

Como $a_3 < a_4$ se sale del ciclo.

Se arriba al final del procedimiento.

Como $K = 3$ es positivo, se concluye que $k^* = 3$.

Desde luego, esta conclusión es clara a partir de la forma establecida para la permutación \mathbf{a} , pero lo que se pretende es ilustrar el algoritmo contenido en la Proposición 6.1. \square

Observación 4.1. El procedimiento descrito en la Proposición 6.1 puede codificarse fácilmente en un lenguaje de programación. A continuación se presenta una posibilidad en *QBasic*:

$K = N - 1$

```

DO WHILE (K >= 1)
    IF (a(K) > a(K + 1)) THEN
        K = K - 1
    ELSE
        EXIT DO
    END IF
LOOP

```

Suponga ahora que $\mathbf{a} \neq \mathbf{M}$ es una permutación dada, y que el correspondiente índice k^* ha sido determinado. En el siguiente teorema se presenta un procedimiento computacional que genera elemento mínimo de $\mathcal{C}_{k^*}(\mathbf{a})$.

Teorema 6.1 Sea $\mathbf{a} \neq \mathbf{M}$ una permutación determinada, y sea k^* como en (5.1). La permutación mínima de $\mathcal{C}_{k^*}(\mathbf{a})$ se obtiene a través del siguiente procedimiento:

- (a) Determine el entero j dado por $j = \max\{i \mid k^* + 1 \leq i \leq N, a_i > a_{k^*}\}$.
- (b) Intercambie a_j and a_{k^*} en la permutación \mathbf{a} para obtener

$$\mathbf{b} = (a_1, a_2, \dots, a_{k^*-1}, a_j, a_{k^*+1}, \dots, a_{j-1}, a_{k^*}, a_{j+1}, \dots, a_N)$$

En este caso, los últimos $N - k^*$ miembros de \mathbf{b} se encuentran en orden descendente, esto es,

$$a_{k^*+1} > a_{k^*+2} > \dots > a_{j-1} > a_{k^*} > a_{j+1} > \dots > a_{N-1} > a_N. \quad (6.3)$$

- (c) Ordene los últimos $N - k^*$ elementos de \mathbf{b} en orden ascendente para obtener

$$\mathbf{a}^* = (a_1, a_2, \dots, a_{k^*-1}, a_j, a_N, a_{N-1}, \dots, a_{j+1}, a_{k^*}, a_{j-1}, \dots, a_{k^*+2}, a_{k^*+1}).$$

Entonces, \mathbf{a}^* es el mínimo de los miembros de $\mathcal{C}_{k^*}(\mathbf{a})$, y por lo tanto, $\mathbf{a}^* = \mathbf{Suc}(\mathbf{a})$.

Demostración. Primero, se verificará que (6.3) ocurre. Observe que (6.2) con $K = k^*$ implica que

$$a_{k^*} < a_{k^*+1} > a_{k^*+2} > \cdots > a_{N-1} > a_N,$$

y si el entero j es como en la parte (a) del procedimiento, entonces

$$a_{k^*+1} > a_{k^*+2} > \cdots >> a_{j-1} > a_j > a_{k^*} > a_{j+1} > \cdots > a_{N-1} > a_N,$$

pues j es el mayor entero dentro del conjunto $\{k^* + 1, k^* + 2, \dots, N\}$ para el cual $a_j > a_{k^*}$, estableciendo (6.3). Ahora suponga que \mathbf{c} es una permutación arbitraria en $\mathcal{C}_{k^*}(\mathbf{a})$. En este caso, k^* es el índice de ruptura \mathbf{c} y \mathbf{a} , y entonces

$$a_i = c_i \quad \text{para cada } i < k^*, \quad (6.4)$$

mientras que $c_{k^*} > a_{k^*}$. Puesto que

$$\begin{aligned} c_{k^*} &\in \{1, 2, \dots, N\} \setminus \{c_1, c_2, \dots, c_{k^*-1}\} \\ &= \{1, 2, \dots, N\} \setminus \{a_1, a_2, \dots, a_{k^*-1}\} \\ &= \{a_{k^*}, a_{k^*+1}, a_{k^*+2}, \dots, a_N\}, \end{aligned}$$

la desigualdad $c_{k^*} > a_{k^*}$ implica que $c_{k^*} \in \{a_{k^*+1}, a_{k^*+2}, \dots, a_N\}$. Notando que a partir de (6.3) se obtiene

$$a_{k^*+1} > a_{k^*+2} > \cdots > a_{j-1} > a_j > a_{k^*} > a_{j+1} > \cdots > a_N,$$

se arriba a la conclusión de que

$$c_{k^*} \geq a_j;$$

observe que es posible que $j = k^* + 1$. Suponga ahora que $\mathbf{c} \neq \mathbf{a}$ y considere las siguientes dos posibilidades:

Caso 1: $c_{k^*} > a_j$.

Bajo esta condición, (6.4) y la definición de \mathbf{a}^* implican que $\mathbf{c} \succ \mathbf{a}^*$.

Caso 2: $c_{k^*} = a_j$.

En este contexto, el índice de ruptura de \mathbf{c} y \mathbf{a}^* , denotado por s , es mayor que k^* , y entonces

$$a_i^* = c_i \quad \text{para } i = 1, 2, \dots, s-1, \quad \text{y } c_s \neq a_s^*. \quad (6.5)$$

Consecuentemente,

$$\begin{aligned} c_s &\in \{1, 2, \dots, N\} \setminus \{c_1, c_2, \dots, c_{s-1}\} \\ &= \{1, 2, \dots, N\} \setminus \{a_1^*, a_2^*, \dots, a_{s-1}^*\} \\ &= \{a_s^*, a_{s+1}^*, a_{s+2}^*, \dots, a_N^*\} \end{aligned}$$

y usando que $c_s \neq a_s^*$, se desprende que $c_s \in \{a_{s+1}^*, a_{s+2}^*, \dots, a_N^*\}$, así que

$$c_s \geq a_{s+1}^*$$

Finalmente, recordando que las componentes de \mathbf{a}^* crecen a partir del elemento $(k^* + 1)$, y que $s \geq k^* + 1$, la anterior desigualdad desplegada implica que

$$c_s > a_s^*,$$

relación que al combinarse con (6.5) implica que $\mathbf{c} \succ \mathbf{a}^*$

En resumen, se tiene que $\mathbf{a}^* \in \mathcal{C}_{k^*}(\mathbf{a})$, y se ha mostrado que cada permutación $\mathbf{c} \in \mathcal{C}_{k^*}(\mathbf{a}) \setminus \{\mathbf{a}^*\}$ satisface que $\mathbf{c} \succ \mathbf{a}^*$, de donde se obtiene que \mathbf{a}^* es el elemento mínimo $\mathcal{C}_{k^*}(\mathbf{a})$. Por lo tanto, la permutación \mathbf{a}^* es el sucesor de \mathbf{a} , por el Teorema 5.1. \square

Ejemplo 6.2. Considere el caso $N = 6$, y sea la permutación \mathbf{a} como en el Ejemplo 6.1, esto es

$$\mathbf{a} = (3, 1, 4, 6, 5, 2),$$

para la cual ya se ha determinado que $k^* = 3$. El procedimiento establecido en el Teorema 5.1 para determinar el sucesor de \mathbf{a}^* es como sigue:

(a) Determinar el mayor entero j dentro del conjunto $\{k^* + 1, \dots, N\}$ para el cual $a_j > a_{k^*}$.

Para encontrar j , note que en el caso presente $k^* = 3$ y $N = 6$, de manera que j debe buscarse dentro del conjunto $\{4, 5, 6, \}$.

Se prueba con $j = 6$. Como $a_6 = 2$ y $a_{k^*} = a_3 = 4$, se tiene que $a_j = a_6 = 2$ **no** es mayor que $a_{k^*} = a_3 = 4$. Por lo tanto, $j < 6$.

Se prueba con $j = 5$. Debido a que $a_5 = 5$ y $a_{k^*} = a_3 = 4$, se tiene que $a_j = a_5 = 5$ **si** es mayor que $a_{k^*} = a_3 = 4$. Por lo tanto, $j = 5$.

(b) Intercambiar a_j y a_{k^*} para obtener la permutación **b**:

$$\mathbf{a} = (3, 1, \underbrace{4}_{k^*=3}, 6, \underbrace{5}_{j=5}, 2)$$

$$\mathbf{b} = (3, 1, \underbrace{5}, 6, \underbrace{4}, 2).$$

Note que las componentes de la nueva permutación **b** se encuentran en orden descendente, como se anticipó en el Teorema 6.1.

(c) En la permutación **b** las últimas $N - k^* = 6 - 3 = 3$ componentes se rearreglan para colocarlas en orden ascendente, obyeniendo la permutación **a***:

$$\mathbf{b} = (3, 1, 5, 6, \underbrace{4}, 2)$$

$$\mathbf{a}^* = (3, 1, 5, 2, \underbrace{4}, 6).$$

Por lo tanto, $\text{Suc}(\mathbf{a}) = \mathbf{a}^* = (3, 1, 5, 2, 4, 6)$. □

Observación 6.1. El procedimiento establecido en el Teorema 6.1 puede ser fácilmente codificado en algún lenguaje de alto nivel. Como en la Observación 5.1, a continuación se muestra una alternativa en *QBasic*, donde K denota al entero k^* :

(a) Dado N , para determinar j es suficiente el siguiente código:

$J = N$

DO WHILE ($a(J) < a(K)$)

$$J = J - 1$$

LOOP

Al final de este ciclo, el valor de J será el mayor entero tal que $a(J) > a(K)$.

(a) El intercambio de $a(J)$ y $a(K)$ se logra con

SWAP $a(J), a(K)$

(c) Debido a que al arribar a esta etapa las componentes del vector \mathbf{a} son descendentes a partir de la entrada $k^* + 1 = K + 1$, colocar éstas en orden ascendente se logra mediante el siguiente código:

$$I = K + 1 : \quad F = N$$

DO WHILE ($F > I$)

SWAP $a(I), a(F)$

$$I = I + 1 : \quad F = F - 1$$

LOOP

□

Implementación del Algoritmo

Después de determinar el miembro mínimo de la clase $\mathcal{P}(N, N)$, y de encontrar un procedimiento para construir el sucesor de una permutación dada, es posible enunciar un algoritmo para listar la familia completa de todas las permutaciones de la población \mathcal{P} . Desde luego, dicho algoritmo es un caso particular de aquél establecido en la Sección 5 del Capítulo 3 para enumerar un conjunto arbitrario. El enunciado preciso en el caso presente es como sigue:

Fase 1: Defina $K = 1$ y $\mathbf{a}_1 = (1, 2, 3, \dots, N)$.

Fase 2: Determine si \mathbf{a}_k tiene un sucesor.

Fase 3: (a) Si \mathbf{a}_k tiene un sucesor, encuentre $\mathbf{b} = \text{Suc}(\mathbf{a})$. Incremente K en una unidad y defina $\mathbf{a}_K = \mathbf{b}$.

(b) Si \mathbf{a}_k no tiene sucesor vaya a la Fase 4.

Fase 4: Se ha concluido la enumeración de todos los miembros de la familia $\mathcal{P}(N, N)$; la lista completa es $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K$, de manera que K es el número de miembros de $\mathcal{P}(N, N)$.

La Codificación. El algoritmo precedente puede ser fácilmente codificado en algún lenguaje de programación. Desde luego, las peculiaridades de la implementación dependerán de la plataforma que se utilice, y a continuación se muestra una forma de hacerlo en el lenguaje *QBasic*:

'Programa: Listado de la Familia de Permutaciones P(N,N)

DECLARE FUNCTION Kstar%(a() AS INTEGER)

DECLARE SUB Sucesor (K AS INTEGER, a() AS INTEGER)

DECLARE SUB Imprimir (a() AS INTEGER)

OPTION BASE 1

DIM N AS INTEGER, I AS INTEGER, K AS INTEGER

COLOR 7, 1

CLS

' *****

PRINT "Generación de las permutaciones de los primeros N enteros positivos."

DO

 INPUT "Introduzca el valor de N (entero mayor o igual a 2): N =", N

LOOP WHILE ((N ≤ 1) OR (N <> INT(N)))

DIM a(N) AS INTEGER

FOR I = 1 TO N

 a(I) = I

NEXT

K = N

DO WHILE (K <> 0)

 CALL Imprimir(a())

 K = Kstar(a())

 IF (K > 0) THEN CALL Sucesor(K, a())

LOOP

END

’ *****

SUB Imprimir (a() AS INTEGER)

STATIC Count AS LONG

DIM I AS INTEGER, U AS INTEGER

U = UBOUND(a)

Count = Count + 1

PRINT Count; ". (";

FOR I = 1 TO U

 PRINT a(I);

NEXT

PRINT ")”

END SUB

FUNCTION Kstar% (a() AS INTEGER)

DIM U AS INTEGER, K AS INTEGER

U = UBOUND(a)

K = U - 1

```

DO WHILE (K >= 1)
    IF (a(K) > a(K + 1)) THEN
        K = K - 1
    ELSE
        EXIT DO
    END IF
LOOP
Kstar = K
END FUNCTION

SUB Sucesor (K AS INTEGER, a() AS INTEGER)
DIM I AS INTEGER, J AS INTEGER, F AS INTEGER
U = UBOUND(a)
J = U
DO WHILE (a(J) < a(K))
    J = J - 1
LOOP
SWAP a(J), a(K)
I = K + 1: F = U
DO WHILE (F > I)
    SWAP a(I), a(F)
    I = I + 1
    F = F - 1
LOOP
END SUB

```

Comentarios. El programa precedente ha sido construido siguiendo las ideas expuestas en las Observaciones 6.1 y 6.2. En las primeras tres líneas se declara la

función denominada "Kstar", la cual acepta una permutación como parámetro, y devuelve el valor de k^* (vea (5.1)); la función retorna el valor cero si la permutación que se le pasa es la máxima. El procedimiento denominado "Sucesor" acepta un entero y una permutación como parámetros. Cuando el entero es el valor k^* que corresponde a la permutación, el subprograma genera el sucesor de la permutación que se le transmitió, colocando al sucesor en la misma ubicación en memoria que la permutación original.

Después de las declaraciones, aparece la sentencia 'OPTION BASE 1', la cual indica que las componentes de un vector (una permutación en el caso presente) se numeran empezando desde el número 1, y posteriormente, la sentencia "DIM" reserva espacio de memoria para N, el tamaño de la población, para K, variable que contendrá el valor del índice k^* para la permutación considerada, y para el contador I que se usa posteriormente en un ciclo 'FOR'. Las sentencias COLOR 7,1 y CLS, establecen que la salida se lleve a cabo en pantalla con fondo azul y letras blancas, y a limpiar la pantalla, respectivamente. El cuerpo principal del programa se ubica entre las líneas de asteriscos. La sentencia PRINT genera un mensaje breve que describe lo que el programa realiza, y a continuación se pide el tamaño de la población; esto se hace dentro de un ciclo "DO WHILE", hasta que el usuario introduce un tamaño válido para la población, el cual debe ser entero y mayor o igual a dos. A continuación se reserva espacio de memoria para un vector de tamaño N, el cual contendrá a la permutación que se analice en cada paso del programa; es ese momento, el vector "a" contiene cero en sus N componentes, y por esta razón se inicializa en el siguiente ciclo FOR, después del cual "a" es e vector $(1, 2, \dots, N)$. A partir de ese momento, se imprime la permutación actual, se evalúa el correspondiente entero k^* usando la función "Kstar", y entonces se calcula el sucesor correspondiente a través del subprograma "Sucesor". Todo esto se hace una y otra vez en el programa por medio de un ciclo DO WHILE, esto es, la condición de entrada al ciclo se ubica al inicio del ciclo. Dicha condición es que K—

el valor calculado de k^* para la penúltima de las permutaciones consideradas—sea no nulo, pues el valor cero de K indica que se ha generado la permutación máxima. Note que debido a que la entrada al ciclo implica una prueba, el valor de K se inicializa antes de entrar al ciclo por primera vez.

Una muestra de la Salida del Programa. Cuando el programa de listado se ejecuta con $N = 4$ se obtiene la siguiente lista del conjunto $\mathcal{P}(4, 4)$, el cual consiste de todas las permutaciones de tamaño cuatro de la población $\mathcal{P} = \{1, 2, 3, 4\}$. Note que el orden en que se generan las permutaciones se indica por un número en negritas a la izquierda de cada miembro de $\mathcal{P}(4, 4)$. Desde luego, el programa escribe la permutación $(1, 2, 3, 4)$ en primer lugar y $(4, 3, 2, 1)$ al final, pues estas permutaciones son la mínima y la máxima respectivamente. Es instructivo ver como se genera una permutación determinada a partir de la precedente. Por ejemplo, considere la generación de la permutación número 17 a partir de la 16.

1	(1, 2, 3, 4)	2	(1, 2, 4, 3)	3	(1, 3, 2, 4)
4	(1, 3, 4, 2)	5	(1, 4, 2, 3)	6	(1, 4, 3, 2)
7	(2, 1, 3, 4)	8	(2, 1, 4, 3)	9	(2, 3, 1, 4)
10	(2, 3, 4, 1)	11	(2, 4, 1, 3)	12	(2, 4, 3, 1)
13	(3, 1, 2, 4)	14	(3, 1, 4, 2)	15	(3, 2, 1, 4)
16	(3, 2, 4, 1)	17	(3, 4, 1, 2)	18	(3, 4, 2, 1)
19	(4, 1, 2, 3)	20	(4, 1, 3, 2)	21	(4, 2, 1, 3)
22	(4, 2, 3, 1)	23	(4, 3, 1, 2)	24	(4, 3, 2, 1)

La permutación generada con el número 16 es

$$\mathbf{a} = (3, 2, 4, 1),$$

y después de escribirla, se llama a la función "Kstar" con el 'parámetro' \mathbf{a} . Dentro de "Kstar" se determina el valor del índice k^* para \mathbf{a} , obteniendo $k^* = 2$. Luego, se llama a la función "Sucesor", con los parámetros $K = k^* = 2$ y \mathbf{a} . Dentro de este subprograma se analiza a 'de derecha a izquierda', buscando el primer índice j para el cual $a_j > a_{k^*} = a_2 = 2$. Este entero resulta ser $j = 3$, provocando que se intercambien las componentes k^* y j de \mathbf{a} , generando la permutación

$$\mathbf{b} = (3, 4, 2, 1).$$

A continuación, se invierte el orden en que aparecen todos los miembros de \mathbf{b} a partir de la entrada $k^* + 1 = 3$. Esto provoca que se genere la sucesión

$$\mathbf{a}^* = (3, 4, 1, 2),$$

la cual ocupa la misma ubicación que originalmente tenía \mathbf{a} ; por supuesto, \mathbf{a}^* es el sucesor de \mathbf{a} y se imprime en el siguiente ciclo. \square

Conclusión

En este capítulo se ha construido un algoritmo para producir una lista de todas las permutaciones de la población \mathcal{P} que consta de los primeros N enteros positivos. El algoritmo presentado en la Sección 7 es una implementación especial del procedimiento general de listado de un conjunto formulado en el Capítulo 3, y fue obtenido después de analizar la construcción del sucesor de una permutación cuando la clase $\mathcal{P}(N, N)$ se considera ordenada a través del orden lexicográfico. Aunque el estudio realizado en las Secciones 3–6 fue, en ocasiones, complejo, el algoritmo que se generó tiene una estructura simple que conduce a una codificación sencilla.

Capítulo 5

Generación de Subconjuntos

Este capítulo persigue el siguiente propósito: Formular un algoritmo para generar una lista de todos los subconjuntos de un tamaño determinado que pueden extraerse de una población. Para alcanzar este objetivo, primero se identifica a los subconjuntos con un tipo especial de vectores en \mathbb{R}^r , y el algoritmo de generación se obtiene analizando a la familia de subconjuntos como una clase lexicográficamente ordenada. El análisis se basa en el procedimiento general de listado establecido en el Capítulo 3.

Introducción

Sea $\mathcal{P} = \{1, 2, \dots, N\}$ el conjunto de los primeros N enteros positivos, donde se supone que $N \geq 2$. Para esta población, la familia $\mathcal{S}(r, N)$ se define por medio de la igualdad

$$\mathcal{S}(r, N) = \{S \mid S \subset \mathcal{P}, \text{ y } S \text{ tiene } r \text{ elementos}\},$$

esto es, los miembros de $\mathcal{S}(r, N)$ son subconjuntos de tamaño r de la población \mathcal{P} ; naturalmente, en esta expresión el entero r se ubica entre 1 y N . El número de miembros de $\mathcal{S}(r, N)$, a los que con frecuencia se les refiere como ‘*combinaciones de N en r* ’, o mediante alguna expresión similar, está dado por

$$\binom{N}{r} = \frac{N!}{r!(N-r)!},$$

expresión que se obtiene a partir de principios fundamentales de conteo; vea, por ejemplo, Dudewicz y Mishra (1988), Liu (1978), Hall(1971). Sin embargo, determinar el número de miembros de una familia es un problema muy distinto a generar

todos los elementos de una forma sistemática, problema que, para la clase $\mathcal{S}(r, N)$, constituye el tema de este capítulo.

Para generar una lista completa de los miembros de $\mathcal{S}(r, N)$, se seguirá la estrategia sugerida por la teoría general expuesta en el Capítulo 3, la cual culminó estableciendo un algoritmo general de listado. Para avanzar en esta dirección, es necesario disponer de un orden total en la clase $\mathcal{S}(r, N)$, y debido a que el único resultado general que se tiene sobre existencia de órdenes totales se refiere a la ordenación lexicográfica de los espacios \mathbb{R}^p , es conveniente observar que cada subconjunto de tamaño r de la población \mathcal{P} puede *identificarse* naturalmente con un vector r -dimensional. Dicha identificación se obtiene de la siguiente manera: Con un subconjunto $S = \{s_1, s_2, \dots, s_r\}$ asocie el vector que se obtiene colocando los miembros de S en orden ascendente. Por ejemplo, si $S = \{7, 2, 9\}$, entonces el vector de \mathbb{R}^3 que corresponde a S es $(2, 7, 9)$. Debido a que un subconjunto no se altera cuando sus miembros se encierran entre llaves de manera distinta, —por ejemplo, $\{7, 2, 9\} = \{9, 2, 7\} = \{2, 7, 9\}$ —esta correspondencia entre subconjuntos y vectores es biunívoca, y se empleará consistentemente en el resto del capítulo, de manera que se pensará en la clase $\mathcal{S}(r, N)$ como un conjunto de vectores de dimensión r con componentes enteras, las cuales se incrementan al recorrer el vector de izquierda a derecha, esto es, cuando crece el índice de las componentes.

La presentación del material ha sido organizada de la siguiente manera: En la Sección 2 se destaca la importancia que en la implementación práctica de las técnicas estadísticas tiene la enumeración de subconjuntos, mientras que en la Sección 3 se determinan los elementos mínimo y máximo de la familia $\mathcal{S}(r, N)$ respecto al orden lexicográfico. La Sección 4 trata sobre el *índice de ruptura* de una pareja de conjuntos, y sobre el rol que esta idea desempeña en el problema de comparar dos conjuntos posteriores a uno dado. En la Sección 5 se caracteriza al sucesor (inmediato) de un conjunto en términos del valor máximo de un índice de ruptura y de la minimización de una familia de sucesores, mientras que en la

Sección 6 se establece un procedimiento, cuyo rasgo fundamental es la sencillez, para generar la lista de todos los miembros de $\mathcal{S}(r, N)$, y se ejemplifica su codificación en una forma estructurada. Finalmente, la exposición culmina en la Sección 7 con algunos comentarios breves.

El Contexto Estadístico

Antes de iniciar el análisis del problema fundamental de este capítulo, y ahondar en sus aspectos sutiles, es conveniente reforzar la motivación para recorrer el trayecto hasta el final. Como se mencionó en el Capítulo 2, el problema de generación de subconjuntos de tamaño dado, se encuentra ‘oculto’ tras la belleza de la teoría asintótica de las funciones estimables. El problema práctico de calcular, explícitamente, el valor que un estadístico de tipo U asume a partir de un conjunto de datos muestrales, es de enorme trascendencia, e invariablemente requiere enumerar todos los subconjuntos de tamaño r de la población $\mathcal{P} = \{1, 2, 3, \dots, N\}$, donde N es el tamaño de la muestra, y r es el grado del estadístico.

Ejemplo 2.1. Considere el problema de obtener un indicador para las diferencias existentes en el nivel de ingreso en una región. Dados los ingresos X_1, X_2, X_3 de tres personas seleccionadas al azar, un posible indicador se obtiene de la siguiente manera:

(i) Obtenga los estadísticos de orden Y_1, Y_2, Y_3 correspondientes a la muestra X_1, X_2, X_3 ;

(ii) Calcule el cociente

$$h(X_1, X_2, X_3) = \frac{Y_3 - Y_1}{(Y_1 + Y_3)/2},$$

el cual es la diferencia máxima de ingreso que se observa en la muestra, *relativa* al ingreso medio observado.

(iii) Como índice de las diferencias de ingreso, se toma a

$$\gamma = E[h(X_1, X_2, X_3)] = E \left[\frac{2(Y_3 - Y_1)}{Y_1 + Y_3} \right].$$

En este caso, suponiendo que la distribución común de las variables X_i es continua y tiene soporte en el conjunto (a, ∞) donde $a > 0$ es fijo y conocido, el mejor estimador de γ es el estadístico de tipo U dado por

$$G = \frac{1}{\binom{N}{3}} \sum_S h(X_{s_1}, X_{s_2}, X_{s_3}),$$

donde la sumatoria se extiende sobre todos los subconjuntos de tamaño 3 de $\{1, 2, \dots, N\}$ (Meddis, 1984), Randles y Wolfe ,1981). Luego, un paso imprescindible para calcular el valor asumido por G , es producir una lista de la familia $\mathcal{S}(3, N)$, para pasar cada uno de estos subconjutos $\{s_1, s_2, s_3\}$ a una rutina que calcule el valor asumido por $h(X_{s_1}, X_{s_2}, X_{s_3})$. Este argumento muestra que el problema de listar los miembros de una clase $\mathcal{S}(r, N)$ tiene gran relevancia en las aplicaciones, y consituye, además del enorme interés propio que tiene el problema, la motivación fundamental para el desarrollo del capítulo.

Los Subconjuntos Máximo y Mínimo

En esta sección se inicia el trayecto hacia la construcción de un algoritmo para generar una lista de todos los miembros de la familia $\mathcal{S}(r, N)$, la cual consta de todos los subconjuntos de tamaño r de la población $\mathcal{P} = \{1, 2, \dots, N\}$. Para analizar este problema, se seguirá una estrategia similar a la empleada para listar todas las permutaciones de tamaño N , esto es, el enfoque que se sigue no es otra cosa que una implementación del procedimiento general de listado enunciado en el Capítulo 3.

Antes de continuar, es oportuno recordar la convención establecida en la Introducción de este capítulo, de acuerdo a la cual *los miembros de un subconjunto* $\mathbf{a} \in \mathcal{S}(r, N)$ *se listan en orden creciente*, de manera que \mathbf{a} se identifica, de manera natural, con una permutación de tamaño r en la cual sus componentes se incrementan al avanzar de izquierda a derecha, esto es,

$$\mathbf{a} = (a_1, a_2, \dots, a_r), \quad 1 \leq a_1 < a_2 < \dots < a_r \leq N. \quad (3.1)$$

Luego, $\mathcal{S}(r, N)$ es un subconjunto de \mathbb{R}^r , espacio en el cual el orden lexicográfico es completo. De conformidad al procedimiento establecido en la Sección 5 del Capítulo 3, la primera condición para generar una lista de los miembros de $\mathcal{S}(r, N)$ es identificar su elemento mínimo. Para alcanzar este propósito, primero se establecerá el siguiente resultado preliminar, el cual desempeña un papel fundamental en el desarrollo de esta capítulo.

Teorema 3.1. Para cada $\mathbf{a} \in \mathcal{S}(r, N)$, se tiene que

$$i \leq a_i \leq N - r + i, \quad i = 1, 2, \dots, r. \quad (3.2)$$

Demostración. Debido a que las componentes a_i son enteros positivos que se incrementan conforme crece el índice i , se tiene que

$$a_1 \geq 1, \quad (3.3)$$

y $a_k - a_{k-1} \geq 1$ siempre que $k > 1$, de tal forma que cuando $i > 1$,

$$\begin{aligned} a_i - a_1 &= \underbrace{(a_i - a_{i-1}) + (a_{i-1} - a_{i-2}) + \dots + (a_2 - a_1)}_{i-1 \text{ términos}} \\ &\geq \underbrace{1 + 1 + \dots + 1}_{i-1 \text{ términos}} \\ &= (i - 1). \end{aligned}$$

Por lo tanto, $a_i \geq a_1 + (i - 1) \geq 1 + (i - 1) = i$ cuando $i > 1$, y combinando este hecho con (3.3) se concluye que $a_i \geq i$ para todo $i = 1, 2, \dots, r$. Para establecer la segunda desigualdad en (3.2), note que

$$a_r \leq N, \quad (3.4)$$

pues todos los miembros de \mathbf{a} pertenecen a la población $\mathcal{P} = \{1, 2, \dots, N\}$; vea (3.1). Observe ahora que para cada entero positivo $i < r$

$$\begin{aligned} a_r - a_i &= \underbrace{(a_r - a_{r-1}) + (a_{r-1} - a_{r-2}) + \dots + (a_{i+1} - a_i)}_{r-i \text{ términos}} \\ &\geq \underbrace{1 + 1 + \dots + 1}_{r-i \text{ términos}} \\ &= (r - i), \end{aligned}$$

donde de nueva cuenta se ha utilizado que $a_k - a_{k-1} \geq 1$. Luego, $a_r - a_i \geq r - i$, lo cual equivale a

$$a_i \leq a_r - (r - i),$$

y combinando esta desigualdad con (3.4) se concluye que $a_i \leq N - (r - i) = N - r + i$ para todo $i = 1, 2, \dots, r$, estableciendo la segunda desigualdad en (3.2). \square

Usando el teorema anterior, no es difícil determinar los elementos máximo y mínimo de la familia $\mathcal{S}(r, N)$.

Proposición 3.1. Si la familia $\mathcal{S}(r, N)$ se ordena lexicográficamente, entonces las siguientes afirmaciones (i) y (ii) son válidas.

(i) El elemento mínimo \mathbf{m} de $\mathcal{S}(r, N)$ se obtiene colocando los primero r enteros positivos en orden ascendente:

$$\mathbf{m} = (1, 2, \dots, r - 1, r).$$

Similaremente,

(ii) El elemento máximo de la familia $\mathcal{S}(r, N)$ se genera colocando los r enteros mayores en la población \mathcal{P} en orden creciente, es decir,

$$\mathbf{M} = (N - r + 1, N - r + 2, \dots, N - 1, N).$$

Demostración. (i) Para comprobar la afirmación, es suficiente verificar que para $\mathbf{a} \in \mathcal{S}(r, N)$ el siguiente enunciado es válido:

$$\mathbf{a} \neq \mathbf{m} \implies \mathbf{m} \prec \mathbf{a}.$$

Suponga que $\mathbf{a} = (a_1, a_2, \dots, a_r)$ es diferente de \mathbf{m} , de manera que $a_s \neq m_s = s$ para algún entero s . Ahora, sea k la primera posición en que difieren las componentes de \mathbf{a} y \mathbf{m} , i.e., $k \in \{1, 2, \dots, r\}$ satisface

$$a_i = m_i = i \quad \text{si } i < k, \quad \text{y} \quad a_k \neq m_k = k.$$

Para concluir, note que $a_k \geq k$, por el Teorema 3.1, de manera que $a_k \neq m_k = k$, implica que $a_k > k = m_k$. En resumen, en la primera posición en que \mathbf{a} y \mathbf{m} difieren, se tiene que la coordenada correspondiente de \mathbf{a} es mayor que la de \mathbf{m} , y entonces $\mathbf{m} \prec \mathbf{a}$.

(ii) Para verificar que \mathbf{M} es el máximo elemento de la familia $\mathcal{S}(r, N)$ se comprobará que

$$\mathbf{a} \neq \mathbf{M} \implies \mathbf{a} \prec \mathbf{M}.$$

Con esta finalidad, suponga que $\mathbf{a} = (a_1, a_2, \dots, a_r) \in \mathcal{S}(r, N)$ es diferente de \mathbf{M} y, como en el argumento precedente, sea k la primera posición en que las componentes de \mathbf{a} y \mathbf{M} difieren. i.e., $1 \leq k \leq r$ satisface

$$a_i = M_i = i \quad \text{si } i < k, \text{ mientras que } \quad a_k \neq M_k = N - r + k.$$

Observando que $a_k \leq N - r + k$ (vea (3.2)), a partir de la condición $a_k \neq M_k = N - r + k$ se obtiene que $a_k < M_k$, mostrando que en la primera posición en la que \mathbf{a} y \mathbf{M} difieren, la componente de \mathbf{M} es mayor que la de \mathbf{a} . Por lo tanto, $\mathbf{a} \prec \mathbf{M}$ siempre que $\mathbf{a} \neq \mathbf{M}$. \square

Sucesores e Índices de Ruptura

Para implementar el algoritmo general de listado en el caso de la familia $\mathcal{S}(r, N)$, es necesario diseñar un procedimiento para determinar el sucesor de un conjunto $\mathbf{a} \in \mathcal{S}(r, N) \setminus \{\mathbf{M}\}$. El primer paso hacia este objetivo se realiza en esta sección, y consiste en estudiar la primera posición en que dos conjuntos distintos en $\mathcal{S}(r, N)$ difieren. De manera similar al caso de permutaciones de rango máximo, el *índice de ruptura* de una pareja (\mathbf{a}, \mathbf{b}) se define como el entero k que satisface

$$a_i = b_i, \quad \text{para } 1 \leq i < k, \quad \text{y} \quad a_k \neq b_k.$$

vea la Definición 4.1 en el Capítulo 4. Observe, además, que para cualquier pareja (\mathbf{a}, \mathbf{b}) de miembros distintos de $\mathcal{S}(r, N)$, el correspondiente índice de ruptura es

un entero entre 1 y r , inclusive. Dado $\mathbf{a} \in \mathcal{S}(r, N)$, el objetivo particular de esta sección es explorar la posibilidad de comparar dos sucesores de un conjunto \mathbf{a} en términos de su índice de ruptura. Para alcanzar este propósito, es conveniente introducir la siguiente notación.

Definición 4.1. Sea $\mathbf{a} \in \mathcal{S}(r, N)$ un subconjunto fijo y diferente de \mathbf{M} . Para cada entero positivo $k = 1, 2, \dots, r$, la clase $\mathcal{R}_k(\mathbf{a})$ consiste de todos los subconjuntos $\mathbf{b} \in \mathcal{S}(r, N)$ que satisfacen

$$\mathbf{a} \prec \mathbf{b} \quad \text{y el índice de ruptura de la pareja } (\mathbf{a}, \mathbf{b}) \text{ es } k.$$

Ejemplo 4.1. Considere el caso $N = 7$ y $r = 3$, y sea $\mathbf{a} = (3, 4, 6)$. Para este subconjunto, se tiene que las tres clases $\mathcal{R}_1(\mathbf{a})$, $\mathcal{R}_2(\mathbf{a})$, $\mathcal{R}_3(\mathbf{a})$ son diferentes del vacío. En efecto,

(a) $\mathbf{b} = (4, 6, 7)$ es distinto de \mathbf{a} , y la primera componente en que \mathbf{a} y \mathbf{b} difieren es la número uno, y entonces el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) es $k = 1$; más aún, $a_1 = 3 < 4 = b_1$, de tal forma que $\mathbf{a} \prec \mathbf{b}$, y entonces $\mathbf{b} \in \mathcal{R}_1(\mathbf{a}) \neq \emptyset$.

(b) Sea $\mathbf{b} = (3, 5, 6)$, el cual es diferente de \mathbf{a} . Puesto que $a_1 = b_1 = 3$ y $a_2 = 4 < 5 = b_2$, se desprende que el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) es $k = 2$, y que $\mathbf{a} \prec \mathbf{b}$. Luego, $\mathbf{b} \in \mathcal{R}_2(\mathbf{a}) \neq \emptyset$.

(c) Defina $\mathbf{b} = (3, 4, 7)$. Este miembro de $\mathcal{S}(3, 7)$ es distinto de \mathbf{a} ; de hecho, la única componente en que \mathbf{a} y \mathbf{b} difieren es en la tercera ($a_3 = 6 < 7 = b_3$). Por lo tanto, $\mathbf{a} \prec \mathbf{b}$ y el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) es tres, lo cual significa que $\mathbf{b} \in \mathcal{R}_3(\mathbf{a}) \neq \emptyset$. □

Ejemplo 4.2. Como antes, considere el caso $N = 7$ y $r = 3$, pero defina al subconjunto \mathbf{a} de manera ligeramente distinta:

$$\mathbf{a} = (3, 4, 7).$$

A continuación se verificará que $\mathcal{R}_3(\mathbf{a}) = \emptyset$, es decir, \mathbf{a} no admite un sucesor \mathbf{b} tal que el índice de ruptura con \mathbf{a} sea tres. Para verificar esta afirmación se utilizará el método de contradicción: Suponga que $\mathbf{b} \in \mathcal{R}_3(\mathbf{a})$. En este caso,

$$\mathbf{a} \prec \mathbf{b}, \quad (4.1)$$

y

$$a_1 = b_1, \quad a_2 = b_2, \quad \text{y} \quad a_3 \neq b_3,$$

pues el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) es tres. Esta última relación combinada con (4.1) implica que $a_3 < b_3$, la cual en el caso presente se reduce $7 < b_3$; sin embargo, esto es una contradicción, pues las componente de \mathbf{b} deben ser enteros menores o iguales a 7 (recuerde que $N = 7$). Por lo tanto el supuesto de que $\mathcal{R}_3(\mathbf{a}) \neq \emptyset$ no puede ser verdadero, y se concluye que $\mathcal{R}_3(\mathbf{a}) = \emptyset$. \square

El resultado principal de esta sección es una versión del Lema 5.1 enunciado en el Capítulo 4, y proporciona un criterio directo para comparar dos sucesores de una conjunto \mathbf{a} por medio de sus índices de ruptura cuando éstos son diferentes

Lema 4.1. Sean \mathbf{b} y \mathbf{d} dos subconjuntos tales que $\mathbf{b} \succ \mathbf{a}$ y $\mathbf{d} \succ \mathbf{a}$. Si $\mathbf{d} \in \mathcal{R}_{k_1}(\mathbf{a})$ y $\mathbf{b} \in \mathcal{R}_k(\mathbf{a})$ donde $k_1 > k$, entonces

$$\mathbf{b} \succ \mathbf{d}$$

En otras palabras, este lema establece que si dos conjuntos \mathbf{b} y \mathbf{d} suceden a un conjunto determinado \mathbf{a} , entonces ‘el menor’ de \mathbf{b} y \mathbf{d} es el conjunto cuyo índice de ruptura con \mathbf{a} es *mayor*.

Demostración del Lema 4.1. Sean \mathbf{b} y \mathbf{d} dos conjuntos, donde $\mathbf{b} \in \mathcal{R}_k(\mathbf{a})$ y $\mathbf{d} \in \mathcal{R}_{k_1}(\mathbf{a})$, y $k < k_1$. En este caso,

$$a_1 = b_1, \quad a_2 = b_2 \quad \cdots \quad a_{k-1} = b_{k-1} \quad a_k < b_k$$

$$a_1 = d_1, \quad a_2 = d_2 \quad \cdots \quad a_{k-1} = d_{k-1} \quad a_k = d_k \quad \cdots \quad a_{k_1-1} = d_{k_1-1}, \quad a_{k_1} < d_{k_1}$$

Puesto que $k < k_1$, se sigue que

$$b_i = d_i, \quad i < k, \quad b_k > d_k,$$

de manera que $\mathbf{b} \succ \mathbf{d}$. □

Ejemplo 4.3. Considere el caso $N = 7$ y $r = 3$, y sea $\mathbf{a} = (3, 4, 6)$. Cada una de los siguientes dos subconjuntos $(4, 6, 7)$ y $(3, 5, 6)$ es un sucesor de \mathbf{a} . En efecto,

- $(4, 6, 7)$ y $\mathbf{a} = (3, 4, 6)$ difieren en la primera componente, y $a_1 = 3 < 4$, de manera que $\mathbf{a} \prec (4, 6, 7)$, y $(4, 6, 7) \in \mathcal{R}_1(\mathbf{a})$.
- $(3, 5, 6)$ y $\mathbf{a} = (3, 4, 6)$ coinciden en la primera coordenada, pero difieren en la segunda componente; debido a que $a_2 = 4 < 5$, se concluye que $\mathbf{a} \prec (3, 5, 6)$ y $(3, 5, 6) \in \mathcal{R}_2(\mathbf{a})$.

De acuerdo al Lema 4.1, se concluye, sin consideración adicional alguna, que

$$(3, 5, 6) \prec (4, 6, 7),$$

pues entre dos sucesores de \mathbf{a} con índice de ruptura distinto, el ‘menor’ de ellos es aquél para el que el índice es mayor. □

Caracterización del Sucesor

El propósito de esta sección es obtener una caracterización del sucesor de un conjunto $\mathbf{a} \in \mathcal{S}(r, N)$ cuando $\mathbf{a} \neq \mathbf{M}$. El resultado principal es el siguiente.

Teorema 5.1. [El sucesor de un conjunto.] Sea $\mathbf{a} \in \mathcal{S}(r, N) \setminus \{\mathbf{M}\}$ un conjunto arbitrario, y defina el entero k^* y el conjunto \mathbf{a}^* mediante

$$k^* = \max\{k \mid 1 \leq k \leq r, \quad \text{y} \quad \mathcal{R}_k(\mathbf{a}) \neq \emptyset\}, \quad (5.1)$$

mientras que

$$\mathbf{a}^* = \min \mathcal{R}_{k^*}(\mathbf{a}). \quad (5.2)$$

En este caso, \mathbf{a}^* es el sucesor de \mathbf{a} , esto es,

$$\mathbf{a} \prec \mathbf{a}^*, \quad \text{y para todo } \mathbf{b} \in \mathcal{S}(r, N) \quad [\mathbf{a} \prec \mathbf{b} \Rightarrow \mathbf{a}^* \preceq \mathbf{b}].$$

Demostración. Primero observe que $\mathbf{a}^* \in \mathcal{R}_{k^*}(\mathbf{a})$ implica que $\mathbf{a} \prec \mathbf{a}^*$, pues $\mathcal{R}_{k^*}(\mathbf{a})$ consiste sólo de conjuntos que suceden a \mathbf{a} . Sea $\mathbf{b} \in \mathcal{S}(r, N)$ tal que $\mathbf{a} \prec \mathbf{b}$. En este caso, $\mathbf{b} \in \mathcal{R}_k(\mathbf{a})$ para algún entero k . En particular, $\mathcal{R}_k(\mathbf{a}) \neq \emptyset$, y entonces $k \leq k^*$ (vea (5.1)). Considere ahora las siguientes dos alternativas, las cuales son las únicas que pueden presentarse al comparar k y k^* :

Caso 1: $k < k^*$.

En esta situación usando que $\mathbf{a}^* \in \mathcal{R}_{k^*}(\mathbf{a})$, el Lema 4.1 implica que $\mathbf{a}^* \prec \mathbf{b}$; recuerde que entre dos sucesores de \mathbf{a} , ‘el menor’ es aquél cuyo índice de ruptura con \mathbf{a} es mayor.

Caso 2: $k = k^*$.

En estas circunstancias, $\mathbf{b} \in \mathcal{R}_{k^*}(\mathbf{a})$, y entonces $\mathbf{a}^* \preceq \mathbf{b}$, pues que \mathbf{a}^* es el elemento mínimo de $\mathcal{R}_{k^*}(\mathbf{a})$; vea (5.2).

En resumen, se ha mostrado que $\mathbf{a} \prec \mathbf{a}^*$, y que para todo conjunto \mathbf{b} que sucede a \mathbf{a} , esto es, $\mathbf{a} \prec \mathbf{b}$, se tiene que $\mathbf{a}^* \preceq \mathbf{b}$. Por lo tanto, \mathbf{a}^* es el sucesor de \mathbf{a} . \square

Cálculo del Sucesor de un Conjunto

El Teorema 5.1 caracteriza al sucesor de un subconjunto $\mathbf{a} \in \mathcal{S}(r, N)$. Sin embargo, el uso efectivo de ese resultado para determinar explícitamente al sucesor de un conjunto determinado en $\mathcal{S}(r, N) \setminus \{\mathbf{M}\}$ requiere, ante todo, de un método ‘práctico’ para determinar k^* in (5.1), así como \mathbf{a}^* , el elemento mínimo de la clase $\mathcal{R}_{k^*}(\mathbf{a})$ (vea (5.2)), problemas que se abordan en esta sección. Primero se analiza la determinación de k^* , y se establece una condición necesaria y suficiente para que una clase $\mathcal{R}_k(\mathbf{a})$ sea no vacía; este resultado se establece a continuación y es

sustancialmente más sencillo que su contraparte para permutaciones presentada en la Proposición 4.2 del Capítulo 4.

Proposición 6.1. Para un entero $k < N$ y un conjunto $\mathbf{a} \in \mathcal{S}(r, N)$, representado mediante

$$\mathbf{a} = (a_1, a_2, \dots, a_r), \quad 1 \leq a_1 < a_2 < \dots < a_r \leq N,$$

las siguientes afirmaciones (a) y (b) son equivalentes:

(a) La clase $\mathcal{R}_k(\mathbf{a})$ es no vacía;

(b) $a_k < N - r + k$.

Para interpretar este resultado, recuerde que siempre se tiene que la k -ésima componente de \mathbf{a} es menor o igual a $N - r + k$, esto es, $a_k \leq N - r + k$, por el Teorema 3.1. Usando este hecho, la Proposición 6.1 puede expresarse verbalmente como sigue: Una clase $\mathcal{R}_k(\mathbf{a})$ es no vacía, si y sólo si a_k es menor que el máximo valor que puede asumir.

Demostración de la Proposición 6.1 [(a) \Rightarrow (b)]. Suponga que la clase $\mathcal{R}_k(\mathbf{a})$ es no vacía, y seleccione $\mathbf{b} \in \mathcal{R}_k(\mathbf{a})$. Por la definición de $\mathcal{R}_k(\mathbf{a})$ se tiene que $\mathbf{a} \prec \mathbf{b}$ y la primera componente en que \mathbf{a} y \mathbf{b} difieren es la k -ésima, es decir,

$$a_i = b_i, \quad i < k, \quad \text{and} \quad a_k < b_k \tag{6.1}$$

Usando ahora el Teorema 3.1 se sigue que $b_k \leq N - r + k$, de manera que esta desigualdad, combinada con la desigualdad en el extremo derecho de (6.1), implica que $a_k < b_k \leq N - r + k$, y entonces $a_k < N - r + k$.

[(b) \Rightarrow (a)]. Suponga que $a_k < N - r + k$. En estas circunstancias, defina el subconjunto $\mathbf{b} = (b_1, b_2, \dots, b_r)$ como sigue:

$$b_i = \begin{cases} a_i, & \text{si } 1 \leq i < k, \\ 1 + a_k, & \text{si } i = k, \\ b_k + (i - k) & \text{si } r \geq i > k. \end{cases} \tag{6.2}$$

A continuación se verificará que \mathbf{b} es un miembro legítimo de $\mathcal{S}(r, N)$, esto es, que las componentes de \mathbf{b} son crecientes y se ubican entre 1 y N (vea (3.1)). Debido a que las componentes de \mathbf{a} crecen y son positivas, la definición de \mathbf{b} implica que sus componentes también tienen estas propiedades, así que es suficiente con verificar que $b_r \leq N$. Para ver que esta desigualdad es válida, note que la condición $a_k < N - r + k$ implica que $b_k = a_k + 1 \leq N - r + k$, de manera que

$$b_r = b_k + (r - k) \leq N - r + k + (r - k) = N,$$

y por lo tanto \mathbf{b} en (6.2) es un miembro de $\mathcal{S}(r, N)$. Por otro lado, note que (6.2) implica que

$$b_i = a_i, \quad \text{si } i < k, \quad \text{y} \quad b_k = a_k + 1 > a_k,$$

y entonces el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) es k y $\mathbf{a} \prec \mathbf{b}$, esto es, $\mathbf{b} \in \mathcal{R}_k(\mathbf{a})$. Este argumento ha mostrado que $a_k < N - r + k \Rightarrow \mathcal{R}_k(\mathbf{a}) \neq \emptyset$, y concluye la demostración de la Proposición 6.1. \square

Ejemplo 6.1. En los Ejemplos 4.1 y 4.2, se consideraron casos en los que se determinó directamente si algunas clases $\mathcal{R}_k(\mathbf{a})$ son no vacías. Ahora se analizarán esos ejemplos, pero utilizando la Proposición 6.1. Considere el contexto determinado por

$$N = 7, \quad r = 3.$$

(a) Suponga que $\mathbf{a} = (3, 4, 6)$. En esta situación,

$$a_1 = 3 < N - r + 1 = 7 - 3 + 1 = 5. \quad \text{Luego, } \mathcal{R}_1(\mathbf{a}) \neq \emptyset$$

$$a_2 = 4 < N - r + 2 = 7 - 3 + 2 = 6. \quad \text{Por lo tanto, } \mathcal{R}_2(\mathbf{a}) \neq \emptyset$$

$$a_3 = 6 < N - r + 3 = 7 - 3 + 3 = 7. \quad \text{En consecuencia, } \mathcal{R}_3(\mathbf{a}) \neq \emptyset.$$

Como es claro a partir de este argumento, la Proposición 6.1 permite decidir, sin esfuerzo sustancial, si una clase $\mathcal{R}_k(\mathbf{a})$ es no vacía. Compare con el enfoque que se siguió en el Ejemplo 4.1.

(b) Ahora suponga que $\mathbf{a} = (3, 4, 7)$. En este caso,

$$a_3 = 7 = N - r + 3,$$

y la Proposición 6.1 implica, de inmediato, que $\mathcal{R}_3(\mathbf{a}) = \emptyset$; compare con el argumento empleado en el Ejemplo 4.2. \square

A continuación se presenta un procedimiento algorítmico que permite generar el índice k^* asociado a un conjunto $\mathbf{a} \in \mathcal{S}(r, N)$; más aún, no es necesario decidir inicialmente si $\mathbf{a} = \mathbf{M}$, pues el procedimiento detecta si esta posibilidad ocurre; observe la similitud con el procedimiento presentado en la Proposición 6.1 del Capítulo 4 referente al análisis de permutaciones.

Proposición 6.2. Sea $\mathbf{a} \in \mathcal{S}(r, N)$ un conjunto dado. El siguiente procedimiento permite determinar si $\mathbf{a} = \mathbf{M}$ o si $\mathbf{a} \prec \mathbf{M}$, y en este último caso, se obtiene el valor de k^* en (5.1).

Datos: Un subconjunto \mathbf{a} de la población $\mathcal{P} = \{1, 2, \dots, N\}$ donde el número de elementos de \mathbf{a} es r , y \mathbf{a} se escribe como en (3.1).

Inicio: Defina $K = r$.

[Bloque Iterativo]

Repetir Mientras $K \geq 1$

Si $a_K = N - r + K$ Entonces Disminuya K en una Unidad

Si $a_K < N - r + K$ Entonces Salga del Ciclo

Fin del Ciclo Repetir

Fin del Procedimiento:

Si $K = 0$ entonces $\mathbf{a} = \mathbf{M}$

Si $K > 0$ entonces $\mathbf{a} \prec \mathbf{M}$ y $k^* = K$.

Demostración. Suponga que al concluir el procedimiento se tiene que se tiene $K = 0$. La única forma en que esto puede ocurrir es que en los diversos pasos por el ciclo del algoritmo se disminuya K en una unidad, lo cual sólo acontece cuando todas las siguientes igualdades ocurren:

$$\begin{aligned} \underbrace{a_r = N - r + r = N}_{K=r}, \quad & \underbrace{a_{r-1} = N - r + (r - 1) = N - 1}_{K=r-1}, \quad \dots \\ \dots \quad & \underbrace{a_2 = N - r + 2}_{K=2}, \quad \underbrace{a_1 = N - r + 1}_{K=1} \end{aligned} \quad (6.3)$$

donde debajo de cada igualdad se indica el valor que K asume dentro del ciclo ‘Repetir’ cuando la ecuación se verifica. Después de comprobar que $a_1 = N - r + 1$ el valor de K se disminuye en una unidad, obteniendo $K = 0$, y al intentar entrar al ciclo la condición $K \geq 1$ no se satisface, de tal forma que el flujo pasa a la siguiente instrucción posterior al ciclo, en la cual se encuentra el final del procedimiento, y se anuncia que $K = 0$, y que \mathbf{a} es \mathbf{M} , el miembro máximo de $\mathcal{S}(r, N)$; las igualdades (6.3) muestran que, efectivamente, este es el caso. Para concluir, suponga que el valor de K al final del procedimiento es positivo. Esto significa que la salida del ciclo no se origina en que la condición de entrada se viole, sino en el hecho de que la igualdad $a_K = N - r + K$ no se satisface; puesto que siempre se verifica que $a_K \leq N - r + K$ (por el Teorema 3.1), se desprende que $a_K < N - r + K$, y entonces la Proposición 6.1 implica que $\mathcal{R}_K(\mathbf{a}) \neq \emptyset$. Finalmente, observe que en cada ejecución del ciclo el valor de K disminuye en una unidad si la condición $a_k = N - r + K$ se verifica, y que la salida se produce cuando se detecta que $a_K < N - r + K$ por primera vez. Esto significa que si al salir del ciclo se tiene que $K > 0$, entonces

$$\begin{aligned} a_r = N - r + r = N, \quad a_{r-1} = N - r + (r - 1) = N - 1, \quad \dots \\ \dots \quad a_{K+1} = N - r + (K + 1), \quad a_K < N - r + K; \end{aligned}$$

esta relación muestra que para cada un entero s que satisfaga $K < s \leq r$, siempre ocurre que $a_s = N - r + s$, de manera que $\mathcal{R}_s(\mathbf{a}) = \emptyset$, por la Proposición 6.1.

Consecuentemente,

$$\mathcal{R}_K(\mathbf{a}) \neq \emptyset, \quad \text{y} \quad \mathcal{R}_s(\mathbf{a}) = \emptyset, \quad \text{si } K < s \leq r,$$

de manera que $K = \max\{k \mid \mathcal{R}_k(\mathbf{a}) \neq \emptyset\}$, y entonces $K = k^*$; vea (5.1). \square

Ejemplo 6.2. Considere el caso $N = 9$ y $r = 5$. A continuación se ejemplificará el uso del procedimiento descrito en la Proposición 6.2 para determinar el valor de k^* para el conjunto

$$\mathbf{a} = (1, 4, 7, 8, 9).$$

Inicialmente, $K = 5$ ($= r$).

[Bloque Iterativo]

Al intentar entrar al ciclo se prueba la condición $K \geq 1$, la cual se satisface y se produce la entrada al ciclo.

Dentro del ciclo se comparan $a_K = a_5 = 9$ y $N - r + K = 9 - 5 + 5 = 9$

Como $a_5 = 9$ se disminuye K en 1 (ahora $K = 4$)

Se va a la entrada del ciclo.

Al intentar entrar al ciclo, se comprueba si la condición $K \geq 1$ es válida; como $K = 4$ se produce la entrada al ciclo.

Dentro del ciclo se comparan $a_K = a_4 = 8$ y $N - r + K = 9 - 5 + 4 = 8$.

Como $a_4 = 8$ se disminuye K en 1 (ahora $K = 3$)

Se va a la entrada del ciclo.

Al intentar entrar al ciclo, se comprueba si la condición $K \geq 1$ es válida; como $K = 3$ se produce la entrada al ciclo.

Dentro del ciclo se comparan $a_K = a_3 = 7$ y $N - r + K = 9 - 5 + 3 = 7$.

Como $a_3 = 7$ se disminuye K en 1 (ahora $K = 2$)

Se va a la entrada del ciclo.

Al intentar entrar al ciclo, se comprueba si la condición $K \geq 1$ es válida; como $K = 2$ se produce la entrada al ciclo.

Dentro del ciclo se comparan $a_K = a_2 = 4$ y $N - r + K = 9 - 5 + 2 = 6$.

Como $a_2 < 6$ se produce la salida del ciclo.

Se arriba al final del procedimiento.

Como $K = 2$ es positivo, se concluye que $\mathbf{a} \prec \mathbf{M}$ y que el correspondiente valor de k^* en (5.1) está dado por $k^* = K = 2$.

Por supuesto, esta conclusión es clara a partir de la forma establecida para el conjunto \mathbf{a} , pero lo que se pretende es ilustrar el algoritmo presentado en la Proposición 6.2. □

Observación 6.1. El procedimiento descrito en la Proposición 6.2 puede codificarse fácilmente en un lenguaje de programación. Una posibilidad en *QBasic* se presenta a continuación:

```

K = R
DO WHILE (K >= 1)
  IF (a(K) = N - R + K) THEN
    K = K - 1
  ELSE
    EXIT DO
  END IF
LOOP

```

Suponga ahora que $\mathbf{a} \neq \mathbf{M}$ es un conjunto determinado, y que el correspondiente índice k^* ha sido evaluado. En la siguiente proposición se presenta un procedimiento computacional que genera el elemento mínimo de $\mathcal{R}_{k^*}(\mathbf{a})$.

Proposición 6.3. Sea $\mathbf{a} \neq \mathbf{M}$ un miembro de $\mathcal{S}(r, N)$, y sea k^* como en (5.1). El elemento mínimo de $\mathcal{R}_{k^*}(\mathbf{a})$ se obtiene a través del siguiente procedimiento:

(a) Inicie con el conjunto $\mathbf{a} = (a_1, a_2, \dots, a_r)$.

(b) Construya el conjunto \mathbf{b} como sigue:

$$\begin{aligned} b_i &= a_i \quad \text{para todo } i < k^*, \\ b_{k^*} &= a_{k^*} + 1 \\ b_s &= b_{k^*} + (s - k^*) \quad \text{para todo } s > k^*. \end{aligned} \tag{6.4}$$

Entonces, $\mathbf{b} = \min \mathcal{R}_{k^*}(\mathbf{a})$ y, consecuentemente, $\mathbf{b} = \mathbf{Suc}(\mathbf{a})$.

Antes de demostrar la validez de la Proposición 6.3, se ilustrará la construcción del sucesor de un conjunto determinado.

Ejemplo 6.3. Suponga que $N = 9$ y $r = 5$, y considere el conjunto

$$\mathbf{a} = (1, 4, 7, 8, 9),$$

el cual fue objeto de estudio en el Ejemplo 6.2, donde se determinó que el correspondiente entero k^* es $k^* = 2$. El conjunto \mathbf{b} en (6.4) se construye como sigue: Para $i < k^* = 2$, $b_i = a_i$. Por lo tanto, $b_1 = a_1 = 1$; en este momento

$$\mathbf{b} = (1, \quad , \quad , \quad , \quad).$$

A continuación la componente número $k^* = 2$ se determina mediante $b_{k^*} = b_2 = a_2 + 1 = 5$; ahora

$$\mathbf{b} = (1, 5, \quad , \quad , \quad).$$

Para $s > k^* = 2$, $b_s = b_{k^*} + (s - k^*)$, de manera que $b_3 = b_2 + (3 - 2) = 5 + 1 = 6$, $b_4 = b_2 + (4 - 2) = 5 + 2 = 7$, y $b_5 = b_2 + (5 - 2) = 5 + 3 = 8$. Por lo tanto,

$$\mathbf{b} = (1, 5, 6, 7, 8)$$

es el sucesor de \mathbf{a} . □

Demostración de la Proposición 6.3. Primero observe que el conjunto \mathbf{b} especificado en (6.4) pertenece a \mathcal{R}_{k^*} , pues la primera componente en la que \mathbf{a} y

\mathbf{b} difieren es la número k^* (i.e., el índice de ruptura de la pareja (\mathbf{a}, \mathbf{b}) es k^*) y $b_{k^*} = a_{k^*} + 1 > a_{k^*}$, lo cual implica que $\mathbf{a} \prec \mathbf{b}$. A continuación se mostrará que \mathbf{b} es el mínimo de $\mathcal{R}_{k^*}(\mathbf{a})$. Para verificar esta afirmación, suponga que $\mathbf{c} \in \mathcal{R}_{k^*}(\mathbf{a})$, y observe que en este caso

$$c_i = a_i, \quad \text{para } i < k^*, \quad \text{y} \quad c_{k^*} > a_{k^*}. \quad (6.5)$$

La desigualdad en esta expresión implica que $c_{k^*} \geq a_{k^*} + 1$, y la definición de \mathbf{b} en (6.4) combinada con (6.5) implica que

$$c_i = b_i, \quad \text{para } i < k^*, \quad \text{y} \quad c_{k^*} \geq b_{k^*} (= a_{k^*} + 1). \quad (6.6)$$

Debido a que las componentes de \mathbf{c} se incrementan conforme el índice aumenta, para $k^* < s \leq r$ se tiene que

$$\begin{aligned} c_i - c_{k^*} &= \underbrace{(c_i - c_{i-1}) + (c_{i-1} - c_{i-2}) + \cdots + (c_{k^*+1} - c_{k^*})}_{i - k^* \text{ términos}} \\ &\geq \underbrace{1 + 1 + \cdots + 1}_{i - k^* \text{ términos}} \\ &= (i - k^*) \end{aligned}$$

y entonces

$$c_i \geq c_{k^*} + (i - k^*) \geq b_{k^*} + (i - k^*) = b_i, \quad i > k^*. \quad (6.7)$$

Las relaciones (6.6) y (6.7) implican que si $\mathbf{c} \in \mathcal{R}_{k^*}(\mathbf{a})$ es diferente de \mathbf{b} , entonces (i) la primera componente en la que \mathbf{b} y \mathbf{c} difieren corresponde a un índice $i \geq k^*$, y (ii) si $c_i \neq b_i$, entonces $c_i > b_i$. Por lo tanto, $\mathbf{c} \succ \mathbf{b}$, probando que $\mathbf{b} = \min \mathcal{R}_{k^*}(\mathbf{a})$, y por lo tanto $\mathbf{b} = \mathbf{a}^* = \text{Suc}(\mathbf{a})$; vea el Teorema 5.1. \square

Observación 6.2. El procedimiento establecido en el Teorema 6.1 puede ser fácilmente codificado en algún lenguaje de alto nivel. Como en la Observación 6.1, a continuación se muestra una alternativa en *QBasic*, donde K denota al entero k^* , y R es el número de elementos en el subconjunto \mathbf{a} . El siguiente código modifica las componente de \mathbf{a} a partir de K -ésima de acuerdo a (6.4).

```

b = a(K) + 1
FOR Count = K TO R
    a(Count) = b + (Count - K)
NEXT

```

Implementación del Algoritmo

Después de determinar el miembro mínimo de la clase $\mathcal{S}(r, N)$, y de encontrar un procedimiento para construir el sucesor de un conjunto dado, es posible enunciar un algoritmo para listar la familia completa de todos los subconjuntos de tamaño r de la población \mathcal{P} . Por supuesto, dicho algoritmo es un caso particular de aquél establecido en la Sección 5 del Capítulo 3 para enumerar un conjunto arbitrario. El enunciado preciso en el caso presente es como sigue:

Fase 1: Defina $C = 1$ y $\mathbf{a}_1 = (1, 2, 3, \dots, r)$.

Fase 2: Determine si \mathbf{a}_C tiene un sucesor.

Fase 3: (a) Si \mathbf{a}_C tiene un sucesor, encuentre $\mathbf{b} = \mathbf{Suc}(\mathbf{a}_C)$. Incremente C en una unidad y defina $\mathbf{a}_C = \mathbf{b}$.

(b) Si \mathbf{a}_C no tiene sucesor vaya a la Fase 4.

Fase 4: Se ha concluido la enumeración de todos los miembros de la familia $\mathcal{S}(r, N)$; la lista completa es $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_C$, de manera que C es el número de miembros de $\mathcal{S}(r, N)$.

La Codificación. El algoritmo precedente ha sido codificado en el lenguaje *QBasic*, aunque puede implementarse fácilmente en cualquier otro lenguaje. El Código, obtenido a partir de las Observaciones 6.1 y 6.2, es el siguiente:

' Programa para generar los subconjunto de tanaño R de una población con N elementos.

```

DECLARE FUNCTION SKstar% (a() AS INTEGER)

```

```

DECLARE SUB SSucesor (K AS INTEGER, a() AS INTEGER)
DECLARE SUB SImprimir (a() AS INTEGER)
OPTION BASE 1
DIM N AS INTEGER, R AS INTEGER, I AS INTEGER, K AS INTEGER
COLOR 7, 1
CLS
PRINT "Generación de los subconjunto de tamaño R de una población con N
elementos"
DO
    INPUT "Introduzca el valor de N (entero mayor o igual a 2): N = ", N
LOOP WHILE ((N <= 1) OR (N <> INT(N)))
DO
    INPUT "Introduzca el valor de R (entero mmenor igual a N): R = ", R
LOOP WHILE ((R < 1) OR (R <> INT(R)) OR (R > N))
    , *****
DIM a(R) AS INTEGER
FOR I = 1 TO R
    a(I) = I
NEXT
K = R
DO WHILE (K > 0)
    CALL SImprimir(a())
    K = SKstar(a())
    IF (K > 0) THEN CALL SSucesor(K, a())
LOOP
END
    , *****

```

```

SUB SImprimir (a() AS INTEGER)

STATIC Count AS LONG
DIM I AS INTEGER, U AS INTEGER

```

```
U = UBOUND(a)
```

```
Count = Count + 1
```

```
PRINT Count; ". (";
```

```
FOR I = 1 TO U
```

```
    PRINT a(I);
```

```
    IF (I < U) THEN PRINT ", ";
```

```
NEXT
```

```
PRINT ")"
```

```
END SUB
```

```
FUNCTION SKstar% (a() AS INTEGER)
```

```
DIM U AS INTEGER, K AS INTEGER
```

```
SHARED N AS INTEGER
```

```
R = UBOUND(a)
```

```
K = R
```

```
DO WHILE (K >= 1)
```

```
    IF (a(K) = N - R + K) THEN
```

```
        K = K - 1
```

```
    ELSE
```

```
        EXIT DO
```

```
    END IF
```

```
LOOP
```

```
SKstar% = K
```

```
END FUNCTION
```

```
SUB SSucesor (K AS INTEGER, a() AS INTEGER)
```

```
DIM R AS INTEGER, Count AS INTEGER
```

```
R = UBOUND(a)
```

```
b = a(K) + 1
```

```
FOR Count = K TO R
```

```
    a(Count) = b + (Count - K)
```

```
NEXT
```

```
END SUB
```

Comentarios. El programa precedente ha sido construido siguiendo las ideas expuestas en las Observaciones 6.1 y 6.2. En las primeras tres líneas se declaran los subprogramas y funciones que se usan en la estructura principal:

- La función denominada “SKstar”, la cual acepta un conjunto como parámetro, y devuelve el valor de k^* correspondiente al conjunto transmitido (vea (5.1)); la función retorna el valor cero si el conjunto sobre el que actúa es el máximo.
- El procedimiento denominado “SSucesor” acepta un entero y un conjunto como parámetros. Cuando el entero es el valor k^* que corresponde al conjunto, el subprograma genera el sucesor del conjunto transmitido, colocando al sucesor en la misma ubicación en memoria que el conjunto que se le transfirió.
- El subprograma denominado “SImprimir” lleva a la pantalla el más reciente de los subconjuntos generados.

Después de las declaraciones, aparece la sentencia ‘OPTION BASE 1’, la cual indica que las componentes de un vector (un subconjunto en el caso presente) se numeran empezando desde el número 1, y posteriormente, la sentencia ”DIM” reserva espacio de memoria para N, el tamaño de la población, para K, variable

que contendrá el valor del índice k^* para el conjunto considerado, R el tamaño de los subconjuntos, y para el contador I que se usa posteriormente en un ciclo 'FOR'. Las sentencias COLOR 7,1 y CLS, tienen la misma utilidad que en el Capítulo 4. La sentencia PRINT genera un mensaje breve que describe lo que el programa realiza, y a continuación se pide el tamaño de la población; esto se hace dentro de un ciclo "DO WHILE", hasta que el usuario introduce un tamaño válido para la población, el cual debe ser entero y mayor o igual a dos. En el siguiente ciclo "DO WHILE" se solicita al usuario que introduzca el tamaño de los subconjuntos considerados, repitiendo la petición hasta que se produzca la entrada de un entero entre 1 y N . Después de la introducción de datos, el cuerpo principal del programa se indica entre líneas de astriscos: La sentencia DIM reserva espacio de memoria para un vector de tamaño R , el cual contendrá al subconjunto que se analice en cada paso del programa; es ese momento, el vector "a" contiene cero en sus R componentes, y por esta razón se inicializa en el ciclo FOR, después del cual "a" es e vector $(1, 2, \dots, R)$.

A partir de ese momento, se imprime el conjunto actual llamando al subprograma "SImprimir", se calcula el correspondiente entero k^* para el conjunto mediante la función "SKstar", y entonces se determina el sucesor correspondiente a través del subprograma "SSucesor" si el valor calculado de k^* es positivo. Todo esto se hace una y otra vez en el programa por medio de una estructura DO WHILE, en la cual la condición de entrada ubica en la entrada del ciclo. Dicha condición es que K —el valor calculado de k^* para el más reciente de los conjuntos considerados—sea no nulo, pues si el valor K devuelto por la función "SKstar" es nulo, entonces el máximo subconjunto ya ha sido generado. Como la entrada al ciclo implica una prueba, el valor de K se inicializa (con el valor $R > 0$) antes de intentar el ingreso al ciclo por primera vez .

Una Muestra de la Salida del Programa. Cuando el programa de listado se ejecuta con $N = 8$ y $r = 3$ se obtiene la siguiente lista de los miembros de la familia

$\mathcal{S}(3, 8)$, el cual consiste de todos los subconjuntos de tamaño tres de la población $\mathcal{P} = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Note que el orden en que se generan los conjuntos se indica por un número en negritas a la izquierda de cada miembro de $\mathcal{S}(8, 3)$.

1	(1, 2, 3)	2	(1, 2, 4)	3	(1, 2, 5)	4	(1, 2, 6)
5	(1, 2, 7)	6	(1, 2, 8)	7	(1, 3, 4)	8	(1, 3, 5)
9	(1, 3, 6)	10	(1, 3, 7)	11	(1, 3, 8)	12	(1, 4, 5)
13	(1, 4, 6)	14	(1, 4, 7)	15	(1, 4, 8)	16	(1, 5, 6)
17	(1, 5, 7)	18	(1, 5, 8)	19	(1, 6, 7)	20	(1, 6, 8)
21	(1, 7, 8)	22	(2, 3, 4)	23	(2, 3, 5)	24	(2, 3, 6)
25	(2, 3, 7)	26	(2, 3, 8)	27	(2, 4, 5)	28	(2, 4, 6)
29	(2, 4, 7)	30	(2, 4, 8)	31	(2, 5, 6)	32	(2, 5, 7)
33	(2, 5, 8)	34	(2, 6, 7)	35	(2, 6, 8)	36	(2, 7, 8)
37	(3, 4, 5)	38	(3, 4, 6)	39	(3, 4, 7)	40	(3, 4, 8)
41	(3, 5, 6)	42	(3, 5, 7)	43	(3, 5, 8)	44	(3, 6, 7)
45	(3, 6, 8)	46	(3, 7, 8)	47	(4, 5, 6)	48	(4, 5, 7)
49	(4, 5, 8)	50	(4, 6, 7)	51	(4, 6, 8)	52	(4, 7, 8)
53	(5, 6, 7)	54	(5, 6, 8)	55	(5, 7, 8)	56	(6, 7, 8)

Desde luego, el programa escribe el conjunto (1, 2, 3) en primer lugar, y a partir de ese momento genera el sucesor del último conjunto generado, hasta llegar al conjunto máximo (6, 7, 8). Es instructivo ver como se genera un conjunto determinado a partir del precedente. Por ejemplo, considere la generación del conjunto 34 a partir del 33.

- El conjunto 33 es (2, 5, 8), para el cual $k^* = 2$. Puesto que $a_2 = 5$, la segunda componente se cambia por 6(= $a_2 + 1$) y las componentes con índice mayor a 2 se obtienen agregando uno a la precedente. Al realizar estos cambios, se obtiene

$$\mathbf{b} = (2, \underbrace{6}_{a_5+1}, \underbrace{7}_{a_5+2}),$$

conjunto que, en su carácter de sucesor de (2, 5, 8), aparece en la lista en el lugar 34.

Conclusión

En este capítulo se ha presentado un algoritmo que permite generar *todos* los subconjuntos de tamaño r de una población con N elementos. La idea básica es

la interpretación de un subconjunto como un vector con componentes crecientes, de tal forma que una clase de subconjuntos está contenida en \mathbb{R}^r , espacio en el que la ordenación lexicográfica es total. Después de notar esta característica, la ruta que condujo hacia el algoritmo siguió el camino trazado en la teoría general desarrollada en el Capítulo tres. La estructura del procedimiento que se obtuvo es similar a la del algoritmo de generación de permutaciones presentado anteriormente, pero es oportuno enfatizar el siguiente aspecto: Dado un conjunto fijo \mathbf{a} , la determinación del valor máximo que puede asumir el índice de ruptura de \mathbf{a} con otro conjunto \mathbf{b} representa un problema sustancialmente más simple que la búsqueda de k^* en el caso de permutaciones, y conociendo este valor, la construcción del sucesor de \mathbf{a} se obtiene de forma directa a través de la definición (6.4).

Finalmente, como se mencionó al final del Capítulo 2, la enumeración de todas las permutaciones de tamaño r de la población \mathcal{P} , se reduce a aplicar, sucesivamente, los procedimientos de generación de los miembros de las familias $\mathcal{S}(r, N)$, y $\mathcal{P}(r, r)$, esto es, un método de cómputo para permutaciones de tamaño menor al de la población se obtiene directamente a partir de los programas enunciados en este y en el capítulo precedente.

Capítulo 5

Retrospectiva

En este trabajo se ha estudiado un problema fundamental para la implementación y análisis de métodos noparamétricos, a saber, la construcción de *algoritmos computacionales para la generación de listas de las familias de permutaciones y subconjuntos*.

El punto de partida para el diseño de los algoritmos de enumeración computacional, fue el desarrollo de un método general de listado de un conjunto abstracto, el cual muestra la equivalencia entre los problemas de enumerar los miembros de un conjunto, y formular un orden total en el mismo; cuando un conjunto se encuentra totalmente ordenado, producir una lista de sus miembros se reduce a analizar dos características del orden establecido:

- (i) Encontrar el elemento mínimo del conjunto \mathcal{A} bajo consideración, y
- (ii) Determinar la función **Suc**, la cual produce el sucesor de cada miembro de \mathcal{A} siempre que éste exista, lo cual efectivamente ocurre cuando el elemento analizado no es el máximo.

Después de comprender los aspectos esenciales de un problema de listado, el paso que condujo al desarrollo exitoso de los algoritmos propuestos, fue la introducción del orden lexicográfico en los espacios Euclidianos; dicho orden tiene la propiedad de ser completo, esto es, permite comparar, sin ambigüedad alguna, dos vectores de la misma dimensión con componentes reales. De esta forma, para un conjunto \mathcal{A} contenido en \mathbb{R}^r , listar sus elementos se reduce a implementar el algoritmo general de listado, para lo cual se requiere determinar la correspondiente función ‘Sucesor’ y el elemento mínimo de \mathcal{A} ; estos dos últimos aspectos deben analizarse en cada caso particular.

La determinación de los elementos mínimos de las familias de permutaciones y subconjuntos fue relativamente simple, en contraste con la especificación de la función 'Sucesor', la cual requirió, especialmente en el caso de las permutaciones, en análisis más complejo. En cualquier caso, independientemente de las dificultades encontradas en el trayecto, el resultado final fue un algoritmo simple de listado, que puede ser codificado sin mayor dificultad en un lenguaje computacional de alto nivel.

Literatura Citada

- Dolciani, M. P. , E. Beckenbach. E. Reilich (1986), *Álgebra II, Publicaciones Cultural*, 1976, México. D.F.
- Dudewicz, E., y N. Mishra, (1988), *Modern Mathematical Statistics, Wiley*, New York.
- Even, S. (1973), *Algorithmic Combinatorics, MacMillan*, New York.
- Fisher, L. y J. MacDonald (1978), *Fixed Effects Analysis of variance, Academic Press*, New York.
- Graybill, F. A. (1985), *Theory and Application of the Linear Model, 1st. Edition, Wadsworth*, Belmont, CA
- Hall, Jr. M. (1971), *Combinatorial Theory, Blaisdell Publishing Co.*, London.
- Hollander, M. y D. A. Wolfe (1981), *Nonparametric Statistical Methods, Wiley*, New York
- Hoeffding, W.(1948), A class of statistics with asytmotically normal distribution, *Annals of Mathematical Statistics*, **19**, 199–325.
- Johnson, S. M. (1963), Generation of permutations by adjacent transpositions, *Mathematics of Computation*, **3**, 282–285.
- Lehmann, E. L. (1971), *Testing Statistical Hypothesis, Wiley*, N.Y.
- Lehmann, E. L. (1975), *Nonparametrics: Statistical Methods Based on Ramks, Holden-Day*, San Francisco.
- Liu, C. L. (1978), *Introduction to Combinatorial Mathematics, McGraw-Hill*, New York.
- Meddis, R. (1984), *Statistics Using Rank, Basil Blackwell*, Ney York.
- Mood, A. M., F. A. Graybill y D. C. Boes (1985), *Introduction to the Theory of Statistics, McGraw-Hill*, New York.
- Munkres, J. R, (1975), *Topology, A First Course, Prentice-Hall*, Englewood Cliffs,

New Jersey.

Papadopoulos, J. S. (1997), *Calculate PI, ABCode*, **2**, June, 1997.

Randles, R. H. y D. A. Wolfe (1981), *Introduction to the Theory of Nonparametric Statistics*, *Wiley*, New York.

Saucedo E., J. M. (1999), *Límites y Normas de Paro en Control Estadístico de Calidad*, Tesis de Maestría en Ciencias, Subdirección de Postgrado, Universidad Autónoma Agraria Antonio Narro, Saltillo, Coahuila, México.

Searle, S. R. (1979), *Linear Models*, *Wiley*, New York.

Serfling, R. J. (1980), *Approximation Theorems of Mathematical Statistics*, 3rd. Edition, *McGraw-Hill*, New York.

Suppes, P. (1971), *Set Theory*, *Van-Nostrand*, New York.