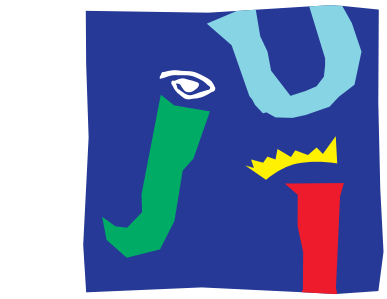

UNIVERSITAT JAUME·I

Detección de Rasgos en Imágenes con Ruido: Una Aproximación con Funciones LISA en Procesos Puntuales Espaciales

Tesis Doctoral

Presentada por: Gil Lorenzo Valentín

Dirigida por: Jorge Mateu Mahiques

Departamento de Matemáticas
Universitat Jaume I de Castellón

Marzo 2005

D. Jorge Mateu Mahiques, Profesor Titular de Universidad del Departamento de Matemáticas de la Universitat Jaume I de Castellón

CERTIFICA: que la presente memoria, *Detección de Rasgos en Imágenes con Ruido: Una Aproximación con Funciones LISA en Procesos Puntuales Espaciales*, ha sido realizada bajo su dirección en el Departamento de Matemáticas por D. Gil Lorenzo Valentín y constituye su Tesis para optar al Grado de Doctor en Ciencias Matemáticas.

Y para que conste, en cumplimiento de la legislación vigente, la presenta ante el Departamento de Matemáticas de la Universitat Jaume I de Castellón, a 21 de Marzo de 2005.

EL Director

Fdo.: Jorge Mateu Mahiques

Agradecimientos

El presente trabajo se ha realizado durante los cursos 2001-2005 en el Departamento de Matemáticas, de la Universitat Jaume I de Castellón. Mi agradecimiento a él.

A mi director de tesis, el profesor Jorge Mateu, por demostrarme una vez más, que insistiendo en el trabajo se consiguen los resultados, por ser referente en lo que concierne a dedicación, constancia e ilusión por la tarea. Mi admiración y mi gratitud.

A quienes han estado cada día, durante este curso, conmigo en el seminario, compartiendo ratos de trabajo, de esfuerzo y también de risas. Mi recuerdo siempre.

A mis amigos y amigas, que en la sombra me han sostenido amablemente, activamente, animando cada decisión, respaldando con todos los medios posibles ésta y otras ilusiones que tengo. Mi incondicional sonrisa.

Al hospital materno infantil de St. Joan de Déu, de Barcelona y en especial a Joaquim Erra i Mas, por proporcionarme material de imágenes para las aplicaciones utilizadas. A Manuel Batalla, por sus aportaciones y comentarios como médico, al respecto.

A mi familia, en especial a mi padre, a mi madre y a mi hermana, por ser siempre motor, soporte y ánimo en todo lo que emprendo, gracias por confiar, más allá de lo razonable, siempre en mí y en mis posibilidades.

A mis sobrinos Marc y Jorge

Índice general

1. Introducción **21**

2. Fundamentos básicos de procesos puntuales espaciales homogéneos **27**
2.1. Introducción 28
2.2. Elementos probabilísticos básicos de un proceso puntual espacial . 28
2.3. Características de un proceso puntual espacial 31
2.3.1. Fundamentos en la teoría de procesos puntuales: características de segundo orden 31
2.3.2. Estadísticos para un patrón de un proceso puntual homogéneo 35
2.3.3. Interacción entre puntos. Procesos de Gibbs 39
2.3.4. Intensidad condicionada 40
2.3.5. Procesos puntuales de Markov 41
2.4. Ejemplos y modelos de procesos puntuales 42
2.4.1. Proceso homogéneo de Poisson 42
2.4.2. Proceso de Poisson inhomogéneo 43
2.4.3. Proceso de Cox 44
2.4.4. Proceso cluster de Poisson 46
2.4.5. Proceso de inhibición simple (SSI) 47
2.4.6. Proceso pairwise interaction 50
2.4.7. Proceso de area-interaction 56
2.4.8. Proceso shot-noise 58
2.4.9. Proceso Candy 59

3. Métodos de clustering jerárquico basados en modelos probabilísticos (MBHC) **61**
3.1. Introducción 62
3.2. Técnicas basadas en modelos Gaussianos y no Gaussianos 63

3.2.1. Elección del número de clusters 66
3.2.2. Estimación de los parámetros mediante el Gibbs sampler . 68
3.2.3. Estrategia para aplicar los métodos Gaussianos 69
3.3. Detección de rasgos en procesos puntuales espaciales mediante métodos MBHC 69
3.3.1. Modelos 70
3.3.2. Algoritmos EM y CEM 70
3.3.3. Estimación del parámetro de forma 72
3.3.4. Elección del número de clusters 73
3.3.5. Software MCLUST 74
3.4. Métodos basados en segmentaciones 75
3.4.1. Introducción 75
3.4.2. Estimación no paramétrica y máximo-verosimilitud de rasgos en procesos puntuales espaciales usando Teselaciones de Voronoï 75
3.4.3. Estimación Bayesiana y segmentación de un proceso puntual usando Teselaciones de Voronoï 77
3.5. Técnicas cluster basadas en distancias al K-ésimo vecino más próximo 78
3.5.1. Introducción 78
3.5.2. Modelización 79
3.6. Simulaciones y aplicaciones 84

4. Fundamentos teóricos sobre funciones LISA locales 95
4.1. Introducción 95
4.2. Funciones individuales versus globales 96
4.3. Distribución teórica 99

5. Detección de rasgos en presencia de ruido. Distancias entre funciones LISA individuales 105
5.1. Introducción 105
5.2. Estudio descriptivo de funciones individuales para un patrón puntual 107
5.3. Clasificaciones de rasgos en presencia de ruido. Ejemplos simulados 129
5.4. Buscando la cantidad de rasgo y ruido simulado 148
5.5. Clasificaciones de rasgos en presencia de ruido. Imágenes reales . . 151
5.6. Análisis comparativo entre el algoritmo EM y el algoritmo SEM . 154

6. Detección de rasgo en presencia de ruido. Análisis Cluster y Escalamiento multidimensional 167
6.1. Introducción 167
6.2. Distribución teórica de las funciones LISA 169
6.3. Análisis cluster jerárquico 171
6.4. Escalamiento multidimensional 172
6.5. Clasificaciones de rasgos en presencia de ruido. Ejemplos simulados 174
6.6. Clasificaciones de rasgos en presencia de ruido. Imágenes reales . . 187

7. Conclusiones y futuras investigaciones 205

8. APÉNDICE: Códigos Splus, R y Fortran 211

Índice de figuras

2.1. Realizaciones de procesos de Poisson homogéneos en el cuadrado unidad 43

2.2. Realizaciones de un proceso de Poisson inhomogéneo con función de intensidad $\lambda(s) = 60[2 + \sin(2\pi x) + \sin(2\pi y)]$ 44

2.3. Realizaciones de un proceso de Poisson inhomogéneo con función de intensidad $\lambda(s) = n + 20\sqrt{n}(x + y)$. 45

2.4. Realizaciones de procesos Matérn con número medio de padres $\lambda = 15$ y número medio de hijos por padre $\mu = 20$. *Primera fila:* El radio de la esfera que rodea cada padre es R=0.01; *Segunda fila:* El radio de la esfera que rodea cada padre es R=0.05; *Tercera fila:* El radio de la esfera que rodea cada padre es R=0.10. 47

2.5. Realizaciones de procesos Thomas con número medio de padres $\lambda = 15$ y número medio de hijos por padre $\mu = 20$. *Primera fila:* La varianza de la distribución Gaussiana que gobierna los desplazamientos de los hijos respecto a los padres es 0.01; *Segunda fila:* La varianza de la distribución Gaussiana que gobierna los desplazamientos de los hijos respecto a los padres es 0.05; *Tercera fila:* La varianza de la distribución Gaussiana que gobierna los desplazamientos de los hijos respecto a los padres es 0.10. 48

2.6. Realizaciones de procesos Neyman-Scott con número medio de padres $\lambda = 15$ y número medio de hijos por padre $\mu = 15$. *Primera fila:* El radio máximo del cluster aleatorio es R=0.01 y la función que genera los clusters aleatorios es uniforme en un disco de radio 0.05; *Segunda fila:* El radio máximo del cluster aleatorio es R=0.05 y la función que genera los clusters aleatorios es uniforme en un disco de radio 0.05; *Tercera fila:* El radio máximo del cluster aleatorio es R=0.05 y la función que genera los clusters aleatorios es uniforme en un disco de radio 0.10. 49
2.7. Realizaciones de un proceso simple de inhibición. 50
2.8. Configuración de un proceso Strauss con radio $R = 15$ en una región 256×256 y la correspondiente función de intensidad condicionada. Claros y sombras indican valores de la función de intensidad condicionada. 52
2.9. Ejemplos de las funciones de interacción de Diggle y de VSC. . . 56
2.10. Configuración de un proceso de area-interaction con radio $R = 15$ en una región cuadrada de 256×256 y el correspondiente exponente de la función de intensidad condicional. Notar que si $\gamma > 1$, las sombras más claras indican valores más altos de la función de intensidad condicional, e inversamente para $\gamma < 1$. 57

3.1. Datos simulados (ruido y rasgo) y retículo regular 30×30 sobre los datos 85
3.2. Cluster jerárquico (software MCLUST) sobre los datos simulados 86
3.3. Clasificación en rasgo y ruido obtenida por el método de cluster jerárquico en combinación con el algoritmo EM 87
3.4. Representación de los valores BIC para los datos simulados . . . 88
3.5. Histogramas para la distancia F_{D_K} con $K = 1, 3, 5, 10$ 89
3.6. Histogramas para la distancia de celda regular al tercer vecino más próximo 90
3.7. Histogramas para la distancia de celda regular al décimo vecino más próximo 90
3.8. Detección de un rasgo con distancias al 5^o vecino más próximo . 91
3.9. Detección de un rasgo con distancias al 10^o vecino más próximo . 91
3.10. Detección de un rasgo con distancias al 15^o vecino más próximo . 92
3.11. Detección de dos rasgos con distancias al 5^o vecino más próximo 92
3.12. Detección de dos rasgos con distancias al 10^o vecino más próximo 93

3.13. Detección de dos rasgos con distancias al 15^o vecino más próximo 93

4.1. Esquema de las componentes asociadas al cálculo de las funciones LISA 104

5.1. Simulación de rasgo cuadrado de 600 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 109

5.2. Simulación de rasgo cuadrado de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 112

5.3. Simulación de rasgo cuadrado de 400 puntos sobre ruido de 400 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 113

5.4. Simulación de rasgo forma L de 400 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 115

5.5. Simulación de rasgo forma L de 600 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 116

5.6. Simulación de rasgo forma L de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 117

5.7. Simulación de rasgo con forma de círculo de 274 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 118

5.8. Simulación de rasgo con forma de círculo de 246 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 120

5.9. Simulación de rasgo con forma de círculo de 615 puntos sobre ruido de 400 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 121

5.10. Simulación de dos rasgos, uno con forma cuadrada de 400 puntos y forma de círculo de 404 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 122

5.11. Simulación de dos rasgos, uno con forma cuadrada de 200 puntos y forma L de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. . . 123

5.12. Simulación de dos rasgos, uno con forma L de 600 y forma de círculo de 591 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. . . . 124

5.13. Simulación de dos rasgos, uno cuadrado de 200 puntos y forma de círculo de 505 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. . . 126

5.14. Simulación de dos rasgos, uno cuadrado de 400 puntos y forma de círculo de 600 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. . . 127

5.15. Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 200 puntos y forma de círculo de 216 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 128

5.16. Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 200 puntos y forma de círculo de 293 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Representaciones de funciones LISA individuales, (izquierda) rasgo cuadrado, (medio) rasgo forma L, (derecha) rasgo circular. . 130

5.17. Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 400 puntos y forma de círculo de 575 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback. 131

5.18. Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 400 puntos y forma de círculo de 602 puntos sobre ruido de 500 puntos en el cuadrado unidad.*Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Representaciones de funciones LISA individuales, (izquierda) rasgo cuadrado, (medio) rasgo forma L, (derecha) rasgo circular. . 132

5.19. Clasificación de un rasgo cuadrado de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 30 puntos en el rasgo simulado y 500 puntos en el ruido simulado. La distancia utilizada es L1. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 136

5.20. Clasificación de un rasgo forma L de 600 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es Kullback. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 138

5.21. Clasificación de un rasgo forma circular de 417 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 30 puntos en el rasgo simulado y 500 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 139

5.22. Clasificación de un rasgo forma circular de 432 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 140

5.23. Clasificación de un rasgo cuadrado de 400 puntos sobre ruido de 400 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 141

5.24. Clasificación de un rasgo forma L de 400 puntos sobre ruido de 500 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 143

5.25. Clasificación de dos rasgos. Uno cuadrado de 200 puntos y otro forma L de 200 puntos sobre ruido de 500 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 144

5.26. Clasificación de dos rasgos. Uno cuadrado de 400 puntos y otro forma L de 600 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 145

5.27. Clasificación de tres rasgos. Uno cuadrado de 100 puntos, otro forma L de 250 puntos y otro de forma circular de 416 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 146

5.28. Clasificación de tres rasgos. Uno cuadrado de 100 puntos, otro forma L de 250 puntos y otro de forma circular de 428 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 147

5.29. Clasificación de tres rasgos. Uno cuadrado de 300 puntos, otro forma L de 400 puntos y otro de forma circular de 434 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 149

5.30. Clasificación de tres rasgos. Uno cuadrado de 300 puntos, otro forma L de 400 puntos y otro de forma circular de 426 puntos sobre ruido de 500 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado. 150
5.31. Tejido conectivo de órgano humano 158
5.32. Clasificación de imagen médica de la Figura (5.31). *Primera fila:* (izquierda) salida del programa MATROX, (derecha) histograma de las distancias de las funciones LISA, utilizando L2. *Segunda fila:* (izquierda) rasgo clasificado, (derecha) ruido clasificado. *Tercera fila:* (izquierda) ruido simulado, (derecha) rasgo simulado. 159
5.33. Costa de California 160
5.34. Clasificación de las posiciones de los epicentros de los terremotos en California, utilizando funciones LISA y divergencia de Kullback. *Primera fila:* (izquierda) representación de los epicentros de los terremotos, en California, (derecha) histograma de las distancias de las funciones LISA, utilizando Divergencia de Kullback. *Segunda fila:* (izquierda) rasgo clasificado, (derecha) ruido clasificado. *Tercera fila:* (izquierda) ruido simulado, (derecha) rasgo simulado. 161
5.35. Rasgo cuadrado de 400 puntos sobre ruido de 400 puntos en el cuadrado unidad. 162
5.36. Tres rasgos, cuadrado, forma L y circular de 300, 400 y 434 puntos respectivamente, sobre ruido de 300 puntos en el cuadrado unidad. 162

6.1. Clasificación de un rasgo cuadrado de 50 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 176

6.2. Clasificación de un rasgo cuadrado de 50 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales , (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 177
6.3. Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 100 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 180
6.4. Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 100 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 181
6.5. Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 182
6.6. Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 183

6.7. Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.6), donde se han eliminado los valores de gran distancia. 190

6.8. Clasificación de un rasgo cuadrado de 150 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 191

6.9. Clasificación de un rasgo cuadrado de 150 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 192

6.10. Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.9), donde se han eliminado los valores de gran distancia. 193

6.11. Clasificación de un rasgo circular de 228 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 194

6.12. Clasificación de un rasgo circular de 228 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 195

6.13. Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.12), donde se han eliminado los valores de gran distancia. 196

6.14. Clasificación de dos rasgos, uno circular de 164 puntos, uno cuadrado de 50 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 197

6.15. Clasificación de un rasgo cuadrado de 50 puntos para la rama de ruido de la Figura (6.14), sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 198

6.16. Clasificación de dos rasgos, uno circular de 164 puntos y uno cuadrado de 50 puntos, sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del rasgo clasificado, (derecha) representación del ruido clasificado. 199

6.17. Clasificación de dos rasgos, uno circular de 164 puntos y uno cuadrado de 50 puntos, sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del rasgo clasificado (II), (derecha) representación del ruido clasificado (II). 200

6.18. Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.16), donde se han eliminado los valores de gran distancia. 201

6.19. Localizaciones espaciales reducidas a (0,1), de focos de incendios durante los años 2000-2001, en la Comunidad Valenciana (España). 201
6.20. Función de intensidad de focos de incendios durante los años 2000-2001, en la Comunidad Valenciana (España). 202
6.21. Clasificación de los focos de los incendios de la Comunidad Valenciana (España), durante los años 2000-2001, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 203
6.22. Clasificación de los focos de los incendios de la Comunidad Valenciana (España), durante los años 2000-2001, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado. 204

Capítulo 1

Introducción

Cada vez más, la digitalización de las imágenes se va introduciendo en nuestra vida diaria. La facilidad con la que se toman, se almacenan y se visualizan, en cualquiera de los formatos (pantalla o papel) y sobre todo, se distribuyen y se copian, hace que esta forma de entender la fotografía gane adeptos cada día y en muchos casos supla a la fotografía convencional. Pero al margen del caracter lúdico, esta técnica de capturar imágenes encierra una poderosa herramienta para cualquier estudio científico.

Ya hace décadas que se utilizan fotografías digitalizadas, sobre todo de satélites y son muchas las disciplinas que han trabajado y trabajan con ellas: en Biología, por ejemplo, permite conocer la localización de una especie concreta de árboles o identificar el trazado de un río subterráneo; en Geología, es útil para conocer localizaciones de minas antipersonales en un campo que haya sido utilizado en guerra o para detectar los epicentros de terremotos; en Medicina, mediante radiografías o resonancias magnéticas es posible conocer las alteraciones de un órgano; en Astronomía para clasificar cuerpos celestes por temperatura de superficie o intensidad de luz, etc... Estos y más ejemplos ilustran la gran utilidad de esta técnica.

En este contexto se enmarca el análisis de ciertos elementos o rasgos subyacentes a nubes de ruidos que los empañan. La detección de este tipo de rasgos es de especial interés y es una de las motivaciones de nuestro trabajo.

Todas estas imágenes se pueden analizar de forma directa. Sin embargo, la dimensión que nos va a interesar es de índole matemática, concretamente Estadística. Automáticamente la foto digitalizada pasa a ser un listado de valores, tantos como píxeles (puntos de luz) tiene la foto. De todos éstos, prestaremos atención a su localización, coordenada x, coordenada y y a la información que nos interese añadir a cada punto. Puede ser un código que indique la escala de grises o color, si es que tiene. Puede ser otra información que a priori interese según el problema que se vaya a estudiar.

Cuando la zona de la imagen sea blanca, consideraremos que no hay información, es decir, no hay puntos de interés que analizar. El resto constituyen el banco de datos para trabajar. En general, se puede llamar individuos a cada uno de estos puntos. Destacaremos aquellos que forman parte de los elementos a detectar y les llamaremos *rasgos* y los puntos restantes que no destacan elementos de interés *ruido*. La motivación inicial establece que lo interesante es aislar los puntos que formen el o los rasgos, de los que son ruido. Este problema ya se ha estudiado y así se ha reflejado en la literatura (McLachlan & Basford, 1988 [71]; Bensmail, Celeux, Raftery & Robert, 1994 [10]). Todos estos estudios se enmarcan dentro de la teoría de Procesos Puntuales.

Definimos un proceso puntual como un conjunto de puntos $\{s_i \in X, \ \ i = 1, ...n\}$ distribuidos irregularmente en una región de interés X y generados por algún mecanismo desconocido. Un proceso puntual es, por tanto, un patrón de puntos, que gráficamente se podría corresponder con una nube de puntos. Aunque en el Capítulo 2 se profundiza en la teoría de procesos puntuales, podemos analizar un poco más esta definición.

La *región de interés* X suele ser un subconjunto del plano, usualmente rectangular (aunque no necesariamente). Cierto que la realidad no lo es y los procesos puntuales pretenden resolver mayoritariamente problemas relacionados con ella, pero nos hemos de limitar a la imagen capturada de ésta, asumiendo que la proyección en el plano XY distorsiona *en cierta medida* la información que nos interesa.

El *mecanismo desconocido* que provocó la distribución de puntos ha sido la clave para abordar el problema. Hay un primer bloque de métodos, los basados en Modelos de Probabilidad de Cluster Jerárquico, que presuponen que las posiciones de los puntos de rasgo o ruido, siguen una distribución de probabilidad,

concretamente la distribución Normal Multivariante y los basados en distancias entre las localizaciones de los individuos, que prescinden de este supuesto.

Ambos abordan el tema del Análisis Cluster, método que trata de buscar grupos en datos, desde dos perspectivas diferentes. Un primer acercamiento al problema de detección de rasgos por el análisis de estos dos métodos, tanto a nivel teórico como su implementación práctica bajo simulaciones, queda reflejado en el Capítulo 3.

Posteriormente, siendo la base metodológica y de aplicación de esta tesis doctoral, nos hemos centrado en el análisis de las funciones LISA, que explican las dependencias locales de un patrón puntual.

Básicamente, este trabajo, retoma un problema de interés real, como es la detección de rasgos en imágenes donde aparecen puntos que no pertenecen al rasgo (ruido) y los pretendemos separar y clasificar. Las últimas aportaciones hechas en este campo, se dirigen a considerar ausencia del modelo de probabilidad que originó la distribución espacial de los puntos en la imagen. En este sentido, los trabajos de partida que tomamos son los de Banfield & Raftery (1992) [7]; Byers & Raftery, (1996) [18], (1997) [19], en los cuales, se utilizan distancias entre las localizaciones de puntos, exactamente al K-vecino más próximo y el algoritmo EM, para encontrar la marca con valores dicotómicos que se asignará a cada individuo a partir de estas distancias, siendo una alternativa rasgo y la otra ruido. De este método propuesto, utilizamos la estructura del algoritmo EM con una modificación, la introducción de un paso intermedio para obtener el valor de una probabilidad que ha de ser estimada, dada la ausencia de modelo de probabilidad de referencia.

Otros trabajos, que nos sirven como punto de partida, son los de las funciones LISA (Cressie & Brant, 1997 [27]; 2001 [28]), en los cuales se definen funciones locales que recogen características de segundo orden alrededor de cada individuo. Cuando se aplicó esta teoría, se hizo sobre grupos definidos y aislados previamente, definiéndose vectores cuyas componentes eran las funciones LISA evaluadas en radios diferentes. De este trabajo, tomamos la estructura teórica, pero no el campo de aplicación.

Uniendo estas dos ideas, ¿por qué no aplicar y calcular vectores de funciones LISA asociadas a cada individuo, que recojan características de segundo orden, por tanto de agregación del patrón puntual, calculando sobre estos vectores

distancias y sobre estas distancias utilizar el algoritmo SEM, de forma que las agrupaciones de vectores que clasifique, al hacerlos corresponder con los puntos originales, den como consecuencia la clasificación buscada en Rasgo y Ruido?

Este trabajo persigue dar respuesta a esta pregunta. Así, para cada punto o individuo, consideramos un vector, cuyas componentes son funciones que denominamos *función LISA* individual. Estas funciones se construyen con ciertas distancias entre los individuos. A partir de un radio, se construye un entorno circular, con una banda de anchura determinada y se computan los individuos que se encuentran en esta banda. Se realiza este proceso para una gran cantidad diferente de radios y con ello obtenemos todas las componentes de los vectores (Capítulo 4).

El objetivo es demostrar que el estudio y análisis de comportamientos individuales de funciones LISA, proporciona características globales de todo el patrón puntual. Concretamente, en nuestro caso, hacemos uso de esta característica para encontrar agrupaciones de puntos a partir de las correspondientes agrupaciones de funciones LISA.

Para ello proponemos dos métodos que, siendo en esencia parecidos, tienen fundamentos y actuaciones diferentes. El primero de ellos se basa en considerar la función LISA de cada uno de los individuos y calcular la función LISA teórica esperada de un punto cualquiera (ruido). Mediante el algoritmo Stochastic EM (aquel utilizado para datos faltantes, en ausencia de distribución de probabilidad para los individuos) y con la utilización de la distancia de la función LISA de cualquier punto a esta media teórica, se clasifican los puntos en ruido y rasgo (Capítulo 5). La novedad radica en utilizar las funciones LISA individuales para cada punto y a partir de ellas utilizar una modificación del algoritmo EM (McLachlan & Basford, 1988 [71]; Bensmail, Celeux, Raftery & Robert, 1994 [10]) introduciendo la parte estocástica, dada la ausencia de distribuciones de probabilidad en todas las variables del proceso. Para calcular las distancias entre las funciones, se utilizan las distancias L1, L2 y Kullback tanto en los ejemplos simulados como en los casos reales con los que se ha trabajado.

El segundo de los métodos consiste en calcular distancias entre funciones LISA individuales utilizando la distancia de Mahalanobis, caso particular de la divergencia de Kullback, cuando la matriz de varianzas-covarianzas entre dos individuos es simétrica. La diferencia entre los dos métodos estriba en que, ahora,

no calculamos la distancia a ningún valor esperado de una función LISA asociada a un punto de un proceso aleatorio (ruido), sino que lo hacemos entre las funciones LISA de los puntos reales de la imagen y todo ello, claro, con el mismo objetivo (Capítulo 6). El cálculo de esta distancia de Mahalanobis proporcionará una matriz de distancias simétrica, que llamaremos D, de dimensión nxn, donde n será la cantidad de individuos o puntos de la imagen. A partir de la obtención de esta matriz, aplicaremos Cluster Jerárquico tradicional y Escalamiento multidimensional, para encontrar los grupos o rasgos en el ruido. Ambos son dos métodos utilizados ampliamente en estadística y a partir de similaridades entre los individuos considerados (en nuestro caso distancias entre funciones LISA) van detectando grupos con características parecidas.

Esta tesis finaliza con una recopilación de todos los códigos utilizados para la elaboración de este trabajo, con el fin de servir de ayuda a futuros investigadores en este campo.

A principios del curso 2001, la idea expuesta comenzó a tomar forma y desde entonces hemos trabajado en ella. Las conclusiones que hemos obtenido se recogen en este trabajo, que pretende aportar una solución más, al problema que lo originó.

Capítulo 2

Fundamentos básicos de procesos puntuales espaciales homogéneos

Indice General

2.1. Introducción 28

2.2. Elementos probabilísticos básicos de un proceso puntual espacial 28

2.3. Características de un proceso puntual espacial 31

2.3.1. Fundamentos en la teoría de procesos puntuales: características de segundo orden 31

2.3.2. Estadísticos para un patrón de un proceso puntual homogéneo 35

2.3.3. Interacción entre puntos. Procesos de Gibbs 39

2.3.4. Intensidad condicionada 40

2.3.5. Procesos puntuales de Markov 41

2.4. Ejemplos y modelos de procesos puntuales 42

2.4.1. Proceso homogéneo de Poisson 42

2.4.2. Proceso de Poisson inhomogéneo 43

2.4.3. Proceso de Cox 44

2.4.4. Proceso cluster de Poisson 46

2.4.5. Proceso de inhibición simple (SSI) 47

2.4.6. Proceso pairwise interaction 50
2.4.7. Proceso de area-interaction 56
2.4.8. Proceso shot-noise 58
2.4.9. Proceso Candy 59

2.1. Introducción

Este primer capítulo recopila la teoría existente hasta el momento de procesos puntuales homogéneos, de forma muy detallada, con el objetivo de mostrar ampliamente el contexto sobre el cual se enmarca nuestro problema. Introduciremos brevemente una clase de procesos puntuales que vienen descritos por medidas de fuerzas o energías que actúan en y entre partículas. A esta clase de procesos se les llama procesos puntuales de Markov (procesos de Gibbs, en general, o procesos puntuales de interacción entre pares de puntos, en particular). Forman una clase útil de modelos para patrones puntuales espaciales.

2.2. Elementos probabilísticos básicos de un proceso puntual espacial

Consideraremos una variable aleatoria $X : \Omega \to \mathbb{N}$, como una aplicación X de un espacio de probabilidad (Ω, Σ, P) con una medida $(\mathbb{N}, \mathcal{P}(\mathbb{N}), \#)$, donde $\mathcal{P}(\mathbb{N})$ es la clase de todos los subconjuntos en $\mathbb{N}$ y $\#$ es la unidad de medida. En el ejemplo típico de la distribución de Poisson con media λ (número esperado de llamadas dentro de ese periodo de tiempo), la probabilidad inducida P_X en $(\mathbb{N}, \mathcal{P}(\mathbb{N}))$ viene dada por:

$$P_X(A) = P(X \in A) = \sum_{n \in A} e^{-\lambda} \frac{\lambda^n}{n!} \qquad A \subset \mathbb{N} \tag{2.1}$$

Al extender el ejemplo sobre un contexto espacial, tenemos un mapa W de la región que representa las llamadas telefónicas. Durante el día queda marcada la localización del origen de cada llamada. El resultado de tal experimento es un conjunto de puntos en W, cuyo número se distribuye Poisson de media λ y las localizaciones se distribuyen aleatoriamente.

Un proceso puntual es un modelo estocástico dado por un conjunto de puntos $\{s_i\}$ en algún conjunto X. Si las localizaciones llevan asociadas medidas o etiquetas, el proceso puntual será un proceso puntual marcado (Ripley, 1977 [85]. Al ser el objetivo un proceso puntual espacial, tomaremos X como una región acotada de $\Re^d$ o un toro, pero más gereralmente X podría ser alguna localización compacta en un espacios de Hausdorff, los cuales, topológicamente tienen una base numerable (Cressie, 1993 [26]). Por ejemplo, sucesos pueden ser árboles en un bosque, ciudades en una región geográfica, o epicentros de terremotos. Las marcas correspondientes pueden ser, respectivamente, especies, tipos o diámetros de árboles, tamaños de ciudades o magnitudes de terremotos (Besag, 1993 [15]), 1994 [16]). Aquí, asumiremos, que las localizaciones de los sucesos $\{s_i\}$ y sus correspondientes marcas $\{Z(s_i)\}$ son realizaciones de algún proceso estocástico de la forma $\{Z(s) : s \in D\}$, donde tanto $Z(.)$ como D son aleatorios (Ripley, 1987 [87]).

El sistema general para un *proceso puntual no marcado* es el siguiente: Sea (X, χ, ν) una medida del espacio tal que $\nu(X) < \infty$ y μ denota el proceso de Poisson en X con medida de intensidad λ. Estamos interesados en un proceso puntual que tenga densidad respecto de μ. Sea $X^n = X \times X \times ... \times X$ (n factores) el conjunto de colecciones de n (no necesariamente distintos) elementos de X. Tomamos $X^o = \varnothing$. Si $n \neq k$, entonces X^n y X^k son disjuntos. Sea X_e el producto cartesiano de la unión disjunta y finita de X^n,

$$X_e = \bigcup_{n=0}^{\infty} X^n \tag{2.2}$$

Cada conjunto $B \in X_e$ se puede expresar de forma única como la unión de conjuntos disjuntos; es decir, $B = \cup_{n=0}^{\infty} B^{(n)}$, donde $B^{(n)} = B \cap X^n$. Sea $\chi^{(n)}$ la menor σ-álgebra de conjuntos en X^n que contiene todos los conjuntos producto $B_1 \times B_2 \times ... \times B_n$, tal que $B_i \in \chi$, $i = 1, ..., n$. Sea χ_e la clase de todos los conjuntos $\cup_{n=0}^{\infty} B^{(n)}$ en X_e, tal que $B^{(n)} \in \chi^{(n)}$, $n = 0, 1, ...$ Entonces, χ_e es la menor σ-álgebra de conjuntos en X_e generados por conjuntos $B^{(n)}$. El par (X_e, χ_e) se llama *espacio exponencial*.

Sea ν^n el n-ésimo producto de ν en X^n para $n \geq 1$ y λ^0 la medida en X^0 del punto $\varnothing$. Entonces μ definido por:

$$\mu(B) = \sum_{n=0}^{\infty} \frac{\nu^n(B \cap X^n)}{n!}, \quad B \in \chi_e \tag{2.3}$$

es una medida de probabilidad en (X_e, χ_e). Un elemento $\mathbf{s} \in X^n$ se toma como un patrón de n puntos en X. El elemento $\varnothing$ de X^0 se interpreta como un patrón sin puntos.

Por otro lado, el sistema general de *proceso puntual marcado* es el siguiente: Sea (Ω, Λ, ρ) la medida de espacio, tal que $(\Omega) < \infty$. Tomamos ahora $\Omega = X \times F$ con $X \subset \mathbb{R}^d$ (algunas veces X se puede considerar como un toro) y $F \subset \mathbb{R}$, $\Lambda = \chi \times \xi$, donde χ es la σ-álgebra Borel de X y ξ son los conjuntos Borel de F y, finalmente, $\rho = \nu \times \pi$, donde ν es una medida finita, no atómica en X (típicamente la medida Lebesgue), π es una medida en F correspondiendo a la distribución de marcas y μ denota la marca de la medida Poisson con intensidad ρ. El espacio exponencial se denota por Ω_e y se equipa con la correspondiente σ-álgebra Λ_e.

Los procesos puntuales marcados y no marcados, se pueden también definir a través de medidas aleatorias. En las siguientes líneas supondremos que tenemos procesos puntuales marcados (el caso no marcado es similar a éste, adaptando los elementos relativos a las marcas). Sea Φ una medida contable en $X \times F$. Entonces $\Phi(A \times B)$ es el número de elementos en A cuyas marcas pertenecen a B, donde $A \in \chi$ y $B \in \xi$. Sea (Ω, Λ, P) un espacio de probabilidad en el conjunto Φ, una colección de localizaciones finitas, con medidas en $X \times F$. En Φ definimos $\mathcal{N}$ como la menor de las σ-álgebras generalizadas por conjuntos de la forma $\{\phi \in \Phi : \phi(A \times B) = n\}$,$\forall A \in \chi$, $\forall B \in \xi$ y $\forall n \in \{0, 1, 2, ...\}$. Entonces, un *proceso puntual marcado* N es un proceso en $X \times F$, es decir, una aplicación de (Ω, Λ, P) en $(\Phi, \mathcal{N})$. Un proceso puntual marcado, definido por (Ω, Λ, P) induce una medida de probabilidad $\prod_N(Y) = P(N \in Y)$,$\forall Y \in \mathcal{N}$. Ésta será la distribución de probabilidad de un proceso puntual marcado. Un proceso puntual marcado lo denotaremos por: $N = [\mathbf{s}, Z(\mathbf{s})] = \{[s_1, Z(s_1)], [s_2, Z(s_2)], ..., [s_n, Z(s_n)]\}$ y el correspondiente no marcado por N_0. Dos importantes conceptos de procesos puntuales son la estacionariedad y la isotropía. Un proceso puntual es *estacionario* si es estadísticamente invariante bajo traslaciones: N y $N_s = \{[\mathbf{s} + t; Z(\mathbf{s})]\}$ tienen la misma distribución para todo t en X. Un proceso puntual es *isotrópico* si su distribución es invariante bajo rotaciones: N y $N_r = \{[r\mathbf{s}; Z(\mathbf{s})]\}$ tienen la misma distribución para cada rotación r con respecto al origen. Si el proceso es a la vez estacionario e isotrópico se conoce como *motion-invariant* (invariante al movimiento) y el proceso trasladado tiene la distribución del proceso original para todos los movimientos euclideanos de los puntos. Si el proceso puntual no tiene sucesos múltiples, los sucesos se llaman simples. La teoría de procesos puntuales ofrece varias características para la descripción cuantitativa de la interacción

entre individuos (árboles en el bosque, por ejemplo). Uno de ellos es *la función pair-correlation*, la cual es análoga a la función de correlación de un proceso estocástico. Esta función se caracteriza por la variabilidad en el sistema de localizaciones. La función pair correlation es de gran valor para muchos problemas de estadística ecológica, por ejemplo: **(1)** Para análisis exploratorios de interacción entre árboles (tipo de interacción, determinación de los rangos de interacción y estimación de su fuerza); **(2)** Para modelización y simulación de bosques para asegurar su realismo asumiendo variantes; **(3)** Para determinar la veracidad de un estimador global en un inventario forestal (tanto en posición como en área).

2.3. Características de un proceso puntual espacial

2.3.1. Fundamentos en la teoría de procesos puntuales: características de segundo orden

Medidas momento y función de intensidad

Las medidas momento de un proceso puntual espacial son análogas al momento de una variable aleatoria o vector aleatorio. Para un proceso puntual no marcado N_0 y un conjunto Borel A, el número de puntos en A, $N(A)$, es una variable aleatoria con primer momento:

$$\mu_{N_0}(A) = E(N(A)) = \int_{\Phi} \phi(A) \prod {}_{N_0}(d\phi) \tag{2.4}$$

La medida μ_{N_0} se llama la *medida media* o *primer momento de la medida* de N_0. La *medida del k-ésimo momento* de N_0 viene dada por:

$$\begin{aligned} \mu_{N_0}^{(k)}(A_1 \times ... \times A_k) &= E(N(A_1)N(A_2)...N(A_k)) = \textstyle\int_{\Phi} \phi(A_1)\phi(A_2)...\phi(A_k) \prod {}_{N_0}(d\phi) \\ &= \textstyle\int_{\Phi} \sum_{s_1,...,s_k \in \phi} I(s_1 \in A_1)...I(s_k \in A_k) \prod {}_{N_0}(d\phi) \\ &= E\left[\textstyle\sum_{s_1,...,s_k \in \phi} I(s_1 \in A_1)...I(s_k \in A_k)\right] \end{aligned} \tag{2.5}$$

donde $A_1, ..., A_k \in \chi$. Notar que $\mu_{N_0}^{(k)}$ es una medida de $(X^k, \chi^{(k)})$. La medida de covarianza de un proceso puntual se define por $C_{N_0}(A_1 \times A_2) = \mu_{N_0}^{(2)}(A_1 \times A_2) - \mu_{N_0}(A_1)\mu_{N_0}(A_2)$, donde $A_1, A_2 \in \chi$.

Sean ds y du pequeñas regiones localizadas en s y u respectivamente. Entonces, la *intensidad de primer orden* se define como:

$$\lambda_0(s) = \lim_{\nu(ds)\to 0} \frac{\mu_{N_0}(ds)}{\nu(ds)} \tag{2.6}$$

si el límite existe. De manera similar la *intensidad de segundo orden* (Ripley, 1976 [83]) se define como:

$$\lambda_{0,2}(s,t) = \lim_{\substack{\nu(ds)\to 0 \\ \nu(du)\to 0}} \frac{\mu^{(2)}_{N_0}(ds \times dt)}{\nu(ds) \times \nu(dt)} \tag{2.7}$$

Por facilidad de notación hemos usado $\lambda(.)$ en vez de $\lambda_{N_0}(.)$. La *covarianza* será entonces $\beta(s,t) = \lambda_{0,2}(s,t) - \lambda(s)\lambda(t)$. Si el proceso puntual es invariante para el movimiento, entonces la función intensidad es una constante y escribimos $\lambda(s) = \lambda_0$ y $\lambda_{0,2}(.,.) = \lambda_0^2$. Entonces, $\beta(s,t) = 0$.

La *medida del k-ésimo momento* de N_0 se define como:

$$\begin{aligned} \alpha^{(k)}_{N_0}(A_1 \times ... \times A_k) &= \int_\Phi \sum_{\substack{s_1,...,s_k \in \phi \\ distinto}} I(s_1 \in A_1)...I(s_k \in A_k) \prod N_0(d\phi) \\ &= E\left[\sum_{\substack{s_1,...,s_k \in \phi \\ distinto}} I(s_1 \in A_1)...I(s_k \in A_k)\right] \end{aligned} \tag{2.8}$$

Particularmente, para $k = 2$, tenemos $\alpha^{(2)}_{N_0}(A_1 \times A_2) = \mu^{(2)}_{N_0}(A_1 \times A_2) - \mu_{N_0}(A_1 \cap A_2)$. Entonces, $\alpha^{(2)}_{N_0}(A_1 \times A_2)$ es la media del número de pares de puntos donde uno de los puntos está en A_1 y el otro en A_2.

Para alguna función medible no negativa $f(s_1, s_2)$, se verifican las siguientes propiedades:

$$E\left[\sum_{s_1}\sum_{s_2} f(s_1,s_2)\right] = \int f(s_1,s_2)\mu^{(2)}_{N_0}(ds_1 \times ds_2)) \tag{2.9}$$

y

$$E\left[\sum_{s_1}\sum_{s_2 \neq s_1} f(s_1,s_2)\right] = \int f(s_1,s_2)\alpha^{(2)}_{N_0}(ds_1 \times ds_2)) \tag{2.10}$$

Usualmente, $\alpha^{(2)}_{N_0}$ tiene una función densidad $l^{(2)}_{N_0}$ llamada *densidad producto de segundo orden* y están relacionadas de la siguiente forma:

$$\alpha^{(2)}_{N_0}(A_1 \times A_2) = \int_{A_1}\int_{A_2} l^{(2)}_{N_0}(s,t)dsdt \tag{2.11}$$

Para dos infinitésimos ds y dt, $\ell^{(2)}_{N_0}(s,t)dsdt$ se puede interpretar como la probabilidad de que haya un suceso en cada uno de los dos conjuntos infinitesimales especificados, conjuntos con áreas ds y dt respectivamente. Si el proceso puntual N_0 es homogéneo, entonces $\ell^{(2)}_{N_0}(s,t)$ depende sólo de la diferencia $h = t - s$. Si utilizamos coordenadas polares, h viene dado por $h = (r, \varphi)$ con $0 \leq \varphi \leq \pi$. En el caso homogéneo, $\ell^{(2)}_{N_0}(s,t) = \ell^{(2)}_{N_0}(r,\varphi) = \ell^{(2)}_{N_0}(h)$. En el caso isotrópico, la densidad producto depende sólo de la distancia r entre los puntos s y t, siendo $\ell^{(2)}_{N_0}(r)$. Usualmente, $\ell^{(2)}_{N_0}$ se normaliza por el cuadrado de la intensidad de primer orden definido, bajo isotropía, como la *función pair-correlation* $g()$,

$$g(r) = \frac{\ell^{(2)}_{N_0}(r)}{\lambda_0^2} \tag{2.12}$$

Bajo isotropía:

$$\alpha^{(2)}_{N_0}(A_1 \times A_2) = \lambda_0^2 \int_{A_1}\int_{A_2} g(r)dr \tag{2.13}$$

Para procesos puntuales completamente aleatorios se tiene que $g(r) = 1$ (procesos de Poisson homogéneos, Diggle, 2003 [35]). Valores de la función pair correlation mayores que uno indican que la distancia entre los puntos alrededor de r es relativamente más frecuente comparada con un proceso puntual completamente aleatorio, definiendo un proceso cluster y valores de $g(r)$ menores que uno que indican que la distancia correspondiente es escasa y eso define un proceso de inhibición. La función pair correlation puede tener todos los valores entre cero e infinito y para grandes valores r tiende a uno.

En mecánica estadística, la cantidad $\dfrac{\lambda_{0,2}(r)}{\lambda_0^2}$ se refiere a la *función de distribución radial*, aunque no es una función de distribución en el sentido estadístico. Nótese que $g(r) = \dfrac{\lambda_{0,2}(r)}{\lambda_0^2}$, es decir, la función pair correlation tiene la misma interpretación que la función de distribución radial.

La distribución de Palm y la medida reducida de segundo momento

La definición de la distribución de Palm se basa en las medidas Campbell. La medida Campbell de un proceso puntual N_0 es una medida en $\chi \times \mathbb{N}$ definida por

$$C_{N_0}(A \times Y) = \int_Y \phi(A) \prod {}_{N_0}(d\phi) \ , A \in \chi, Y \in \mathbb{N} \tag{2.14}$$

Como C_{N_0} es absolutamente continua respecto de μ_{N_0}, por el teorema de Radon-Nikodym, existe una medida determinada de manera única $P_{N_0,s}$ en $(\Phi, \mathbb{N})$ como

$$C_{N_0}(A \times Y) = \int_B P_{N_0,s}(Y)\mu_{N_0}(ds) \tag{2.15}$$

La medida $P_{N_0,s}$ se llama *distribución de Palm* de N_0 con respecto a s.

La medida reducida de Campbell de un proceso puntual N_0 es una medida en $\chi \times \mathbb{N}$ definida por:

$$C'_{N_0}(A \times Y) = \int_\Phi \int_B I[(\phi - \delta_s) \in Y]\phi(ds) \prod {}_{N_0}(d\phi) \ , \ A \in \chi, Y \in \mathbb{N} \tag{2.16}$$

Para un suceso s de ϕ, $\phi - \delta_s$ representa el patrón puntual ϕ excepto el suceso s. Para cada realización $\phi \in \Phi$, la integral $\int_B I[(\phi - \delta_s) \in Y]\phi(ds)$ contiene el número de sucesos s de ϕ, que están en B, tal que $\phi - \delta_s \in Y$. Ciertamente, $0 \leq C'_{N_0}(AxY) \leq C_{N_0}(AxY) \leq \mu_{N_0}(A)$ y por tanto C'_{N_0} es absolutamente continua respecto de μ_{N_0}. Por el teorema de Radon-Nikodym tenemos:

$$C'_{N_0}(A \times Y) = \int_B P'_{N_0,s}(Y)\mu_{N_0}(ds) \tag{2.17}$$

donde $P'_{N_0,s}$ se llama *distribución de Palm reducida* de N_0 respecto de s. $P'_{N_0,s}$ se puede interpretar como la distribución condicional de N_0 en el conjunto reducido $X - \{s\}$, dado que hay un suceso $s \in X$. El proceso puntual en $X - \{s\}$ con medida de probabilidad $P'_{N_0,s}$ se llama *proceso reducido de Palm* de N_0.

La medida del segundo momento de la distribución de Palm reducida es importante para modelar el proceso puntual estacionario. Supongamos que el proceso N_0 es estacionario y con intensidad λ_0. Entonces la *medida reducida del segundo momento* $\mathcal{K}_{N_0}$ viene dada por:

$$\lambda_0 \mathcal{K}_{N_0}(A) = \int_\Phi \phi(A) P'_{N_0,0}(d\phi) = \int_\Phi \phi(A - \{0\}) P_{N_0,0}(d\phi) \ , \ A \in \chi \tag{2.18}$$

donde $P'_{N_0,0}$ es la distribución de Palm reducida original. Si $A = b(0,h)$ es la bola cerrada de radio h centrada en el origen, entonces $\mathcal{K}_{N_0}(b(0,h))$ se denota como $K_{N_0}(h)$ y a menudo se le llama K-función (Diggle, 2003 [35]; Ripley, 1976, 1977 [83], [84]). $\lambda_0 K_{N_0}(h)$ representa el número esperado de sucesos extra de N_0 dentro de la distancia h de un suceso de N_0.

Bajo estacionariedad tenemos que:

$$\lambda_0^2 \mathcal{K}_{N_0}(dh) = \ell^{(2)}_{N_0}(h) dh \tag{2.19}$$

y bajo isotropía,

$$\lambda_0^2 dK_{N_0}(r) = 2\pi r \ell^{(2)}_{N_0}(r) dr \tag{2.20}$$

Consecuentemente, obtenemos:

$$g(r) = \frac{1}{2\pi r} \frac{d}{dr} K_{N_0}(r) \tag{2.21}$$

o alternativamente,

$$K_{N_0}(r) = \int_0^r g(u) 2\pi u du, \ r > 0 \tag{2.22}$$

Para un proceso de Poisson homogéneo $K_{N_0}(r) = \pi r^2$.

Una práctica común es usar la versión transformada de la K-función $L_{N_0}(r) = \sqrt{K_{N_0}(r)/\pi}$. Para procesos de Poisson $L(r) = r$, una línea recta con pendiente 1. La transformación $K \to L$ tiene también la ventaja de que para un proceso puntual, la varianza del estimador de $L(r)$ es aproximadamente constante para todas las distancias r. En estadística espacial, la función pair correlation se utiliza a menudo para análisis de datos exploratorios y la función K de Ripley o la función L, para contraste de hipótesis.

2.3.2. Estadísticos para un patrón de un proceso puntual homogéneo

Estimación de la medida reducida del segundo momento y K-función

Asumiremos que tenemos n puntos, $s = \{s_1, s_2, ..., s_n\}$ observados en una ventana X. Cada punto consta de dos coordenadas en el plano $s_i = (x_i, y_i)$,

$i = 1, 2, ..., n$. Usualmente $\lambda_0^2 K_{N_0}(r)$ y $\lambda_0^2 \mathcal{K}_{N_0}(A)$ son estimados en primer lugar y los estimadores de $K_{N_0}(r)$ y $\mathcal{K}_{N_0}(A)$ se obtienen dividiendo $\widehat{\lambda}_0^2 = \frac{n}{\nu(X)}$ (a menudo, $\nu(X) = |X|$, la medida Lebesgue). En la obtención del estimador, debemos distinguir entre el caso anisotrópico y el isotrópico (Heinrich & Liebscher, 1997 [52]). Tambien, la función L se estima por:

$$\widehat{L}_{N_0}(r) = \sqrt{\widehat{K}_{N_0}(r)/\pi} \tag{2.23}$$

Supongamos ahora que estamos en el caso más general. Un estimador $k_{N_0}(r)$ de $\lambda_0^2 K_{N_0}(r)$ vendrá dado por:

$$k_{N_0}(r) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1_{b(0,r)}(s_j - s_i)}{\nu(X_{s_j} \cap X_{s_i})} \tag{2.24}$$

Aquí X_{s_j} significa la ventana X trasladada por s_j. Cabe notar que:

$$\nu(X_{s_j} \cap X_{s_i}) = \nu(X \cap X_{s_i - s_j}) = \nu(X \cap X_{s_j - s_i}) \tag{2.25}$$

Análogamente, $\lambda_0^2 \mathcal{K}_{N_0}(A)$ se puede estimar por:

$$k_{N_0}(A) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1_A(s_j - s_i)}{\nu(X_{s_j} \cap X_{s_i})} \tag{2.26}$$

Notar que si X es un rectángulo de lados a y b, entonces $\nu(X_{s_j} \cap X_{s_i}) = |a - |x_i - x_j|| \, |b - |y_i - y_j||$.

Particularmente, para el caso donde $A = S(r, \varphi)$, $K(r, \varphi)$, $0 \leq \varphi \leq \pi$, se puede utilizar, usando coordenadas polares,

$$\lambda^2 \widehat{K}(r, \varphi) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1_{[0,r]}(d(s_i, s_j)) 1_{[0,\varphi]}(\varphi(s_i, s_j))}{\nu(X_{d(s_i,s_j)} \cap X_{\varphi(s_i,s_j)})} \tag{2.27}$$

Sea $d(s_i, s_j) = r$ y $\varphi(s_i, s_j) = \gamma$ y sean (x, y) las coordenadas cartesianas correspondientes a (r, γ). Entonces $\nu(X_{d(s_i,s_j)} \cap X_{\varphi(s_i,s_j)}) = |a - x| \, |b - y|$ en el caso de región rectangular con lados a y b.

Bajo los supuestos de **homogeneidad e isotropía** tenemos los siguientes estimadores:

El estimador Ohser para $\lambda_0^2 K_{N_0}(r)$ (Ohser & Stoyan, 1981 [79]) viene dado por:

$$k_{N_0}^O(r) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1_{[0,r]}(\|s_j - s_i\|)}{c_X(\|s_j - s_i\|)} \tag{2.28}$$

donde $c_X(r)$ es el conjunto de isotropía de la función de covarianza de la ventana X, que en general, para $h = (r, \varphi)$ se define como la media de la función del conjunto covarianza (o covariograma geométrico) $\gamma_X(h) = \nu(X \cap X_h)$ (el área de intersección de X y X trasladado por h unidades),

$$c_X(h) = \frac{1}{\pi} \int_0^{\pi} \gamma_X(r, \varphi) d\varphi \tag{2.29}$$

$c_X(r)$ se puede considerar como un factor de corrección de bordes y para círculos y rectángulos tenemos la forma:

1. Si X es un círculo de diámetro d

$$c_X(r) = \begin{cases} \frac{d^2}{2}[\arccos(\frac{r}{d}) - \frac{r}{d}(1 - (\frac{r}{d})^2)^{1/2}], & si\ 0 < r \leq d \\ 0 & si\ r > d \end{cases} \tag{2.30}$$

2. Si X es un rectángulo de lados a y b, $a < b$, la fórmula (ver Osher & Stoyan, 1981 [79]) es:

   a) si $0 < r < a$

$$c_X(r) = ab - r(2a + 2b - r)/\pi \tag{2.31}$$

   b) si $a \leq r < b$

$$c_X(r) = \left\{ ab\left(\pi - 2\arccos\frac{a}{r}\right) - a^2 - 2br\left(1 - \left(1 - \left(\frac{a}{r}\right)^2\right)^{1/2}\right)\right\}/\pi \tag{2.32}$$

   c) si $b \leq r < (a^2 + b^2)^{1/2}$

$$c_X(r) = \left\{ab\left(2\arcsin\tfrac{b}{r} - 2\arccos\tfrac{a}{r}\right)\right\}/\pi + \\ + \left\{2r\left[a\left(\left(1 - \left(\tfrac{b}{r}\right)^2\right)^{1/2} - \tfrac{a}{r}\right) - b\left(\left(1 - \left(\tfrac{a}{r}\right)^2\right)^{1/2} - \tfrac{b}{r}\right)\right] + a^2 + b^2 - r^2\right\}/\pi \tag{2.33}$$

d) en otro caso, $c_X(r) = 0$

El estimador Ripley para $\lambda_0^2 K_{N_0}(r)$ (Ripley, 1976 [83]) viene dado por:

$$k_{N_0}^R(r) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1_{[0,r]}(\|s_j - s_i\|) b_{ij}}{\nu(X^{\|s_j - s_i\|})} \tag{2.34}$$

para $0 \leq r \leq r^* = \sup\{r : \nu(X^r) > 0\}$, donde $X^r = \{s \in X : b(s,r) \cap X \neq \varnothing\}$ y $b_{ij} = 2\pi/\alpha_{ij}$, siendo α_{ij} la suma de los ángulos centrales que pertenecen a los arcos del círculo centrado en s_i de radio $\|s_j - s_i\|$ sobre X. Para r pequeño y X convexo, $X^r = X$ y r^* es igual a la longitud de la diagonal.

Estimación de la función pair-correlation

En este caso, la función densidad producto $l_{N_0}^{(2)}(r)$ se estima en primer lugar. Una estimación de la función pair correlation viene dada por $\widehat{g}(r) = \dfrac{\widehat{l}_{N_0}^{(2)}(r)}{\widehat{\lambda}_0^2}$ (Stoyan, Bertram & Wendroch, 1993 [97]; Stoyan & Stoyan, 2000 [100]). Para la obtención del estimador necesitamos aplicar funciones núcleo. Una función núcleo adecuada es el núcleo de Epanechnikov, que viene dado por:

$$K_\varepsilon(h) = \begin{cases} (1 - h^2/\varepsilon^2)\frac{3}{4\varepsilon} & -\varepsilon \leq h \leq \varepsilon \\ 0 & \text{otro caso} \end{cases} \tag{2.35}$$

El parámetro ε determina la suavidad de la estimación. La experiencia computacional recomienda un valor de ε de:

$$\varepsilon = (1/10)\sqrt{\nu(X)/\lambda_0} \tag{2.36}$$

Para el caso **homogéneo e isotrópico**,

$$\widehat{l}_{N_0}^{(2)}(r) = \frac{1}{2\pi r} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{k_\varepsilon(r - \|s_i - s_j\|)}{\nu(X_{s_j} \cap X_{s_i})} \tag{2.37}$$

Notar que $-\varepsilon \leq r - \|s_i - s_j\| \leq \varepsilon$ es equivalente a $r - \varepsilon \leq \|s_i - s_j\| \leq r + \varepsilon$.

Y las estimaciones correspondientes de la densidad producto $l_{N_0}^{(2)}(r)$, usando las expresiones de Ripley & Osher son, respectivamente:

$$\widehat{l}_{N_0}^{(2),R}(r) = \frac{1}{2\pi r}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{k_\varepsilon(r-\|s_i-s_j\|)b_{ij}}{\nu(X^{\|s_j-s_i\|})} \tag{2.38}$$

y

$$\widehat{l}_{N_0}^{(2),O}(r) = \frac{1}{2\pi r c_X(r)}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}k_\varepsilon(r-\|s_i-s_j\|) \tag{2.39}$$

Por ejemplo, un estimador para la función pair correlation es:

$$\widehat{g}(r) = \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}k_\varepsilon(r-\|s_i-s_j\|)/(\widehat{\lambda}_0^2 2\pi r c_X(r)),\ r>0 \tag{2.40}$$

Estimaciones de funciones individuales L y g

En ciertas ocasiones, nos puede interesar determinar aquellos puntos que en un proceso puntual juegan un papel particularmente importante. Pueden ser puntos de un origen diferente al resto de puntos, o puntos que estén en el patrón por error. En este caso, obtendremos relaciones de vecindad con el resto de puntos diferentes a las usuales. Estas diferencias se pueden caracterizar por las funciones individuales L y g definidas a continuación. Así, si s es un punto diferente y $s_1, ..., s_n$ son los puntos normales del patrón, las funciones individuales se definen como:

$$L_s(r) = \sqrt{\frac{\nu(X)N(b(s,r))}{n\pi}} \tag{2.41}$$

y

$$g_s(r) = \frac{\nu(X)\sum_{i=1}^{n}k_\varepsilon(r-\|s-s_i\|)}{2\pi r n} \tag{2.42}$$

2.3.3. Interacción entre puntos. Procesos de Gibbs

Un *proceso puntual de Gibbs* finito es un proceso puntual simple y finito, con una función de densidad f que cumple la condición de positividad

$f(\mathbf{s}) > 0$, lo que implica $f(\mathbf{t}) > 0$ para todo $\mathbf{t} \subset \mathbf{s}$.

Un conjunto $H \subset X_e$ se llama *hereditario* si la relación

$$\mathbf{s} \in H \text{ implica } \mathbf{t} \in H \text{ para todo } \mathbf{t} \subset \mathbf{s}$$

De esta forma, un proceso finito de Gibbs tiene como función densidad f, definida en el conjunto hereditario $H = \{\mathbf{s} : \ f(\mathbf{s}) > 0\}$ a:

$$f(\mathbf{s}) = \exp\left\{-v_0 - \sum_{i=1}^{n(\mathbf{s})} v_1(s_i) - \sum_{i<j} v_2(s_i, s_j) - \sum_{i<j<k} v_3(s_i, s_j, s_k) - \dots\right\} \quad (2.43)$$

donde v_0 es constante y $v_k : S^k \to \mathbb{R}^\infty$ son funciones simétricas para $k = 1, 2, \dots$.

La función individual v_k se llama *potencial de interacción* y los procesos puntuales se pueden clasificar por el orden de interacción de los puntos.

Un proceso puntual tiene *interacción entre puntos de orden k* si $\phi : S_e \to [0, \infty]$ tal que $k = \sup\{j \in \mathbb{N} : v_j \ncong 0\}$. Sabiendo que una función $\phi : S_e \to [0, \infty]$ tal que

$$\phi(\varnothing) = \exp(-v_0), \quad \phi(\{\psi\}) = \exp(-v_1(\psi)),$$

$$\phi(\{\psi, \eta\}) = \exp(-v_2(\psi, \eta)), \quad \dots$$

se puede reescribir la función f como

$$f(\mathbf{s}) = \prod_{\mathbf{t} \subset \mathbf{s}} \phi(\mathbf{t})$$

y se puede llamar a la función ϕ la *función de interacción* del proceso puntual dado por f.

2.3.4. Intensidad condicionada

La intensidad condicionada Papangelou de un proceso puntual X es la función $\lambda : S \times S_e \to [0, \infty)$ que satisface la expresión:

$$E\left[\sum_{s_i \in X \cap A} g(s_i, X)\right] = \int_A E[\lambda(u, \mathbf{s}) g(u, \mathbf{s})] d\nu(u) \quad (2.44)$$

para todo $A \in \mathcal{B}$ y $g : S \times S_e \to [0, \infty)$ (fórmula de 'Nguyen-Zessin'). Una posible interpretación es:

$$\lambda(u; \mathbf{s}) = \lim_{\Delta\mathbf{u} \downarrow \{u\}} \frac{P(X \cap \Delta\mathbf{u} \neq \varnothing \mid X \setminus \Delta\mathbf{u} = \mathbf{s} \setminus \Delta\mathbf{u})}{\nu(\Delta\mathbf{u})}$$

donde el límite viene dado por el decrecimiento de los vecinos $\Delta\mathbf{u}$ de $u \in X$. Esta expresión proporciona

$$\lambda(u; \mathbf{s}) = \begin{cases} \dfrac{f(\mathbf{s} \cup \{u\})}{f(\mathbf{s})}, & u \notin \mathbf{s} \\ \\ \dfrac{f(\mathbf{s})}{f(\mathbf{s} \setminus \{s_i\})}, & u = s_i \in \mathbf{s} \end{cases}$$

en los procesos de Gibbs. Donde $u \notin \mathbf{s}$, $\lambda(u; \mathbf{s})$ se puede interpretar como la probabilidad de añadir un nuevo punto u a la configuración.

2.3.5. Procesos puntuales de Markov

La noción de campos aleatorios de Markov nos lleva a la definición de procesos puntuales de Markov basados en la definición de interacción entre puntos. La familia de procesos de Markov es un buen modelo para la regularidad en patrones puntuales, pero no se muestra igual de satisfactorio para una gran variedad de patrones agrupados. Nuevos modelos vienen a llenar este vacío, tales, como los procesos de interacción de área, los procesos de shot-noise y los procesos de interacción quermass. En esta sección, un proceso puntual finito de Gibbs, con una función de densidad expresada como en (2.43), se designa para mostrar la interacción entre puntos 'vecinos'. Con esta idea, una relación de vecindad (reflexiva y simétrica) entre puntos de X se representa como $\sim$. Esto nos permite decir que un proceso puntual con este tipo de consideraciones tiene una función de densidad del tipo:

$$f(\mathbf{s}) = \exp\left\{ -v_0 - \sum_{i=1}^{n(\mathbf{s})} v_1(s_i) - \sum_{\substack{i<j \\ s_i \sim s_j}} v_2(s_i, s_j) - \sum_{\substack{i<j<k \\ s_i \sim s_j \\ s_i \sim s_k \\ s_j \sim s_k}} v_3(s_i, s_j, s_k) - \dots \right\} \tag{2.45}$$

Se puede decir que f es una función densidad de un 'Proceso puntual de Markov' con cada función v_k verificando la propiedad

> $v_k(s_1, s_2, \ldots, s_k) = 0$ si el conjunto $\{s_1, s_2, \ldots, s_k\}$ no está formado sólo por pares de vecinos para $k = 2, 3, \ldots$.

Una función de densidad $f : S_e \to [0, \infty)$ se llama una *función Markov* si:

- El conjunto $H = \{\mathbf{s} : \ f(\mathbf{s}) > 0\}$ es hereditario.
- La expresión $\dfrac{f(\mathbf{s} \cup \psi)}{f(\mathbf{s}) + f(\mathbf{s} \cup \psi)}$ depende sólo de $\psi \in S$ y del conjunto de puntos $\{u \in \mathbf{s} : \ u \sim \psi\}$.

Una función $\phi : X_e \to [0, \infty)$ se llama *función de interacción* si $\phi(\mathbf{s}) \neq 1$ implica $\psi \sim \eta$ para todo $\psi, \eta \in \mathbf{s}$.

Un *proceso puntual de Markov* es un proceso puntual cuya distribución tiene una función de densidad f que es una función Markov.

2.4. Ejemplos y modelos de procesos puntuales

2.4.1. Proceso homogéneo de Poisson

Un proceso homogéneo de Poisson, es el pilar sobre el cual se basa la teoría de procesos puntuales (Diggle, 1986 [33]). Representa el caso más simple de mecanismo estocástico para la generación de patrones puntuales estocásticos, y se aplica para trabajar con el supuesto de aleatoriedad espacial completa, aunque en la práctica esto no pase. El proceso de Poisson cumple:

1. Para algún $\lambda > 0$, y alguna región finita del plano A, el número de puntos que se encuentran en esa región $N(A)$, tiene una distribución Poisson con media $\lambda |A|$, siendo $|A| = \nu(A)$ el área de A.

2. Dados $N(A) = n$, los n sucesos en A forman una muestra aleatoria e independiente de una distribución en A.

Para demostrar que 1 y 2 son consistentes por ellos mismos, estableceremos primero:

3. Para cualquiera dos regiones disjuntas A y B, $N(A)$ y $N(B)$ son independientes.

En la Figura (2.1) se muestran realizaciones de un proceso de Poisson homogéneo en el cuadrado unidad.

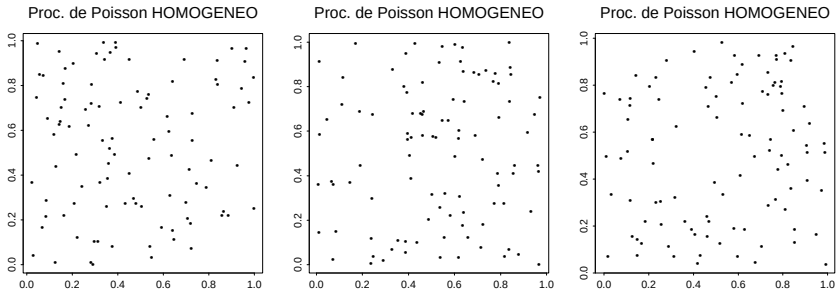

Figura 2.1: Realizaciones de procesos de Poisson homogéneos en el cuadrado unidad

2.4.2. Proceso de Poisson inhomogéneo

Una clase de proceso puntual no estacionario se obtiene si la constante de intensidad λ del proceso de Poisson se reemplaza por una variable con función de intensidad $\lambda(x)$ (Diggle, 1986 [33]). Esto define la clase de procesos de Poisson inhomogéneos, definidos bajo las condiciones:

1. $N(A)$ tiene una distribución Poisson con media $\int_A \lambda(x)dx$.

2. Dados $N(A) = n$, los n sucesos en A forman una muestra aleatoria e independiente de una distribución en A con función de densidad de probabilidad (PDF) proporcional a $\lambda(x)$.

Las Figuras (2.2) y (2.3) muestran realizaciones de procesos puntuales de Poisson inhomogéneos. El patrón puntual de la Figura (2.2) se basa en la función de intensidad

$$\lambda(\mathbf{s}) = 60\left[2 + sen(2\pi x) + sen(2\pi y)\right]$$

con $\mathbf{s} = (x, y)$, mientras que los mostrados en la Figura (2.3) se basa en la función de intensidad

$$\lambda(\mathbf{s}) = n + 20\sqrt{n}\,(x + y)$$

donde $\mathbf{s} = (x, y)$.

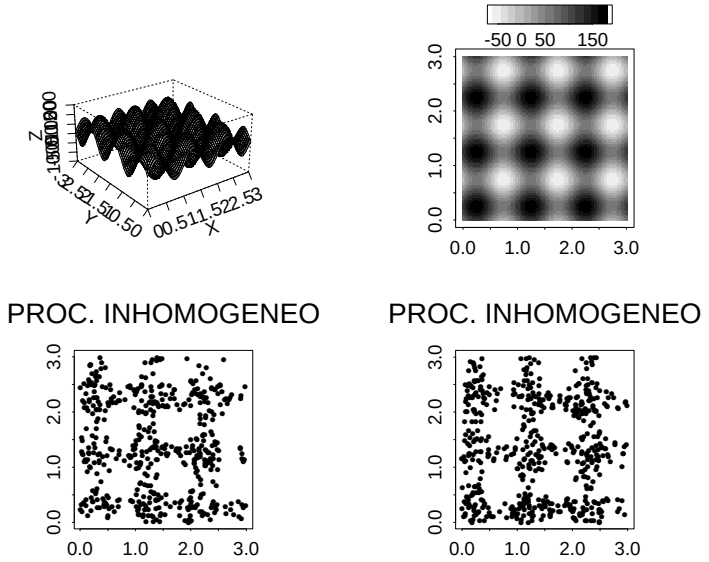

Figura 2.2: Realizaciones de un proceso de Poisson inhomogéneo con función de intensidad $\lambda(s) = 60[2 + \sin(2\pi x) + \sin(2\pi y)]$

2.4.3. Proceso de Cox

En algunas aplicaciones, es razonable pensar en variar espacialmente la función intensidad $\lambda(s)$ de un proceso de Poisson para ser ella misma un proceso estocástico $\Lambda(s)$. Definiremos un proceso de Cox por medio de las siguientes características:

1. $\Lambda(s)$ es un proceso estocástico no negativo

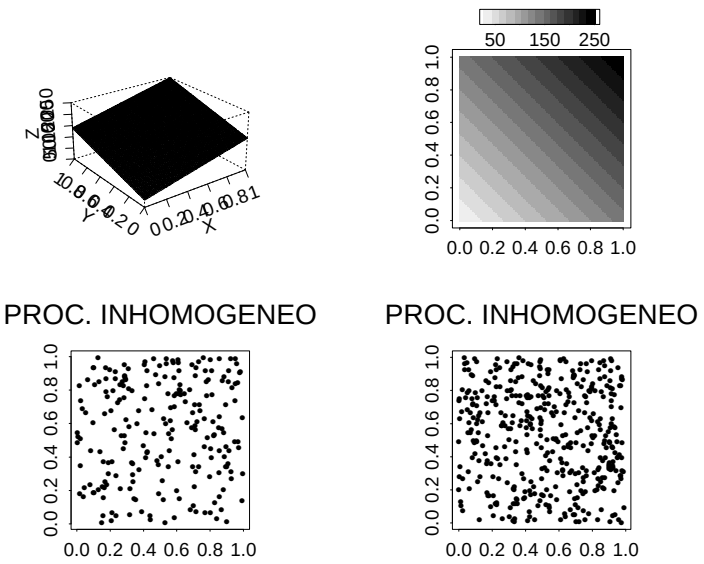

Figura 2.3: Realizaciones de un proceso de Poisson inhomogéneo con función de intensidad $\lambda(s) = n + 20\sqrt{n}(x + y)$.

2. Dada una realización de $\Lambda(s)$, el proceso puntual es un proceso de Poisson inhomogéneo con función de intensidad $\Lambda(s)$

El resultado del proceso puntual hereda las propiedades del proceso $\Lambda(s)$ de una forma natural:

1. El proceso puntual es estacionario y/o isotrópico según sea estacionario y/o isotrópico $\Lambda(s)$

2. Si $\Lambda(s)$ es estacionario e isotrópico, con $E[\Lambda(s)] = \mu$ y $Cov\{\Lambda(s), \Lambda(t)\} = \gamma(\|s - t\|)$, entonces el proceso puntual tiene intensidad μ y covarianza $\gamma(.)$. En particular, la función de segundo momento reducido es:

$$K(r) = \pi r^2 + 2\pi\mu^{-2} \int_0^r \gamma(z)\, z dz \tag{2.46}$$

El proceso de Cox se usa a menudo para modelar fenómenos en los que la distribución espacial de los sucesos se presupone que es resultado de la variación de factores ambientales, por ejemplo en epidemiología.

2.4.4. Proceso cluster de Poisson

Se trata de un proceso muy común en la teoría de procesos puntuales y ha de cumplir las siguientes condiciones:

1. Los *padres* tienen la forma de un proceso de Poisson homogéneo de intensidad ρ

2. El número de *hijos* por padre es una variable aleatoria M, independiente para cada *padre*

3. La posición de cada *hijo* respecto a su *padre* es una variable bivariante e independiente para cada hijo

4. El proceso puntual observado consiste en la superposición de todos los *hijos*

Un proceso cluster de Poisson es muy diferente conceptualmente a un proceso de Cox, pero puede producir patrones similares o incluso no distinguirse.

Los *procesos de Neyman-Scott*, (Neyman & Scott, 1956 [78]), constituyen una familia particular de los procesos cluster de Poisson, en los que alrededor de cada padre se distribuye un cluster de puntos hijos. El número de puntos por cluster se distribuye de forma aleatoria siguiendo una cierta función de probabilidad discreta. La localización de estos hijos respecto a cada padre se genera de forma independiente para cada padre siguiendo una función de densidad de probabilidad.

El *proceso de Thomas* pertenece a la familia de los procesos de Neyman-Scott, donde los padres vienen generados a partir de una distribución de Poisson con intensidad λ. Posteriormente, cada padre es reemplazado por un cluster de puntos cuyo número se distribuye según una Poisson de media μ y sus posiciones respecto a cada padre proceden de una distribución Gaussiana.

Finalmente, otra subclase de los procesos Neyman-Scott es la definida por *Matérn* en 1960. Los procesos Matérn se caracterizan porque cada padre viene rodeado de una esfera de radio R en la que se distribuyen aleatoriamente m puntos según una distribución binomial. El número de hijos sigue una distribución de Poisson con media μ.

En las Figuras (2.4), (2.5) y (2.6) se muestran realizaciones de todos estos tipos de procesos cluster de Poisson para diferentes conjuntos de parámetros, tal y como se especifica en el pie de cada Figura.

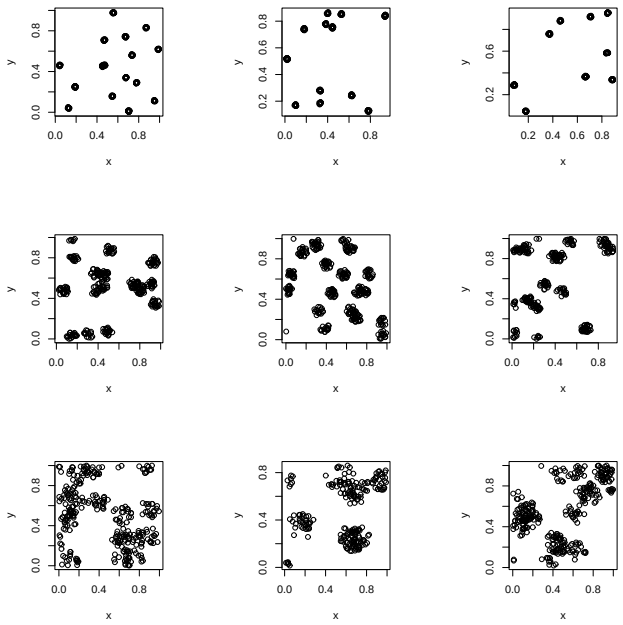

Figura 2.4: Realizaciones de procesos Matérn con número medio de padres $\lambda = 15$ y número medio de hijos por padre $\mu = 20$. *Primera fila:* El radio de la esfera que rodea cada padre es R=0.01; *Segunda fila:* El radio de la esfera que rodea cada padre es R=0.05; *Tercera fila:* El radio de la esfera que rodea cada padre es R=0.10.

2.4.5. Proceso de inhibición simple (SSI)

Construiremos ahora patrones regulares mediante la imposición de una distancia mínima permisible δ, entre dos elementos cualesquiera. Esto puede reflejarse en física o biología, o puede manifestarse en más situaciones, como competición entre plantas o territorio de animales. Procesos de este tipo no incorporan aleatoriedad y se llamarán simplemente procesos de inhibición, una noción que puede ser formalizada de diversas formas. Para una terminología contextualizada, llamaremos *intensidad de grupo* de un proceso de inhibición a

$$\tau = \lambda\pi\delta^2/4 \tag{2.47}$$

donde λ es la intensidad. Así τ es la proporción en el plano cubierto por discos no solapados de diámetro δ . Notar que la intensidad máxima de τ se obtiene por

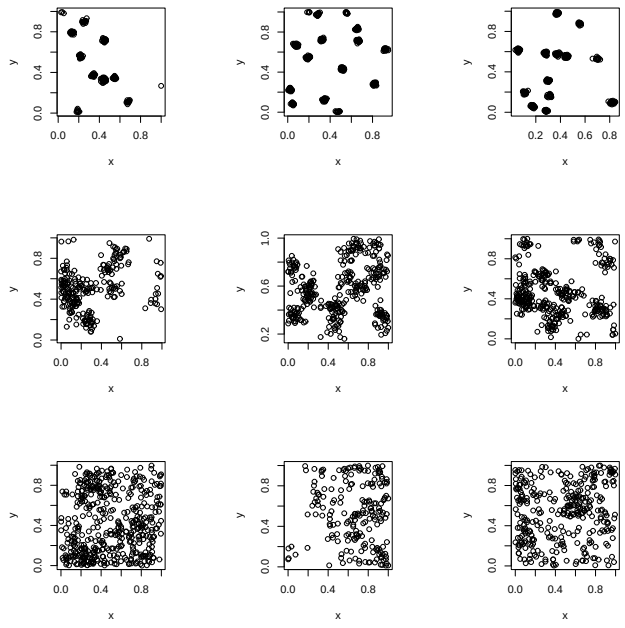

Figura 2.5: Realizaciones de procesos Thomas con número medio de padres $\lambda = 15$ y número medio de hijos por padre $\mu = 20$. *Primera fila:* La varianza de la distribución Gaussiana que gobierna los desplazamientos de los hijos respecto a los padres es 0.01; *Segunda fila:* La varianza de la distribución Gaussiana que gobierna los desplazamientos de los hijos respecto a los padres es 0.05; *Tercera fila:* La varianza de la distribución Gaussiana que gobierna los desplazamientos de los hijos respecto a los padres es 0.10.

discos cerrados cuyos centros forman un retículo en forma de triángulo equilátero.

Matérn (1986) [70], describe un proceso de Poisson de intensidad ρ para la detección de todos los pares de elementos a una distancia menor que δ. La probabilidad que un suceso arbitrario sobreviva, es por lo tanto $\exp(-\pi\rho\delta^2)$ y la intensidad de un proceso de inhibición simple es:

$$\lambda = \rho \exp\left(\pi\rho\delta^2\right) \tag{2.48}$$

Con esto podemos definir un proceso de inhibición secuencial simple, considerando una secuencia de n sucesos S_i en A y siendo $d(s,t)$ la distancia entre dos puntos x e y, de la siguiente forma:

1. S_i se distribuyen uniformemente en A

2. Dados $\{S_j = s_j, j = 1, ..., i-1\}, S_j$ se distribuyen uniformemente en la intersección de A con $\{t : d(t, s_j) \geq \delta, j = 1, ..., i-1\}$

Un proceso de inhibición secuencial se parametriza, de manera más natural, por una intensidad de grupo $\tau = n\pi\delta^2/4A$. La Figura (2.7) representa realizaciones de un proceso SSI para diferentes valores de δ.

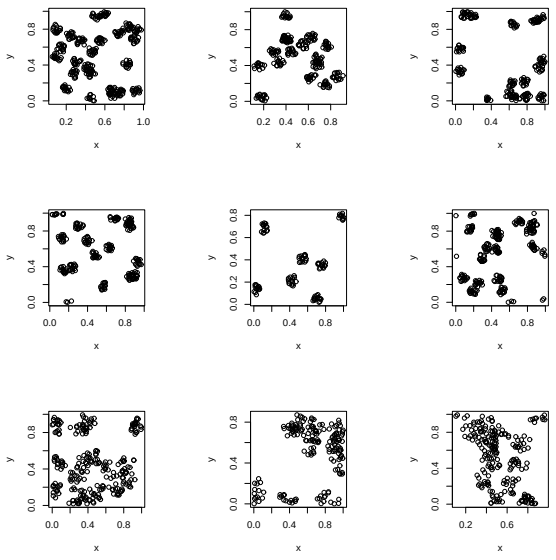

Figura 2.6: Realizaciones de procesos Neyman-Scott con número medio de padres $\lambda = 15$ y número medio de hijos por padre $\mu = 15$. *Primera fila:* El radio máximo del cluster aleatorio es R=0.01 y la función que genera los clusters aleatorios es uniforme en un disco de radio 0.05; *Segunda fila:* El radio máximo del cluster aleatorio es R=0.05 y la función que genera los clusters aleatorios es uniforme en un disco de radio 0.05; *Tercera fila:* El radio máximo del cluster aleatorio es R=0.05 y la función que genera los clusters aleatorios es uniforme en un disco de radio 0.10.

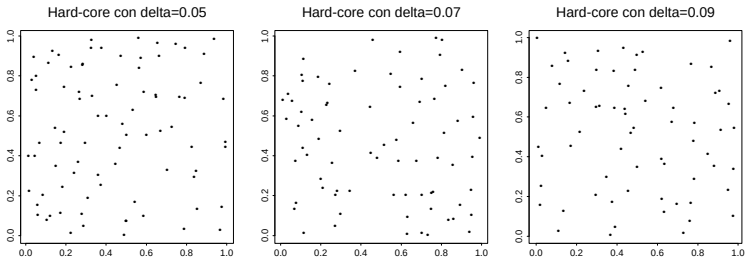

Figura 2.7: Realizaciones de un proceso simple de inhibición.

2.4.6. Proceso pairwise interaction

Al incrementar el orden de la interacción entre puntos, aparecen nuevos modelos. Al permitir un segundo orden de interacción, podemos escribir la función de densidad como:

$$f(\mathbf{s}) = \alpha \prod_{i=1}^{n(\mathbf{s})} b(s_i) \prod_{i<j} h(s_i, s_j)$$

para $\mathbf{x} \in W$, α la constante normalizadora, $b : W \to [0, \infty]$ la función de intensidad y $h : W \times W \to [0, \infty]$ la función de interacción entre puntos ($h(\psi, \psi) = 1$ para todos $\psi \in W$). Una condición suficiente para la integrabilidad de estas funciones es la condición $h(,) \leq 1$. En este caso, al proceso se le llama *puramente inhibitorio*. La intensidad condicional de estos procesos viene dada por:

$$\lambda(u; \mathbf{s}) = \begin{cases} b(u) \prod_{s_i \in \mathbf{s}} h(u, s_i) & u \notin \mathbf{s} \\ b(u) \prod_{s_i \in \mathbf{s} \setminus \{u\}} h(u, s_i) & u \in \mathbf{s} \end{cases}$$

i.e., depende sólo de u y en los puntos ψ de $\mathbf{s}$ tal que $h(u, \psi) \neq 1$. Esta observación nos permite considerar el proceso pairwise como un proceso tipo Ripley - Kelly Markov bajo la relación $\psi \sim \eta$ si y sólo si $h(\psi, \eta) \neq 1$.

Algunos procesos puntuales de este tipo.

Proceso Strauss

Para una distancia dada fija $R > 0$, el proceso de Strauss se define por la función de densidad

$$f(\mathbf{s}) = \frac{1}{c(\beta,\gamma)}\beta^{n(\mathbf{s})}\gamma^{S(\mathbf{s})}$$

donde $\frac{1}{c(\beta,\gamma)} > 0$ es la constante normalizadora, $\beta, \gamma > 0$ son los parámetros y $S(\mathbf{s})$ es el número de vecinos (Strauss, 1975 [101]; Kelly & Ripley, 1976 [58]). Obsérvese que

$$S(s_1, ..., s_n) = \sum_{i=1}^{n}\sum_{j>i}\mathbf{1}\left[\|s_i - s_j\| \leq r\right] \tag{2.49}$$

donde $\mathbf{1}[\cdot]$ es la función indicatriz.

Se puede demostrar que no es una función integrable para $\gamma > 1$, aunque se ha usado para simular los patrones en agrupaciones.

Considerando la función de densidad como un indicador de la probabilidad de una realización de puntos, se puede analizar cómo el parámetro γ determina la bondad de las realizaciones (para $n(\mathbf{s})$). Cuando $\gamma < 1$, la probabilidad disminuye con el número de pares de R-vecinos. De esta forma, los valores $\gamma < 1$ dan más probabilidad a los patrones regulares (para n pequeño comparado al tamaño de la ventana). El caso $\gamma = 1$ reproduce el proceso de Poisson, mientras $\gamma > 1$ produce los patrones en agrupaciones .

Gráficamente, un punto aislado en W crea una inhibición (γ) que es uniforme dentro de la bola de radio R y nula fuera de ella. Una configuración de puntos con intensidad condicional $\lambda(u;\mathbf{s}) = \beta\gamma^{S(\mathbf{s}\cup\{u\})-S(\mathbf{s})}$ muestra la creación de una escala de inhibiciones $(\beta\gamma, \beta\gamma^2, \beta\gamma^3, \ldots)$ en la región W, dependiendo de dónde y cómo la unión de discos (de centros perteneciendo a la configuración y radio R) interactúan. Fuera de ellos $\lambda(u;\mathbf{x}) = \beta$, es decir, no hay interacción alguna, ver Figura (2.8)).

Podemos expresar la densidad de un proceso de Strauss de forma exponencial a través de un vector bivariante $t(\mathbf{s}) = (n(\mathbf{s}), S(\mathbf{s}))$ y $\theta = (\theta_1, \theta_2)$, para obtener

$$f(\mathbf{s};\theta) = \frac{1}{c(\theta)}g(\mathbf{s};\theta)$$

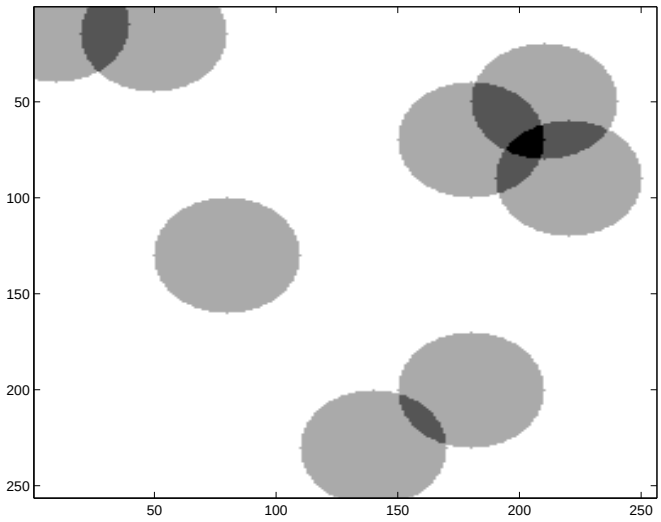

Figura 2.8: Configuración de un proceso Strauss con radio $R = 15$ en una región 256×256 y la correspondiente función de intensidad condicionada. Claros y sombras indican valores de la función de intensidad condicionada.

y

$$g(\mathbf{s};\theta) = e^{<t(\mathbf{s}),\theta>} = \exp(n(\mathbf{s})\theta_1 + S(\mathbf{s})\theta_2)$$

donde $\theta_1 = \log(\beta)$ y $\theta_2 = \log(\gamma)$.

Fue apuntado por Kelly & Ripley (1976) [58] que $c(\theta)$ es finita si y sólo si $\theta_2 \leq 0$. Así, el espacio de parámetros canónicos de la familia es $\Theta = \{\theta \in \mathbb{R}^2 : \theta_2 \leq 0\}$. El caso $\theta_2 = 0$ corresponde a un proceso de Poisson con intensidad β y $\theta_2 < 0$ corresponde a la regularidad de sucesos.

Condicionado al número de puntos, el correspondiente parámetro θ_1 se elimina dando una familia exponencial de un parámetro con estadístico canónico $S(\mathbf{s})$. Ahora $g(\mathbf{s};\theta) = \exp(S(\mathbf{s})\theta_2)$ y $c(\theta)$ es finita para todo $\theta \in \Theta$. El límite de modelos resultantes del proceso de Strauss se puede consultar en Geyer & Thompson (1995) [44].

Procesos Triplets

Al añadir un clique de tercer orden a un proceso puntual de Markov (Ripley & Kelly, 1977 [85]), para obtener un proceso que permita atracciones positivas

entre pares de puntos, se obtiene:

$$W(\mathbf{s}) = \frac{1}{6}\sum_{i=1}^{n}\sum_{j\neq i}\sum_{k\neq i\neq j} I\left[\|s_i - s_j\| \leq r\right] I\left[\|s_j - s_k\| \leq r\right] \cdot I\left[\|s_i - s_k\| \leq r\right] \tag{2.50}$$

que corresponde al número de triplets de puntos que son mutuamente vecinos. Se define $t(\mathbf{s})$ como el vector 3-dimensional $(n(\mathbf{s}), S(\mathbf{s}), W(\mathbf{s}))$, donde $S(\mathbf{s})$ es el mismo que en el proceso de Strauss (2.49). A la familia exponencial que tiene como estadístico canónico $t(\mathbf{s})$ y densidad no normalizada:

$$g(\mathbf{s};\theta) = e^{<t(\mathbf{s}),\theta>} = \exp(n(\mathbf{s})\theta_1 + S(\mathbf{s})\theta_2 + W(\mathbf{s})\theta_3) \tag{2.51}$$

se le llama *proceso triplets.*

La constante normalizadora $c(\theta)$ es finita si y sólo si $\theta_2 \leq 0$ y $\theta_3 \leq 0$ ó si $\theta_3 < 0$. Así, el espacio de parámetros canónicos de la familia puede ser :

$$\Theta = \left\{\theta \in \mathbb{R}^2 : \theta_3 < 0\right\}$$

Proceso de saturación

Al definir para cada punto $s \in \mathbf{s}$

$$m_s(\mathbf{s}) = \sum_{t\in\mathbf{s}\ ,\ t\neq s} I\left[\|s - t\| \leq r\right] \tag{2.52}$$

se obtiene un nuevo modelo de agrupación.

El doble del par estadístico vecino S(**s**) (2.49) es $\sum_{s\in\mathbf{s}} m_s(\mathbf{s})$. Al exigir una cota superior $d > 0$ en la influencia de cualquier punto individual, obtenemos

$$U(\mathbf{s}) = \sum_{s\in\mathbf{s}} \text{mín}(d, m_s(\mathbf{s})) \tag{2.53}$$

Sea $t(\mathbf{s})$ el vector $(n(\mathbf{s}), U(\mathbf{s}))$. A la familia exponencial con estadístico canónico $t(\mathbf{s})$ y densidad no normalizada

$$g(\mathbf{s};\theta) = e^{<t(\mathbf{s}),\theta>} = \exp(n(\mathbf{s})\theta_1 + U(\mathbf{s})\theta_2)$$

se le llama *proceso de saturación.*

Proceso de paso

Al reemplazar la interacción de radio fijo en el proceso de Strauss por una función de paso con constante de interacción en cada intervalo, se obtiene un nuevo proceso de familia exponencial. Sean $0 = r_1 < r_2 < ... < r_p$ con $p \geq 1$ números reales fijos. La función de interacción de tipo de Strauss se define por:

$$e(t) = \begin{cases} \gamma_i \ , \ \text{si } t = \|s_i - s_j\| \in (r_{i-1}, r_i] \ , \ i = 1, ..., p \\ 1 \ , \ \text{en otro caso.} \end{cases}$$

que proporciona la siguiente función de verosimilitud para $\mathbf{s} = (s_1, ..., s_n)$:

$$f(\mathbf{s};\beta, \gamma_1, ..., \gamma_p) = \frac{1}{c(\beta, \gamma_1, ..., \gamma_p)} \beta^{n(\mathbf{s})} \prod_{i=1}^{p} \gamma_i^{S_i(\mathbf{s})} \tag{2.54}$$

donde $S_i(\mathbf{s})$ es el número de pares de vecinos con distancias en $(r_{i-1}, r_i]$, $i = 1, ..., p$. El estadístico canónico es:

$$t(\mathbf{s}) = (n(\mathbf{s}), S_1(\mathbf{s}), ..., S_p(\mathbf{s}))$$

y el espacio de parámetros

$$\theta = (\theta_1 = \log \beta, \theta_2 = \log \gamma_1, ..., \theta_{p+1} = \log \gamma_p)$$

La constante normalizadora será finita para todo $\gamma_i \leq 1$, $i = 1, ..., p$.

Proceso particular

Una función de interacción con parámetro $\theta = (\alpha_1, \alpha_2)$ muy curiosa, se define por:

$$e(t) = 1 + \alpha_1 \exp(-\alpha_2 t^2)$$

El parámetro α_2 proporciona información sobre la intensidad del agrupamiento o regularidad de sucesos y el parámetro α_1 clasifica el patrón en Poisson ($\alpha_1 = 0$), agregado ($\alpha_1 > 0$) o regular ($\alpha_1 < 0$). Se puede encontrar este modelo en Mateu (1995) [65].

Proceso Hard core

Cuando se toma $\gamma = 0$ se obtiene un proceso con probabilidad nula para cada configuración que contiene dos R-puntos vecinos. Es el ejemplo más extremo de inhibición que podemos encontrar, la presencia de un punto anterior a cualquier otro punto prohibe la existencia de éste dentro de la bola de radio R. La función de la interacción $\phi(s,t)$ es de la forma:

$$\phi(s,t) = \begin{cases} 0 & d(s,t) \leq R \\ 1 & d(s,t) > R \end{cases} \qquad s,t \in W$$

Cada configuración tiene la misma probabilidad, demostrando que el proceso hard core es equivalente al proceso de Poisson condicionado al límite mínimo R más bajo de las distancias entre puntos. De hecho, la intensidad condicional es:

$$\lambda(u;\mathbf{s}) = \begin{cases} \beta, & \inf_i d(u,s_i) > R \\ 0, & \inf_i d(u,s_i) \leq R \end{cases}$$

Este proceso es el natural para modelar los centros de las celdas de radio $R/2$ dispuestos aleatoriamente sobre un cuadrado.

Los procesos Diggle y los procesos Very Soft Core

La familia de procesos llamada de Diggle, (Diggle (1986) [33]), tiene como función de interacción (mostrada en la Figura (2.9))

$$\phi(s,t) = \begin{cases} 1-(1-(\frac{d(s,t)}{\theta})^2)^2 & d(s,t) \leq \theta \\ 1 & d(s,t) > \theta \end{cases} \qquad s,t \in W$$

para algún parámetro $\theta > 0$.

Otro ejemplo de regularidad (o repulsión entre los puntos) es el modelo *Very Soft Core* (también llamado modelo *Soft Core*). La función de la interacción es:

$$\phi(s,t) = 1 - \exp\left(-\left(\frac{d(s,t)}{R}\right)^2\right) \qquad s,t \in W$$

y la función de densidad viene dada por:

$$f(\mathbf{s}) = \alpha\beta^{n(s)} \prod_{i<j} \left[1 - \exp\left(-\frac{d(s_i,s_j)^2}{R^2}\right)\right] \quad \mathbf{s} \in W_e$$

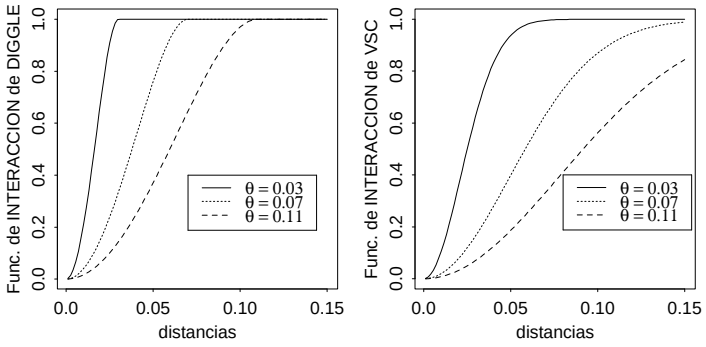

Figura 2.9: Ejemplos de las funciones de interacción de Diggle y de VSC.

Ambos modelos construyen patrones regulares debido a su inhibición en la interacción entre puntos. Una diferencia entre ellos es el rango de interacción, siendo finito en el proceso de Diggle.

2.4.7. Proceso de area-interaction

El proceso de *area-interaction* surgió como una alternativa para el proceso de interacción entre pares de puntos. Generalmente, se define por:

$$f(\mathbf{s}) = \alpha\beta^{n(\mathbf{s})}\gamma^{-A(\mathbf{s})}$$

donde $\beta > 0$ es un paramétro de media, $A(\mathbf{s})$ representa el área de la unión de discos con los centros en los puntos de $\mathbf{s}$ y un radio dado $R > 0$ y $\gamma > 0$ es un parámetro de interacción. El orden de interacción entre los puntos es ilimitado y por la fórmula de inclusión-exclusión:

$$\begin{aligned} A(\mathbf{s}) = &\sum_{i=1}^{n(\mathbf{s})} \mu(B(s_i, R)) - \sum_{i<j} \mu(B(s_i, R) \cap B(s_j, R)) \\ &+ \sum_{i<j<k} \mu(B(s_i, R) \cap B(s_j, R) \cap B(s_k, R)) - \cdots \end{aligned}$$

la influencia de la función de interacción v_k no es trivial para cualquier $k \in \mathbb{N}$.

Se satisface la propiedad de Markov, puesto que

$$\begin{aligned}\frac{f(\mathbf{s} \cup \xi)}{f(\mathbf{s})} &= \beta\gamma^{-[A(\mathbf{s}\cup\xi)-A(\mathbf{s})]} \\ &= \beta\gamma^{-\mu(B(\xi,R)\cap[\cup_i B(s_i,R)]^c)} \\ &= \beta\gamma^{-\mu\left(B(\xi,R)\cap\left[\cup_{s_i\sim\xi} B(s_i,R)\right]^c\right)}\end{aligned}$$

(donde $\sim$ denota la relación de vecindad) para clusters de $B(\xi, R)$ únicamente. La Figura (2.10) muestra en tonos de grises el valor del exponente de la media condicional para una configuración dada.

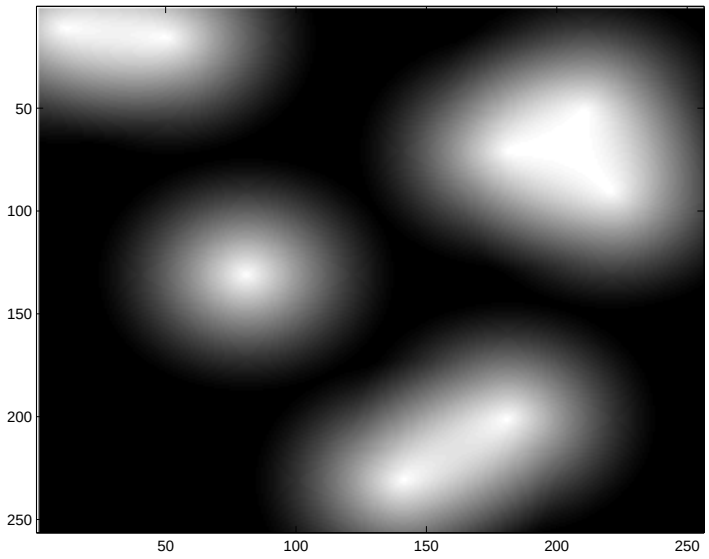

Figura 2.10: Configuración de un proceso de area-interaction con radio $R = 15$ en una región cuadrada de 256×256 y el correspondiente exponente de la función de intensidad condicional. Notar que si $\gamma > 1$, las sombras más claras indican valores más altos de la función de intensidad condicional, e inversamente para $\gamma < 1$.

Los procesos puntuales de area-interaction son convenientes para modelizar procesos biológicos, como distribuciones de población.

2.4.8. Proceso shot-noise

El Proceso *Shot-noise*, generaliza los procesos Strauss y area-interaction. La zona de influencia de una realización particular se considera como una función de influencia. Si $\mathbf{s}$ es una aplicación, $\kappa : W \times W \to \mathbb{R}^+$ es una función no negativa Borel (llamada función de influencia) y a es un punto arbitrario de W,

$$\xi_{\mathbf{s}}(a) = \sum_{i=1}^{n(\mathbf{s})} \kappa(a, s_i)$$

es la influencia de $\mathbf{s}$ hacia a. Para un proceso puntual N, el proceso $\xi_N(a)$ se llama *proceso shot-noise* generado por N en a y con la función de influencia κ.

La función de cobertura es el ejemplo principal de la función ξ. Se trata de tomar como función de influencia $\kappa(a, b) = \mathbf{1}\{a \in Z(b)\}$, donde $Z(b)$ es un conjunto compacto (la zona de influencia generada por b). La función de cobertura,

$$c_{\mathbf{s}}(a) = \sum_{i=1}^{n(\mathbf{s})} \mathbf{1}\{a \in Z(s_i)\}$$

cuenta el número de zonas a (por ejemplo, solapando las bolas).

Una *función potencial* $f : \mathbb{R} \to \mathbb{R}$ se utiliza para definir el proceso shot-noise. Es un proceso puntual en el que la función de densidad se define por:

$$p(\mathbf{s}) = \alpha\beta^{n(\mathbf{s})}\gamma^{-\int_W f(\xi_{\mathbf{s}}(a))d\nu(a)} \qquad \mathbf{s} \in W_e$$

para los parámetros $\beta, \gamma > 0$ y la constante normalizadora $\alpha > 0$. La existencia de este proceso puntual puede asegurarse por la condición de estabilidad de Ruelle que requiere que f satisfaga la desigualdad

$$|f(\xi_{\mathbf{s}}(a))| \leq Cn(\mathbf{s}), \qquad \forall a \in W, \quad \forall \mathbf{s} \in W_e$$

para alguna constante $C > 0$. Para verificar la propiedad de Markov en el sentido de Ripley - Kelly, la condición hereditaria es trivial y debe cumplirse que la proporción:

$$\frac{p(\mathbf{s} \cup \eta)}{p(\mathbf{s})} = \beta \exp\left\{-(\log\gamma)\left\{\int_W f(\xi_{\mathbf{s}\cup\eta}(a))d\mu(a) - \int_W f(\xi_{\mathbf{s}}(a))d\mu(a)\right\}\right\}$$

dependa sólo de los puntos de $\mathbf{s}$ y de los vecinos de η bajo la relación

$$s \sim t \;\text{ sii }\; \{e \in W \;:\; \kappa(e, s) \cdot \kappa(e, t) > 0\} \neq \varnothing.$$

2.4.9. Proceso Candy

Un ejemplo de proceso puntual marcado es el definido como modelo Candy por Stoica, Descombes & Zerubia [95]. Se intenta modelizar situaciones donde los individuos no son puntos sino segmentos. Se obtiene redes como caminos o ríos, obtenidas por fotografía aérea o por satélite. Una realización del proceso es un conjunto de segmentos, que van formando una trama y cada segmento se genera como un punto con tres marcas. Cada 'punto' es $s = (k_s, l_s, \theta_s)$ un triplete donde aparece la posición, la longitud y el ángulo. La densidad funcional del modelo es:

$$f(\mathbf{s}) = \alpha\beta^{n(\mathbf{s})} \left\{ \prod_{i=1}^{n(\mathbf{s})} \exp\left[\frac{l_i - l_{\max}}{l_{\max}} \right] \right\} \gamma_1^{n_f(\mathbf{s})} \gamma_2^{n_m(\mathbf{s})} \gamma_3^{n_r(\mathbf{s})} \gamma_4^{n_o(\mathbf{s})}$$

donde $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in (0, 1)$, $\beta > 0$ son los parámetros del modelo y $n(\mathbf{s})$, $n_f(\mathbf{s})$, $n_m(\mathbf{s})$, $n_r(\mathbf{s})$, $n_o(\mathbf{s})$ representan, respectivamente, el número total de segmentos, el número de los segmentos libres, el número de segmentos que intersectan con otros segmentos por el punto medio, el número de pares de segmentos que se cruzan con un ángulo demasiado agudo y el número de segmentos desorientados. El modelo pone penalizaciones de probabilidad a cada segmento libre y sólo conecta segmentos con ciertas características.

Capítulo 3

Métodos de clustering jerárquico basados en modelos probabilísticos (MBHC)

Indice General

3.1. Introducción 62

3.2. Técnicas basadas en modelos Gaussianos y no Gaussianos 63

3.2.1. Elección del número de clusters 66

3.2.2. Estimación de los parámetros mediante el Gibbs sampler 68

3.2.3. Estrategia para aplicar los métodos Gaussianos 69

3.3. Detección de rasgos en procesos puntuales espaciales mediante métodos MBHC 69

3.3.1. Modelos 70

3.3.2. Algoritmos EM y CEM 70

3.3.3. Estimación del parámetro de forma 72

3.3.4. Elección del número de clusters 73

3.3.5. Software MCLUST 74

3.4. Métodos basados en segmentaciones 75

3.4.1. Introducción 75

3.4.2. Estimación no paramétrica y máximo-verosimilitud de rasgos en procesos puntuales espaciales usando Teselaciones de Voronoï 75
3.4.3. Estimación Bayesiana y segmentación de un proceso puntual usando Teselaciones de Voronoï 77
3.5. Técnicas cluster basadas en distancias al K-ésimo vecino más próximo 78
3.5.1. Introducción 78
3.5.2. Modelización 79
3.6. Simulaciones y aplicaciones 84

3.1. Introducción

Las técnicas MBHC (del inglés Model-based hierarchical Clustering) se basan en la suposición de que los datos vienen generados por mixturas de distribuciones de probabilidad, distinguiendo, cuando sea posible, entre las distribuciones de los rasgos frente a las del ruido (McLachlan & Basford, 1988 [71]; Bensmail, Celeux, Raftery & Robert, 1994 [10]).

Estos métodos consideran a priori una información básica para el desarrollo del modelo, ya que determinará de un modo decisivo la cantidad, la forma, el tamaño y la orientación de los clusters. Si suponemos que los datos proceden de una distribución Normal multivariante o modelo Gaussiano, de los dos parámetros a considerar en dicho modelo (vector de medias y matriz de varianzas-covarianzas), cobra especial importancia la descomposición espectral de la matriz de varianzas-covarianzas Σ. Esta matriz en el grupo k-ésimo se suele reparametrizar de la siguiente forma:

$$\Sigma_k = \lambda_k D_k A_k D_k^{'}, \tag{3.1}$$

donde

1. λ_k denota el primer valor propio de la descomposición espectral de Σ_k y controla el *volumen*,

2. D_k es la matriz de vectores propios de Σ_k y controla la *orientación* de los clusters,

3. A_k es la matriz diagonal de valores propios divididos por k y controla la *forma* de los clusters.

Consideraremos que los datos vienen dados por X, una matriz de n individuos (filas) por p medidas (columnas). Además, denotaremos por $X_i = (x_{i1}, \cdots, x_{ip})'$ el vector asociado al individuo i-ésimo, donde x_{ij} representará la observación i-ésima de la variable j-ésima.

Teniendo en cuenta esta idea central, podemos considerar tres pasos básicos de un método MBHC :

1. El algoritmo ha de inicializarse con una partición inicial.

2. Optimización de esta partición.

3. Utilización de un criterio de parada.

El objetivo de este capítulo es proponer y desarrollar métodos MBHC basados en distancias al K-ésimo vecino más cercano, para la detección de rasgos en imágenes, en presencia de ruido (Mateu & Lorenzo, 2000 [66], 2001 [67], 2002 [68]).

3.2. Técnicas basadas en modelos Gaussianos y no Gaussianos

Supongamos ahora que el modelo de probabilidad asociado a las mixturas de las distribuciones de los datos procede de una distribución Normal multivariante o distribución Gaussiana (Banfield & Raftery, 1993 [8]; Adams, Smith, & Johnson, 1986 [1]; Aitkin, Anderson & Hinde, 1981 [2]; Celeux, & Govaert, 1995 [21]). Asumimos que la población de interés consiste en G subpoblaciones diferentes y que la densidad de una observación x en la subpoblación k-ésima viene dada por $f_k(x;\theta)$ para algún vector θ de parámetros desconocidos. Entonces, $f_k(x;\theta) \simeq MVN(\mu_k, \Sigma_k)$ (MVN denota una distribución normal multivariante) siendo $\theta = (\mu_k, \Sigma_k)$ (Murtagh 1985 [77]; Upton & Fingleton, 1985 [104]).

Asociadas a cada una de las n observaciones o individuos X_i, definimos la variable de etiquetas identificadoras $\gamma \equiv (\gamma_1, ..., \gamma_n)$ donde $\gamma_i = k$ si X_i viene dada

por la subpoblación k-ésima. La función de verosimilitud para los parámetros θ y γ viene dada por:

$$L(X;\theta,\gamma)=\prod_{i=1}^{n}f_{\gamma_i}\left(X_i;\theta\right)=\prod_{k=1}^{G}\prod_{i\in E_k}f_k\left(X_i;\theta\right) \tag{3.2}$$

donde $E_k=\{i:\gamma_i=k\}$. Dado que $f_k\left(x;\theta\right)$ es una distribución Normal multivariante, $f_k\left(x;\theta\right)$ toma la expresión (Tierney & Kadane, 1986 [103])

$$f_k\left(X;\theta\right)=(2\pi)^{-\frac{p}{2}}\left|\Sigma_k\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(X_i-\mu_k\right)'\Sigma_k^{-1}\left(X_i-\mu_k\right)\right\} \tag{3.3}$$

y por tanto, la función de verosimilitud (3.2) viene dada por

$$L(X;\theta,\gamma)=c\prod_{k=1}^{G}\prod_{i\in E_k}\left|\Sigma_k\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(X_i-\mu_k\right)'\Sigma_k^{-1}\left(X_i-\mu_k\right)\right\} \tag{3.4}$$

donde,

1. c es una constante,

2. Un estimador máximo verosímil de μ_k es $\bar{x}_k=n_k^{-1}\sum_{i\in E_k}X_i$.

Como hemos comentado, cada matriz de varianzas-covarianzas Σ_k se reparametriza en la forma

$$\Sigma_k=\lambda_k D_k A_k D_k^{'}$$

donde D_k es la matriz ortogonal de vectores propios de Σ_k, A_k es la matriz diagonal cuyos elementos son proporcionales a los valores propios de Σ_k y λ_k es un escalar y será el primer valor propio de la descomposición espectral de Σ_k.

La *orientación* de las componentes principales de Σ_k viene determinada por D_k, mientras que A_k determina la *forma* de la densidad; λ_k determina el *volumen* o tamaño del correspondiente elipsoide asociado a la distribución, que es proporcional a $\lambda_k^p|A_k|$. La matriz $A_k=diag\left\{\alpha_{1k},\cdots,\alpha_{pk}\right\}$ con $1=\alpha_{1k}\geq\alpha_{2k}\geq\cdots\geq\alpha_{pk}>0$ determina la forma en el siguiente sentido: si los α_{jk} son de magnitud similar, el k-ésimo cluster tenderá a ser esférico, mientras que si $\alpha_{2k}<<1$,

estará concentrado sobre una línea; si $\alpha_{2k} \approx 1$ y $\alpha_{3k} << 1$, estará concentrado sobre un plano bidimensional en un espacio p-dimensional y así sucesivamente.

Las características de forma, volumen y orientación de las distribuciones se estiman a partir de los datos y pueden variar o mantenerse fijas entre los clusters. Así, por ejemplo, si $\Sigma_k = \sigma^2 I = \lambda I$ para todo k, siendo I la matriz identidad, maximizar (3.2) equivale a minimizar la suma de cuadrados dentro de cada grupo, que es el criterio de Ward (1963) [105]. Por tanto, el método de Ward se corresponde con la situación en la que los clusters son hiperesféricos con la misma varianza. Si los clusters no fueran de este tipo, por ejemplo que fueran más alargados o delgados, el método de Ward tendería a romper los clusters en trozos más esféricos. Si $\Sigma_k = \Sigma$, para todo k, obtenemos el método de varianza constante (Friedman & Rubin, 1967 [38]; Wolfe, 1970 [108]).

Si reemplazamos en (3.4) μ_k por su estimador máximo verosímil, obtenemos la log-verosimilitud concentrada

$$l(X;\theta,\gamma) = d - \frac{1}{2}\sum_{k=1}^{G}\left\{tr(W_k\Sigma_k^{-1}) + n_k \log|\Sigma_k|\right\}, \tag{3.5}$$

donde W_k es la matriz muestral de productos cruzados para el cluster k-ésimo

$$W_k = \sum_{i\in E_k}\left(X_i - \bar{x}_k\right)\left(X_i - \bar{x}_k\right)'. \tag{3.6}$$

Hay que hacer notar que $\frac{W_k}{n_k}$ es el estimador máximo verosímil de Σ_k.

El criterio para maximizar la log-verosimilitud (3.5) depende de la forma y de la restricción impuesta sobre la matriz Σ_k. Así, tenemos:

1. Si $\Sigma_k = \sigma^2 I$ (método de Ward), la log-verosimilitud (3.5) se maximiza eligiendo aquel γ que minimiza $tr(W)$ con $W = \sum_{k=1}^{G} W_k$.

2. Si $\Sigma_k = \Sigma$, la log-verosimilitud (3.5) se maximiza eligiendo aquel γ que minimiza $|W|$.

3. Si $\Sigma_k = \lambda_k I$ (criterio que generaliza el de Ward), tenemos densidades esféricas pero con tamaños diferentes de los clusters.

   En este caso, la log-verosimilitud (3.5) se maximiza eligiendo aquel γ que minimiza $\sum_{k=1}^{G} n_k \log tr(\frac{W_k}{n_k})$.

4. Si Σ_k no está restringido de ninguna forma, la log-verosimilitud (3.5) se maximiza eligiendo aquel γ que minimiza $\sum_{k=1}^{G} n_k \log |W_k/n_k|$.

5. Si permitimos que las orientaciones varíen entre los clusters (D_k diferentes) pero manteniendo constante el tamaño ($\lambda_k = \lambda$) y la forma ($A_k = A$) y escribimos la descomposición espectral de W_k como $W_k = L_k \Omega_k L_k^{'}$, donde $\Omega_k = diag\{w_{1k}, \cdots, w_{pk}\}$ con w_{jk} el j- ésimo valor propio de W_k, entonces la log-verosimilitud (3.5) se maximiza eligiendo aquel γ que minimiza $S = \sum_{k=1}^{G} S_k$, con $S_k = tr\left(A^{-1}\Omega_k\right)$.

6. Si permitimos que tanto la orientación (D_k) como el tamaño (λ_k) varíen entre clusters y mantenemos constante la matriz de forma $A_k = A$, $\forall k = 1, ..., G$, la log-verosimilitud (3.5) se maximiza eligiendo aquel γ que minimice $S^* = \sum_{k=1}^{G} n_k \log(\frac{S_k}{n_k})$ (Berger & Sellke, 1987 [12]).

El modelo (3.2) es lo suficientemente general como para abarcar clusters con distribuciones no Gaussianas. Los desarrollos anteriores son válidos en general pero ahora las reparametrizaciones se hacen de forma local. Este tipo de modelos serían útiles en ecología, cuando los datos incluyen localizaciones geográficas de plantas o animales que se pueden agrupar en líneas irregulares (ríos, valles, etc).

Además, el modelo (3.2) se puede generalizar para incluir *ruido*. Normalmente, no todas las observaciones caen dentro de los grupos ya formados; forman por sí solas clusters de un único elemento. Si denotamos por ruido este tipo de observaciones y suponemos que estas observaciones están distribuidas según una Poisson de intensidad ν, entonces si A es el hipervolumen de la región, $E = \cup_{k=1}^{G} E_k$ y $n_0 = n - \sum_{k=1}^{G} n_k$, (3.2) se transforma en:

$$L(X; \theta, \nu, \gamma) = \frac{(\nu A)^{n_0} e^{-\nu A}}{n_0!} \prod_{i \in E} f_{\gamma_i}(X_i; \theta). \tag{3.7}$$

3.2.1. Elección del número de clusters

Normalmente, saber el número exacto de clusters tiene un gran coste de tiempo computacional y por tanto se suele evitar este problema mediante el uso de técnicas Bayesianas (Fraley & Raftery, 1998 [36]).

En este contexto, buscamos estimar el número de clusters, como una elección entre modelos que compiten por los mismos datos. La solución exacta Bayesiana consiste en encontrar la distribución de probabilidad a posteriori $p(G|X)$ de cada

número de clusters G, dados unos datos X. Definiremos el *factor Bayes* de un modelo frente a otro como el cociente entre la probabilidad del primer modelo frente al segundo, cuando ninguno de los dos, a priori, tiene ventajas sobre el otro, es decir,

$$B_{rs} = \frac{p\left(X \mid G = r\right)}{p\left(X \mid G = s\right)}. \tag{3.8}$$

Para encontrar este factor Bayes, hay que definir la probabilidad integrada

$$p\left(X \mid G = r\right) = \sum_{\gamma \in \Gamma} \iint L(X; \theta, \nu, \gamma) p\left(\theta, \nu, \gamma\right) d\theta d\nu \tag{3.9}$$

donde $\Gamma_r = \{0, 1, ..., r\}^n$, $L(X; \theta, \nu, \gamma)$ es la verosimilitud definida en (3.7) y $p\left(\theta, \nu, \gamma\right)$ es la función de densidad conjunta a priori de θ, ν y γ.

Centrando nuestra atención en el cálculo de $B_{r,r+1}$ $(r = 1, ..., n-1)$ hay que tener en cuenta que:

$$p\left(G = r \mid X\right) = \frac{B_{rs_0} p\left(G = r\right)}{\sum_{t=1}^{n-1} B_{ts_0} p\left(G = t\right)} \tag{3.10}$$

donde

1. $B_{rs} = \prod_{t=1}^{s-r} B_{r+t-1,r+t}$ y $B_{sr} = B_{rs}^{-1}$
2. $p\left(G = r\right)$ es la probabilidad a priori de que haya r clusters.

Finalmente, aproximamos el cálculo de $B_{r,r+1}$ $(r = 1, ..., n-1)$ como sigue. En un algoritmo cluster jerárquico, la elección entre $G = r + 1$ y $G = r$ es una decisión entre mezclar o no dos clusters en uno. En el caso de la Normal multivariante p-dimensional, esto es exactamente una comparación estándar de hipótesis anidadas en el modelo lineal general y Smith & Spiegelhalter (1980) [90] demostraron que, en este caso:

$$-2 \log\left(B_{r,r+1}\right) \approx \lambda_r - \left\{\frac{3}{2} + \log\left(p n_{r,r+1}\right)\right\} 2\delta_r = E_r \tag{3.11}$$

donde λ_r es el estadístico del cociente de verosimilitudes, δ_r es la disminución en el número de parámetros causada por pasar de $G = r + 1$ a $G = r$ y $n_{r,r+1}$ es el número de observaciones en el cluster formado. Esta aproximación se suele llamar método AWE (Approximate Weight of Evidence).

3.2.2. Estimación de los parámetros mediante el Gibbs sampler

Como hemos visto, Banfield & Raftery (1993) [8] partieron de la función de verosimilitud (3.2) dependiente de los parámetros μ_k y Σ_k del modelo. Esta expresión es un caso particular del modelo general de mixturas de distribuciones normales con todas las probabilidades π_k iguales. Este modelo general viene dado por:

$$p(X|\theta,\pi) = \prod_{i=1}^{n}\prod_{k=1}^{G}\pi_k f_k\left(X_i;\theta\right) \tag{3.12}$$

con $f_k\left(X_i;\theta\right)$ una función de densidad Normal multivariante y $\pi = (\pi_1, \cdots, \pi_G)$ un vector de probabilidades con $\sum_{k=1}^{G}\pi_k = 1$. El objetivo será estimar el vector de parámetros $\theta = (\mu_1, \cdots, \mu_G, \Sigma_1, \cdots, \Sigma_G)$.

La novedad que se propone es utilizar el GS (*Gibbs Sampler)* para estimar los parámetros de la distribución Gaussiana, sin tener en cuenta las componentes específicas de la matriz de varianzas-covarianzas. Para ello, se proponen los siguientes pasos (Bensmail, 1997 [11]; Younes, (1989) [109]):

1. Simular las variables de clasificación t_{ik} en relación con su probabilidad a posteriori

$$t_{ik} = \frac{\pi_k f\left(X_i\,|\mu_k,\Sigma_k\right)}{\sum_j \pi_j f\left(X_i\,|\mu_j,\Sigma_j\right)}\; i = 1,...,n;\;\; k = 1,...,G \tag{3.13}$$

2. Simular el vector π de probabilidades en relación con su distribución a posteriori condicionada sobre los γ_i.

3. Simular el parámetro θ, del modelo en relación con su distribución a posteriori condicionada sobre los γ_i.

Para encontrar el número de clusters (y también el modelo adecuado), se utiliza una aproximación al factor Bayes por medio del resultado del GS. Para ello, el factor Bayes que se propone para contrastar un modelo M_1 en relación a otro M_0 viene dado por:

$$B_{10} = \frac{p\,(D\,|M_1\,)}{p\,(D\,|M_0\,)} \tag{3.14}$$

donde las probabilidades se calculan como:

$$p\,(D\,|M_k\,) = \int p\,(D\,|\theta_k, M_k\,)\,p\,(\theta_k\,|M_k\,)\,d\theta_k \tag{3.15}$$

θ_k es el vector de parámetros de M_k y $p\,(\theta_k\,|M_k\,)$ es su densidad a priori para $k = 0, 1$.

El cálculo de estas probabilidades es complicado y Bensmail et al. (1997) [11] proporcionan expresiones aproximadas basadas en las cadenas del método GS.

3.2.3. Estrategia para aplicar los métodos Gaussianos

Los pasos para una buena aplicación de los métodos MBHC pueden ser (Fraley & Raftery, 1998 [36]; Fraley & Raftery, 1999 [37]):

1. Determinar el máximo número posible de clusters a considerar (M) y un conjunto de parametrizaciones candidatas de Σ. En general, M ha de ser cuanto más pequeño mejor.

2. Aplicar métodos de cluster jerárquico aglomerativo a aquellas parametrizaciones que lo admitan y obtener las correspondientes clasificaciones.

3. Para cada parametrización y número de clusters $2, ..., M$ aplicar el algoritmo EM empezando con la clasificación obtenida en el paso anterior.

4. Por último, calcular el número BIC para cada parametrización y número de clusters, que nos proporcionará la matriz de valores BIC, de donde obtendremos la parametrización y número óptimo de clusters. Estos valores los podemos representar y un primer máximo local indica la evidencia de ese modelo (Gupta *et al.*, 1986 [48]; Redner & Walker, 1984 [82]).

3.3. Detección de rasgos en procesos puntuales espaciales mediante métodos MBHC

Consideremos el problema de detectar rasgos en imágenes cubiertas por ruido. Por ejemplo los rasgos pueden ser campos de minas (rasgos en general) sobre una

superficie. En este contexto disponemos de una imagen tomada por un avión de reconocimiento (Dasgupta & Raftery, 1998 [29]); (Kaufman & Rousseeuw, 1990 [57]). Después de tratarla, la imagen se reduce a un conjunto de objetos, algunos de los cuales serán minas y otros ruido (como por ejemplo, rocas u objetos metálicos). Los objetos identificados son pequeños y se pueden representar por puntos sin pérdida de información relevante. La tarea del investigador será determinar si hay o no minas y dónde están localizadas (Cox, 1975 [24]). Los procesos puntuales, como procesos estocásticos, pueden dar solución, en combinación con los métodos MBHC, a la detección e identificación de los rasgos y como tal los presentamos en esta sección.

Consideramos un modelo Gaussiano para los rasgos o clusters y un modelo Poisson para el ruido (Gower, 1967 [47]; Marriott, 1975 [63]). Se plantean dos etapas:

1. En la primera, proponemos un modelo jerárquico basado en clustering que proporciona una primera estimación de los rasgos.

2. En la segunda estos clusters se 'refinan' utilizando el algoritmo E.M. (Dempster et al., 1977 [30]). Finalmente, el número de rasgos se encuentra usando una aproximación en la probabilidad de cada número de rasgos.

3.3.1. Modelos

Para la primera etapa, partimos de la ya conocida descomposición espectral de la matriz Σ de varianzas-covarianzas, $\Sigma_k = \lambda_k D_k A_k D_k'$, pero adaptándola al caso que nos ocupa, es decir, con patrones puntuales espaciales (McLachlan & Basford, 1988) [71]). En este caso, la forma de los clusters es la misma, pero su volumen y orientación son diferentes, es decir, $A_k = diag\{1, \alpha\}$, $k = \{1, ..., G\}$ donde $\alpha < 1$ (ya que en un proceso puntual espacial hay dos dimensiones, $d = 2$). Además, $\alpha = \frac{\lambda_2}{\lambda_1}$, $(\lambda_1 > \lambda_2)$. Cuando α es mucho más pequeño que 1, los clusters que resultan tenderán a ser largos y estrechos, mientras que si se acerca a uno, los clusters tenderán a ser de una forma más circular. Por esta razón α se llama el *parámetro forma* (Milligan, 1985 [73]).

3.3.2. Algoritmos EM y CEM

Consideremos ahora el algoritmo EM (Dempster et al., 1977 [30]) como método para la obtención de la estimación máximo verosimilitud en presencia

de datos faltantes. Supongamos una población que tiene por distribución una mixtura de densidades de la forma:

$$f(X;\theta) = \sum_{k=1}^{G} \pi_k f_k(X;\theta) \tag{3.16}$$

donde:

1. $f_k(X;\theta)$ se distribuye como una $MVN(\mu_k, \Sigma_k)$
2. $\sum_{k=1}^{G} \pi_k = 1$

Así, para n observaciones de esta mixtura de distribuciones, los datos completos del proceso puntual espacial serán de la forma $y_i = (x_i, z_i)$, donde $z_i = (z_{i1}, ..., z_{iG})$, con

$$z_{ik} = \begin{cases} 1 & \text{si la } i\text{-ésima observación está en el } k\text{-ésimo cluster} \\ 0 & \text{en otro caso} \end{cases} \tag{3.17}$$

El vector de z_i sigue una distribución multinomial con parámetros $(1; \pi_1, ..., \pi_G)$. Ésto permite calcular la función de verosimilitud completa

$$L(y;\theta) = \sum_{i=1}^{n} \sum_{k=1}^{G} z_{ik} \{\log \pi_k + \log f_k(x_i;\theta)\} \tag{3.18}$$

En el primer paso del algoritmo EM (E-step), hay que calcular el estimador para las etiquetas que hemos puesto a cada observación $\widehat{z}_{ik} = p_{ik} = E(z_{ik} | x_1, ..., x_n; \theta)$, que es la probabilidad a posteriori de que x_i esté en el cluster k-ésimo. El paso siguiente es el M (M-step) que consiste en la obtención de los estimadores máximo verosímiles de θ y π. Para ello, se maximiza la versión estimada de (3.18) dada por:

$$L^*(y;\theta) = \sum_{i=1}^{n} \sum_{k=1}^{G} \widehat{z}_{ik} \{\log \pi_k + \log f_k(x_i;\theta)\} \tag{3.19}$$

Una modificación del algoritmo EM, desarrollada específicamente para la clasificación de modelos, se conoce con el nombre de algoritmo CEM, el cual introduce

un nuevo paso, el C-step, entre el E-step y el M-step, que optimiza el resultado obtenido en el E-step (Celeux & Govaert, 1992 [20]).

El modo de optimizar el E-step se hace mediante el cálculo de

$$z_{ik} = I\left(p_{ik} = \max_{1\leq j\leq G} p_{ij}\right), \tag{3.20}$$

es decir, redondeando las etiquetas calculadas en el paso anterior a cero o uno; los otros dos pasos quedan igual, en el E-step calculamos p_{ik} (probabilidad de que x_{ik} pertenezca al cluster k-ésimo) y en el M-step maximizamos la clasificación completa de la función de verosimilitud.

3.3.3. Estimación del parámetro de forma

El parámetro forma α se estimará por máxima verosimilitud mediante la verosimilitud de clasificación maximizada obtenida por el método CEM, o incluso mejor, mediante la verosimilitud maximizada del modelo de mixturas dada por el algoritmo EM.

Supongamos que tenemos una mixtura de distribuciones Gaussianas que satisfacen $\Sigma_k = \lambda_k D_k A_k D_k^{'}$, con $A_k = A = diag\left\{1, \alpha, ..., \alpha\right\}$, $(k = 1, ..., G)$ y un ruido que se distribuye como un proceso de Poisson homogéneo. Sea n_0 el número de puntos del ruido, d la dimensión de los datos (usualmente $d = 2$) y $\widehat{\Sigma}_k$ la matriz de varianzas-covarianzas muestral para el cluster k-ésimo y cuya descomposición espectral es $\widehat{\Sigma}_k = L_k \Omega_k L_k'$. Si comenzamos a partir de la función de verosimilitud maximizada, asumiendo que α es conocida,

$$\begin{aligned} 2\log L = &-(n-n_0)\left(d\log(2\pi) + d\left(1-\log(d)\right) + \log(|A|)\right) \\ &- d\sum_{k=1}^{G} n_k \log\left(\frac{tr(\Omega_k A^{-1})}{n_k}\right) - 2n_0\log(V) \end{aligned} \tag{3.21}$$

donde V es el volumen ocupado por los datos en R^d. Una buena definición de V viene dada por el volumen del rectángulo más pequeño cuyos lados son paralelos a los ejes coordenados, que contienen a los datos,

$$V = \prod_{j=1}^{d}\left(\max_{i=1,...,n}\left\{x_{ij}\right\} - \min_{i=1,...,n}\left\{x_{ij}\right\}\right) \tag{3.22}$$

Finalmente, maximizando la probabilidad (3.21), con respecto a α, obtenemos

$$\sum_{k=1}^{G} \left[\frac{n_k}{n - n_0} \right] \frac{w_{k2} + ... + w_{kd}}{\alpha w_{k1} + ... + w_{kd}} = 1 - \frac{1}{d} \tag{3.23}$$

La ecuación (3.23) es un polinomio en α de orden G con G distintas raíces como máximo. Sin embargo, sólo nos interesan las que se encuentran en $[0, 1]$. En el caso particular en el que sólo hay un grupo ($G = 1$),

$$\widehat{\alpha} = \frac{w_2 + .. + w_d}{(d-1)\, w_1} \tag{3.24}$$

Por lo general, se resuelve (3.23) mediante un sistema de búsqueda de posibles valores de α en entornos de celda.

3.3.4. Elección del número de clusters

Determinamos el número óptimo de clusters representando cada elección como un modelo estadístico y comparándolos mediante el factor Bayes. Hay una serie de razones que lo avalan (contempla la distribución asintótica en modelos mixtos, considera modelos que se pueden comparar por métodos frecuencialistas, utiliza los p-valores) (Friedman & Rubin, 1967 [38]; Sokal & Michener, 1958 [92]).

Los factores que se necesitan en este problema son cocientes de integrales en elevadas dimensiones que usualmente no son fáciles de estudiar analíticamente. Podríamos aplicar, como en el caso anterior, el método AWE.

El modo de actuar sería introducir varias posibilidades de números de cluster y sobre estas posibilidades aplicar el factor Bayes. Nos quedaríamos con el modelo cuyo número de clusters obtenga un mayor factor. Sin embargo este método no funciona bien cuando hay mucho ruido (Dasgupta & Raftery, 1998 [29]).

Ahora, gracias al algoritmo EM, podemos encontrar la función de verosimilitud maximizada de las mixturas y utilizar un método BIC (Bayesian Information Criterion) (Kass & Raftery, 1995 [56]; Smith & Spiegelhalter, 1980 [90]; Smith & Roberts, 1993 [91]) dado por:

$$2 \log p\,(x\,|G) \approx 2L\left(x; \widehat{\theta}, G\right) - m_G \log n = BIC \tag{3.25}$$

donde

1. $p\,(x\,|G)$, es la probabilidad integrada de los datos dado que hay G clusters,

2. $L\left(x;\widehat{\theta},G\right)$ es la función de verosimilitud maximizada con G clusters,

3. m_G es el número de parámetros independientes a estimar en el modelo con G clusters.

El modelo que proporcione el valor más grande de BIC, será el mejor respecto a este criterio (Sokal & Sneath, 1963 [93]).

3.3.5. Software MCLUST

MCLUST es un software para cluster y análisis discriminante escrito en lenguaje Fortran e implementado en S-PLUS (Rowlingson & Diggle, 1993 [89]). Incluye también las posibilidades de cluster jerárquico, aplicación del algoritmo EM y criterios de información bayesiana (BIC) (Fraley & Raftery, 1999 [37]). Este software es de libre acceso y se encuentra en:

http://www.stat.washington.edu/fraley/mclust/home.html

En MCLUST cada cluster se representa por un modelo Gaussiano:

$$\Phi_k\left(X;\theta\right)=\left(2\pi\right)^{-\frac{p}{2}}\left|\Sigma_k\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(X_i-\mu_k\right)'\Sigma_k^{-1}\left(X_i-\mu_k\right)\right\} \tag{3.26}$$

donde X son los datos y k indica en qué cluster nos encontramos. Los clusters son elipsoidales, centrados en medias μ_k.

Σ_k determina sus otras formas geométricas, de cuya descomposición:

$$\Sigma_k=\lambda_k D_k A_k D_k^{'}$$

se ha hablado al principio del capítulo (Lance & Williams, 1966a [59] ; 1966b [60])

El programa proporciona métodos interactivos EM para máxima verosimilitud, en el E-step, calcula una matriz Z, de tal forma que cada z_{ik} es una estimación de la probabilidad condicional de que la observación i pertenezca al grupo k, dando una estimación del parámetro. En el M-step, calcula la estimación del parámetro.

En el límite, los parámetros convergen en máxima verosimilitud al modelo mixto Gaussiano:

$$\prod_{i=1}^{n} \sum_{k=1}^{G} \tau_k \Phi_k \left(x_i \,|\mu_k, \Sigma_k \right) \tag{3.27}$$

3.4. Métodos basados en segmentaciones

3.4.1. Introducción

Los métodos basados en segmentaciones tienen el mismo objetivo que el análisis cluster (detección de rasgo en presencia de ruido).

En los dos apartados de esta sección se utiliza la misma técnica, pero con enfoques diferentes. Se consideran polígonos de Voronoï (Møller, 1994 [75]), los cuales contienen un elemento de la muestra y de tal forma que recubren el área completa de la imagen a estudio.

El hilo conductor es la detección de rasgos en ruido y todo ello utilizando una técnica novedosa respecto del apartado anterior. En el primer caso el tratamiento será una discriminación de áreas y en el segundo, el cálculo de distancias a una red de puntos simulada (Green & Sibson, 1977 [49]).

3.4.2. Estimación no paramétrica y máximo-verosimilitud de rasgos en procesos puntuales espaciales usando Teselaciones de Voronoï

En este apartado se propone un algoritmo de máxima-verosimilitud para encontrar el dominio de dos variables aleatorias en R^d, donde el dominio de ambas variables es toda la superficie (Allard, 1995 [3]).

El dominio soporte se busca con las teselaciones de Voronoï y como estimador se utiliza la unión de polígonos Voronoï. La eficiencia es del orden de $1/n^2$, asumiendo que para dimensiones altas no es rigurosamente cierto.

Veamos de una forma más detallada el desarrollo. Sean los siguientes elementos:

- K, un compacto de R^d,
- $\{x_i\}$ un conjunto de observaciones aleatorias, con dos variables:
  - U_k, con soporte K
  - $U = A_o$ con soporte A_o, de tal forma que $A_o \subseteq U_k$

- A_o es un conexo, una pieza sin agujeros,
- x es de UA_o con probabilidad p, es decir: $P(x \in U_{A_o}) = p$
- $P(x \in U_k) = 1 - p$
- La función de densidad de x: $f(x) = \frac{p}{a_0} 1_{A_o}(x) + \frac{1-p}{k}$, $x \in K$ donde:
  - a_0 y k son la medida de Lebesgue de A_o y K.
  - k lo podemos suponer 1 sin pérdida de generalidad. Así:

$$f(x) = \frac{p}{a_0} 1_{A_o}(x) + 1 - p, \quad x \in K \tag{3.28}$$

Nota : Si no se conoce p, podemos estimarlo mediante : $\widehat{p} = \frac{m-an}{(1-a)n}$

El problema ahora es que hay que estimar A, donde A es un conjunto finito de polígonos que vienen dados por los datos que contienen: celdas de Voronoï.

* Si es de una dimensión: $\widehat{A}$ es un intervalo.

* Si es de dos dimensiones: $\widehat{A}$ será un rectángulo.

La colección de puntos induce una subdivisión natural de K en "territorios" disjuntos, es decir: asignamos a cada x_i la parte de K más cercana que no se le podría asignar a otro x_j y a esta subdivisión de K, $\nu(c)$ se le llama *Voronoï tessellation*. Con esta construcción de la celda, claramente existe un único punto llamado núcleo de la celda, siendo polítopos convexos: líneas para R^1, polígonos convexos para R^2, y poliedros convexos para R^3 (Chernoff & Rubin, 1954 [23]).

Esta propiedad facilita la computación, ya que de no ser así, el coste computacional es del orden de n^2. Se utiliza en agricultura, astrofísica metalurgia, biología, física... puesto que relaciona el concepto de "área de influencia".

El estimador de máxima verosimilitud para A, se definiría como la unión de polígonos Voronoï, maximizando la probabilidad:

$$L_p\left(\bar{x}, A\right) = \left(\frac{1}{n}\right)^n \left(\frac{m}{a}\right)^m \left(\frac{n-m}{1-a}\right)^{n-m} \tag{3.29}$$

o su logaritmo neperiano:

$$l_p\left(\bar{x}, A\right) = -n \ln n + m \ln \frac{m}{a} + (n-m) \ln \frac{n-m}{1-a} \tag{3.30}$$

con el problema de tener 2^n posibilidades para estos polígonos. Hay que encontrar restricciones en la forma de A, como por ejemplo, encontrar el estimador que sea la unión de las m celdas Voronoï más pequeñas, para algún m.

Además de lo expuesto, se pueden utilizar una serie de métodos para encontrar el estimador para A:

Método 1: Se parte de una solución posible para A. Para cada m de las n posibilidades se considera una aproximación máximo verosímil y se obtienen m celdas (o m pequeños polígonos), denotadas por Am. Transformamos Am quitando pequeñas componentes conexas, rellenando agujeros, redondeando bordes para unirlos y formamos $Am\prime$. La aproximación será obtener el conjunto $Am\prime$, con la mayor probabilidad posible. Este algoritmo recibe el nombre de "algoritmo morfológico".

Método 2: Se inicializa a partir de un conjunto vacío, hasta que todas las celdas se unen y forman A, con una probabilidad que se maximiza en cada iteración, éste será el algoritmo greedy.

Método 3: Combina los métodos anteriores. Primero un algoritmo morfológico para inicializar y después se toma el resultado obtenido como punto de partida del algoritmo greedy.

Se ha propuesto un algoritmo (Chernoff & Rubin, 1954 [23]) que maximiza la probabilidad basada en una teselación natural del espacio dado por los datos de un proceso puntual en presencia de ruido. La probabilidad se fundamenta en la mezcla uniforme del proceso y se trabaja para diferentes dimensiones en el espacio y formas de A_o.

3.4.3. Estimación Bayesiana y segmentación de un proceso puntual usando Teselaciones de Voronoï

El problema, en este caso, sigue siendo detectar rasgos en presencia de ruido usando teselaciones de Voronoï, en un conjunto creado por puntos que provienen de una partición de la región de interés. El modelo que lo genera se estima usando cadenas de Markov con el método de Monte Carlo ($MCMC$) (Hinde & Miles 1980 [53]; Lavine & West, 1992 [61])

El marco de la solución propuesta sería, igual que en el apartado anterior, un conjunto $S = \bigcup_{k=1}^{K} B_k$, donde $\{B_k\}_{k=1}^{K}$ son conjuntos con intersección vacía dos a dos. Las teselaciones de Voronoï, serían una partición de regiones de B_k, de la forma: $\{(u_k, v_k), k = 1, ..., K\} \quad \forall x \in B_k,$ y cumpliendo que $d(x, (u_k, v_k)) < d(x, (u_l, v_l)), l \neq k$.

Para poder aplicar el $MCMC$, tenemos que tener un conjunto creado o simu-

lado de puntos -que no es el proceso puntual- y el algoritmo cambia la forma de la celda de la teselación añadiendo o eliminando estos puntos creados, con cambios estrictamente locales, en términos de relaciones de vecindad, que producen cambios a nivel general (Møller, 1994 [75]).

El proceso puede ser de dos formas: fijando el número de teselas y variando la forma, o con un número de teselas variable si las razones de flexibilidad y movilidad de la muestra lo permiten.

En el primero de los casos, fijamos el número de teselas y variamos la forma del siguiente modo: proponiendo un movimiento de un punto generado a pequeña distancia, posteriormente planteando un movimiento a mayor distancia cambiando el punto por cualquiera de la región. Por último se pueden formular cambios a pequeños grupos de puntos.

En el segundo caso, el de cambiar el número de teselas, se hace porque si el número es pequeño, no se captan detalles en el proceso. Añadir o quitar teselas puede producir cambios en el resultado (Chalmond, 1989 [22]; Diebolt & Robert, 1994 [31]).

3.5. Técnicas cluster basadas en distancias al K-ésimo vecino más próximo

3.5.1. Introducción

Una versión de los métodos cluster jerárquicos basados en modelos adaptados al contexto de procesos puntuales espaciales, es aquella en la que se trabaja con un cierto tipo de distancias entre las observaciones, más concretamente con distancias al k-ésimo vecino más cercano.

Nuevamente supondremos que tenemos unos rasgos (clusters) que detectar sobre los que existe un ruido (observaciones que no pertenecen a ningún rasgo). Problemas de este tipo pueden ser la detección de minas antipersonales aisladas en una imagen de un reconocimiento aéreo, detección de los epicentros en terremotos, etc. (Banfield & Raftery, 1992 [7])

Este problema puede verse también como el problema de buscar subregiones de mayor densidad dentro de una región donde se desarrolla un proceso puntual (Byers & Raftery, 1996 [18], 1997 [19]). La gran ventaja de este método de distancias, con respecto a las versiones más generales de los modelos MBHC, es que en este caso no hace falta hacer ninguna suposición sobre la forma o número de

rasgos.

3.5.2. Modelización

Asumiremos que el ruido está distribuido como un *proceso puntual de Poisson homogéneo* y el rasgo también está distribuido como un *proceso de Poisson* restringido a la imagen y que puede tener cualquier forma.

Así, al considerar un proceso de Poisson homogéneo, podemos encontrar la distribución de la distancia al K-vecino más próximo, D_K, de un punto aleatorio en el proceso $x \in [0, \infty)$

$$P\left(D_K \geq x\right) = \sum_{k=0}^{K-1} \frac{e^{-\lambda\pi x^2}\left(-\lambda\pi x^2\right)^k}{k!} = 1 - F_{D_K}(x) \tag{3.31}$$

Si D_K es mayor que x entonces debe ser uno de los $0, 1, ..., K-1$ puntos en el círculo. Así la densidad $f_{D_K}(x)$ vendrá dada por:

$$\begin{aligned}
f_{D_K}(x) &= \frac{dF_{D_K}(x)}{dx} = \frac{d\left(1 - \frac{e^{-\lambda\pi x^2}\left(-\lambda\pi x^2\right)^k}{k!}\right)}{dx} = \\
&= -\sum_{k=0}^{K-1}\left[\frac{e^{-\lambda\pi x^2}\left(-2\lambda\pi x\right)\left(\lambda\pi x^2\right)^k}{k!} + \frac{e^{-\lambda\pi x^2} k\left(\lambda\pi x^2\right)^{k-1}\left(2\lambda\pi x\right)}{k!}\right] = \\
&= -\sum_{k=0}^{K-1} \frac{e^{-\lambda\pi x^2} 2\lambda^k \pi^k x^{2k-1}\left(-\lambda\pi x^2 + k\right)}{k!} = \\
&= e^{-\lambda\pi x^2} 2 \sum_{k=0}^{K-1} \frac{\left(\lambda\pi\right)^k x^{2k-1}\left(\lambda\pi x^2 - k\right)}{k!} = \\
&= e^{-\lambda\pi x^2} 2 \left[\sum_{k=0}^{K-1} \frac{\left(\lambda\pi\right)^{k+1} x^{2k+1}}{k!} - \sum_{k=0}^{K-1} \frac{\left(\lambda\pi\right)^k x^{2k-1} k}{k!}\right] = \\
&= e^{-\lambda\pi x^2} 2 \left[\sum_{k=0}^{K-1} \frac{\left(\lambda\pi x^2\right)^k \lambda\pi x}{k!} - \sum_{k=0}^{K-1} \frac{\left(\lambda\pi x^2\right)^k x^{-1} k}{k!}\right] = \\
&= e^{-\lambda\pi x^2} 2 \left[\lambda\pi x \sum_{k=0}^{K-1} \frac{\left(\lambda\pi x^2\right)^k}{k!} - \left(\lambda\pi x^2\right) \sum_{k=0}^{K-1} \frac{\left(\lambda\pi x^2\right)^{k-1}}{(k-1)!}\right]
\end{aligned} \tag{3.32}$$

Pero desarrollando los sumatorios anteriores, se observa que van anulándose los términos y la expresión simplificada queda:

$$e^{-\lambda\pi x^2}2\lambda\pi x\left[\frac{(\lambda\pi x^2)^{K-1}}{(K-1)!}\right] = \frac{e^{-\lambda\pi x^2}2\left(\lambda\pi\right)^K x\left(x^2\right)^{K-1}}{(K-1)!} = \\ = \frac{e^{-\lambda\pi x^2}2\left(\lambda\pi\right)^K x^{2K-1}}{(K-1)!} = f_{D_K}(x) \quad (3.33)$$

Si observamos (3.33), vemos que se trata de la función de densidad de una variable aleatoria con una distribución Gamma transformada, $Y \sim \Gamma(K, \lambda\pi)$ donde $Y = (D_K)^2$. Más concretamente es un ejemplo de una *Distribución Gamma Generalizada* (Stacy, 1962 [94]).

Dada esta estructura, encontrar el estimador máximo verosímil de la razón λ es sencillo. Denotamos, por conveniencia, $x_i = d_i$ y tenemos:

$$f_{D_K}\left(d_i;\lambda\right) = \frac{e^{-\lambda\pi d_i^2}2\left(\lambda\pi\right)^K {d_i}^{2K-1}}{(K-1)!}, \quad (3.34)$$

de donde la función de verosimilitud para d_i toma la forma,

$$L\left(d_i;\lambda\right) = \prod_{i=1}^{n} f_{D_K}\left(d_i;\lambda\right) = \frac{e^{-\lambda\pi\sum_{i=1}^{n} d_i^2}\left[2\left(\lambda\pi\right)^K\right]^n \prod_{i=1}^{n} d_i^{2k-1}}{(K-1)!} \quad (3.35)$$

Tomando logaritmos y derivando obtenemos,

$$\ln L\left(d_i;\lambda\right) = \left[-\lambda\pi\sum_{i=1}^{n} {d_i}^2 + n\left[\ln 2 + k\left[\ln\lambda + \ln\pi\right]\right] + \ln\left[\prod_{i=1}^{n} d_i^{2k-1}\right]\right] \\ - \ln\left((k-1)!\right) \quad (3.36)$$

$$\frac{d\ln L\left(d_i;\lambda\right)}{d\lambda} = -\pi\sum_{i=1}^{n} {d_i}^2 + \frac{nk}{\lambda}, \quad (3.37)$$

de donde podemos calcular el estimador máximo verosímil para la intensidad λ,

$$\widehat{\lambda} = \frac{nk}{\pi\sum_{i=1}^{n} d_i^2} \quad (3.38)$$

Algoritmo EM sobre la mixtura de distribuciones

Teniendo en cuenta que el modelo más simple está formado por un rasgo (todos los clusters) y por un ruido (el resto de observaciones), ambos distribuidos como dos procesos de Poisson homogéneos espaciales, podemos decir que la distribución de D_k es una mixtura de las dos distribuciones Gamma Generalizadas anteriormente comentadas

$$D_K \simeq p\Gamma^{(1/2)}(K, \lambda_1\pi) + (1-p)\Gamma^{(1/2)}(K, \lambda_2\pi) \tag{3.39}$$

siendo $\Gamma^{(1/2)}(a, b)$ la distribución de la Gamma Generalizada dada en (3.34).

En realidad disponemos de los siguientes datos $X = \{x_i, y_i, d_i, \delta_i\}_{i=1}^n$ con $\delta_i \in \{0, 1\}$ y $\delta_i = 1$, si el i-ésimo punto se encuentra en alguno de los clusters y $\delta_i = 0$, si se encuentra en el ruido.

Una aplicación del algoritmo EM (Dempster et al., 1977 [30]; Celeux & Govaert, 1992 [20]; Hathaway, 1986 [51]) nos permite obtener estimaciones de los parámetros λ_1 , λ_2 y p. El objetivo de la aplicación de este algoritmo es encontrar el valor de las variables δ_i, $\forall i \in \{1, ..., n\}$, en un cierto número de iteraciones, como resultado de operar los valores obtenidos en la iteración anterior. Los dos pasos de este algoritmo son:

1. **E-step**. En este paso obtendremos el valor esperado de las variables δ_i desconocidas $E(\widehat{\delta}_i^{(t+1)})$, a partir de los valores estimados de $p(t), \lambda_1(t), \lambda_2(t)$ en la iteración anterior.

   Teóricamente el paso es encontrar la función

$$Q(p, \lambda_1, \lambda_2 | X, d) = E(\ln(L(p, \lambda_1, \lambda_2 | d_i, \delta_i)), \tag{3.40}$$

   pero como esta expresión se complica mucho, la calculamos sólo para el individuo i-ésimo:

$$
\begin{aligned}
Q_i(p, \lambda_1, \lambda_2 | d_i, \delta_i) &= \\
&= P(\lambda_1 | d_i, \delta_i) = \frac{p(d_i, \delta_i | \lambda_1)}{p(d_i, \delta_i | \lambda_1) p(\lambda_1) + p(d_i, \delta_i | \lambda_2) p(\lambda_2)} = \\
&= \left[\begin{array}{c} Si : p(\lambda_1) = \widehat{p}\,(t)\,,\ p(\lambda_2) = 1 - \widehat{p}\,(t)\,, \\ p\,(d_i, \delta_i\,|\lambda_1\,) = f_{D_k}(d_i; \widehat{\lambda}_1\,(t)), \\ p\,(d_i, \delta_i\,|\lambda_2\,) = f_{D_k}(d_i; \widehat{\lambda}_2\,(t)) \end{array} \right] = \\
&= \frac{\widehat{p}\,(t)\, f_{D_k}(d_i; \widehat{\lambda}_1\,(t))}{\widehat{p}\,(t)\, f_{D_k}(d_i; \widehat{\lambda}_1\,(t)) + (1 - \widehat{p}\,(t))\, f_{D_k}(d_i; \widehat{\lambda}_2\,(t))} \qquad (3.41)
\end{aligned}
$$

2. **M-step**. En este paso se obtienen los estimadores para λ_1 y λ_2 que en la iteración posterior se utilizarán para realizar el E-step. Sin pérdida de generalidad los calculamos a partir de la función de distribución de las distancias (Gamma Generalizada), calculando la función de verosimilitud, después aplicando el logaritmo neperiano, derivando e igualando a cero, es decir, calculando los estimadores de máxima verosimilitud. El detalle está en elegir la cantidad de observaciones en el rasgo o ruido según calculemos λ_1 ó λ_2, tomando la siguiente estimación de la probabilidad p,

$$\widehat{p}(t+1) = \sum_{i=1}^{n} \frac{\widehat{\delta}_i(t+1)}{n} \qquad (3.42)$$

La función de verosimilitud viene dada por:

$$L = \prod_{i=1}^{n} f_{D_K}(d_i; \lambda) \qquad (3.43)$$

Para λ_1,

$$
\begin{aligned}
L &= \prod_{i=1, \delta_i=1}^{n} f_{D_K}\,(d_i; \lambda) = \prod_{i=1, \delta_i=1}^{n} \frac{e^{-\lambda_1 \pi d_i^2} 2\,(\lambda_1 \pi)^K\, d_i^{2K-1}}{(K-1)!} = \\
&= \frac{e^{\sum_{i=1,\delta_i=1}^{n} \lambda_1 \pi d_i^2 \delta_i}\, 2^{\sum_{i=1,\delta_i=1}^{n} \delta_i}\, (\lambda_1 \pi)^{\sum_{i=1,\delta_i=1}^{n} \lambda_1 \pi d_i^2 \delta_i} \prod_{i=1,\delta_i=1}^{n} d_i^{2K-1} \delta_i}{[(K-1)!]^{\sum_{i=1,\delta_i=1}^{n} \delta_i}} \qquad (3.44)
\end{aligned}
$$

y su logaritmo será,

$$\begin{aligned}\ln L &= \sum_{i=1}^{n}\lambda_1\pi d_i^2\delta_i + \sum_{i=1}^{n}\delta_i\left(\ln 2\right) + k\left(\sum_{i=1}^{n}\delta_i\right)\ln\left(\lambda_1\pi\right) + \\ &+ \sum_{i=1}^{n}\ln\left(d_i^{2K-1}\delta_i\right) - \sum_{i=1}^{n}\delta_i\ln\left(K-1\right)!, \end{aligned} \tag{3.45}$$

de donde al derivar obtenemos

$$\begin{aligned}\frac{\partial\ln L}{\partial\lambda_1} &= 0 = \frac{\partial}{\partial\lambda_1}\left[-\lambda_1\pi\sum_{i=1}^{n}d_i^2\delta_i + K\left(\sum_{i=1}^{n}\delta_i\right)\left[\ln\lambda_1 + \ln\pi\right]\right] = \\ &= -\pi\sum_{i=1}^{n}d_i^2\delta_i + K\left(\sum_{i=1}^{n}\delta_i\right)\frac{1}{\lambda_1}, \end{aligned} \tag{3.46}$$

y finalmente el estimador para λ_1 viene dado por:

$$\widehat{\lambda}_1 = \frac{K\sum_{i=1}^{n}\delta_i}{\pi\sum_{i=1}^{n}d_i^2\delta_i} \tag{3.47}$$

Para λ_2,

$$\begin{aligned}L &= \prod_{i=1,\delta_i=0}^{n} f_{D_K}\left(d_i;\lambda\right) = \prod_{i=1,\delta_i=0}^{n}\frac{e^{-\lambda_2\pi d_i^2}2\left(\lambda_2\pi\right)^K d_i^{2K-1}}{(K-1)!} = \\ &= \frac{e^{\sum_{i=1,\delta_i=0}^{n}\lambda_2\pi d_i^2\delta_i}2^{\sum_{i=1,\delta_i=0}^{n}\delta_i}\left(\lambda_2\pi\right)^{\sum_{i=1,\delta_i=0}^{n}\lambda_2\pi d_i^2\delta_i}\prod_{i=1,\delta_i=0}^{n}d_i^{2K-1}\delta_i}{\left[(K-1)!\right]^{\sum_{i=1,\delta_i=0}^{n}\delta_i}} \end{aligned} \tag{3.48}$$

y su logaritmo será,

$$\begin{aligned}\ln L &= \sum_{i=1}^{n}\lambda_2\pi d_i^2\left(1-\delta_i\right) + \sum_{i=1}^{n}\delta_i\left(\ln 2\right) + \\ &+ k\left(\sum_{i=1}^{n}\left(1-\delta_i\right)\right)\ln\left(\lambda_2\pi\right) + \\ &+ \sum_{i=1}^{n}\ln\left(d_i^{2K-1}\left(1-\delta_i\right)\right) - \sum_{i=1}^{n}\left(1-\delta_i\right)\ln\left(K-1\right)!, \end{aligned} \tag{3.49}$$

de donde al derivar obtenemos

$$\begin{aligned}\frac{\partial \ln L}{\partial \lambda_2} &= 0 = \\ &= \frac{\partial}{\partial \lambda_2}\left[-\lambda_2 \pi \sum_{i=1}^{n} d_i^2 (1-\delta_i) + K\left(\sum_{i=1}^{n}\delta_i\right)[\ln \lambda_2 + \ln \pi]\right] = \\ &= -\pi \sum_{i=1}^{n} d_i^2 (1-\delta_i) + K\left(\sum_{i=1}^{n}(1-\delta_i)\right)\frac{1}{\lambda_2}, \end{aligned} \tag{3.50}$$

y finalmente el estimador para λ_2 viene dado por:

$$\widehat{\lambda}_2 = \frac{K\sum_{i=1}^{n}(1-\delta_i)}{\pi\sum_{i=1}^{n} d_i^2 (1-\delta_i)} \tag{3.51}$$

Notar que para no sobrecargar la notación, en las expresiones anteriores no se ha indicado a qué iteración pertenecían los δ_i (que son los obtenidos en el E-step) y los estimadores. Las expresiones, teniendo en cuenta la iteración quedarían:

$$\widehat{\lambda}_1^{(t+1)} = \frac{K\sum_{i=1}^{n}\widehat{\delta}_i^{(t+1)}}{\pi\sum_{i=1}^{n} d_i^2 \widehat{\delta}_i^{(t+1)}} \tag{3.52}$$

$$\widehat{\lambda}_2^{(t+1)} = \frac{K\sum_{i=1}^{n}\left(1-\widehat{\delta}_i^{(t+1)}\right)}{\pi\sum_{i=1}^{n} d_i^2 \left(1-\widehat{\delta}_i^{(t+1)}\right)} \tag{3.53}$$

El método de clasificación consiste en atribuir una observación como rasgo si su δ_i final se encuentra más cerca del 1 y como ruido en otro caso.

3.6. Simulaciones y aplicaciones

En esta sección, mostramos los resultados obtenidos al aplicar, sobre un mismo conjunto de datos simulados, los diferentes métodos de clustering jerárquico y los métodos basados en distancias entre sucesos.

El conjunto de datos que se utiliza (Figura 3.1) consta de 700 puntos, 300 de los cuales son rasgo y corresponderían a un proceso aleatorio en la región $[0{,}4, 0{,}6] \times [0{,}2, 0{,}6]$, mientras que los 400 puntos restantes son ruido, también definidos como un proceso aleatorio, pero esta vez en toda la región $[0, 1] \times [0, 1]$.

En esta misma Figura, en la segunda gráfica aparece un retículo regular de 900 celdas (30×30). Esta malla colocada sobre los datos, nos permitirá calcular posteriores distancias.

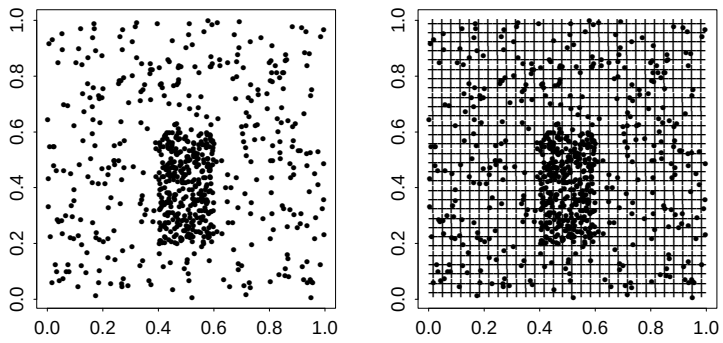

Figura 3.1: Datos simulados (ruido y rasgo) y retículo regular 30×30 sobre los datos

En primer lugar intentamos clasificar el rasgo y ruido por medio de técnicas basadas en cluster jerárquico. En la Figura (3.2) se muestra el resultado de la utilización del software MCLUST aplicando el método inicial aglomerativo o jerárquico. Se observa que los puntos representados por círculos se encuentran aglutinados en la zona donde se encuentra el rasgo; el ruido sería las localizaciones en forma de cuadrados.

Observamos que la clasificación obtenida por este método se ajusta adecuadamente a los datos simulados.

En la Figura (3.3) se muestra el resultado de la aplicación del método jerárquico basado en el algoritmo EM, sobre los datos simulados. Hemos representado las dos clasificaciones por separado para observar claramente lo que se identifica como rasgo y lo que se considera ruido. Observamos que el rasgo tiene una forma más cuadrada, tanto en la parte superior (hay puntos que han sido considerados ruido y no lo son), como en la parte inferior (puntos que son ruido han sido

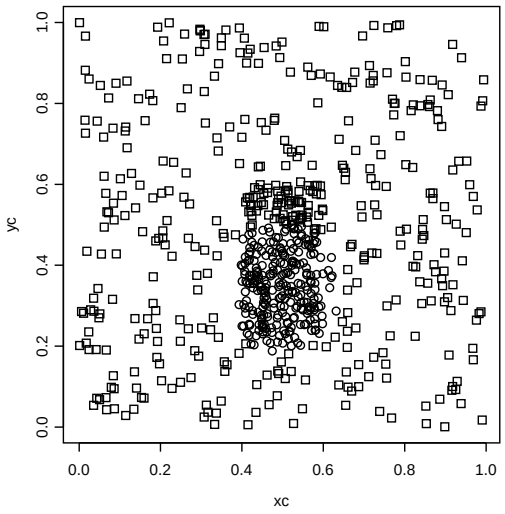

Figura 3.2: Cluster jerárquico (software MCLUST) sobre los datos simulados

considerados rasgo).

Los métodos jerárquicos comentados se basan en 6 tipos distintos de modelos que se diferencian en el tipo de estructura derivada de la descomposición espectral de la matriz de varianzas-covarianzas. En la Figura (3.4) se muestran los valores BIC de las observaciones anteriores, para los seis tipos de modelos.

Al considerar únicamente dos clusters (ruido y rasgo), las líneas son rectas y los máximos se alcanzan, para todos los modelos, en dos grupos diferenciados de puntos, lo que confirma que cualquiera que sea el modelo utilizado, los clusters detectados son dos.

En segundo lugar, aplicamos a los datos simulados diferentes tipos de distancias, las definidas al K-ésimo vecino más cercano y las definidas de celda al K-ésimo vecino más cercano.

En la Figura (3.5) se muestran histogramas de frecuencias absolutas para las distancias calculadas con la función F_{D_K}. En el eje de abcisas, se representa la distancia al K-ésimo vecino más próximo, mientras que en el de ordenadas aparece la frecuencia absoluta de la cantidad de observaciones que se aglutinan alrededor de una distancia dada.

Se observa que al aumentar el valor de K (es decir, al aumentar el número del

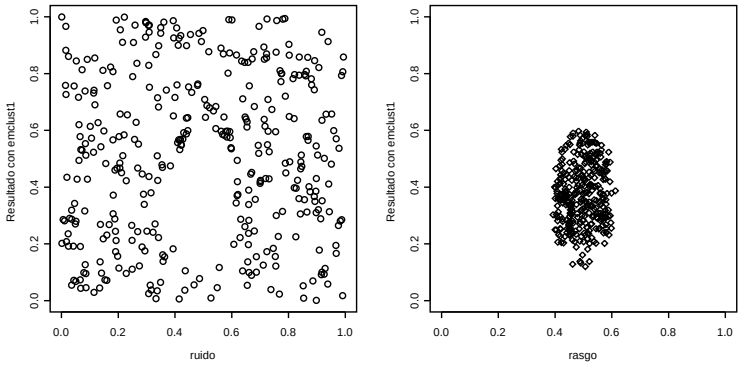

Figura 3.3: Clasificación en rasgo y ruido obtenida por el método de cluster jerárquico en combinación con el algoritmo EM

vecino más próximo que tomaremos como referencia para calcular la distancia), la bimodalidad en el histograma se hace más evidente.

Las distancias alrededor de la primera moda (distancias más pequeñas), corresponden a los valores de las distancias de los puntos del rasgo, mientras que la segunda moda corresponde a los valores de las distancias de los puntos del ruido. De esta forma las distancias al K-ésimo vecino más cercano sirven para discriminar entre rasgo y ruido.

Las siguientes tablas muestran algunas medidas descriptivas de centralización tanto para el rasgo (primera tabla), como para el ruido (segunda tabla) de los datos simulados. En este caso se han tomado distancias de celda al K-ésimo vecino más próximo con $K = 1, 3, 5, 10$. Estas descripciones muestran claras diferencias entre distancias del ruido y rasgo, más evidentes al aumentar el valor de K.

nndistFk (rasgo)	Min	Q_{25}	Mediana	Media	Q_{25}	Max
$K = 1$	0.001071	0.005459	0.008011	0.008237	0.010858	0.01823
$K = 3$	0.005813	0.01134	0.01338	0.01434	0.01675	0.02709
$K = 5$	0.008987	0.01613	0.01923	0.01908	0.02138	0.03088
$K = 10$	0.01917	0.02605	0.0281	0.02824	0.03102	0.04037

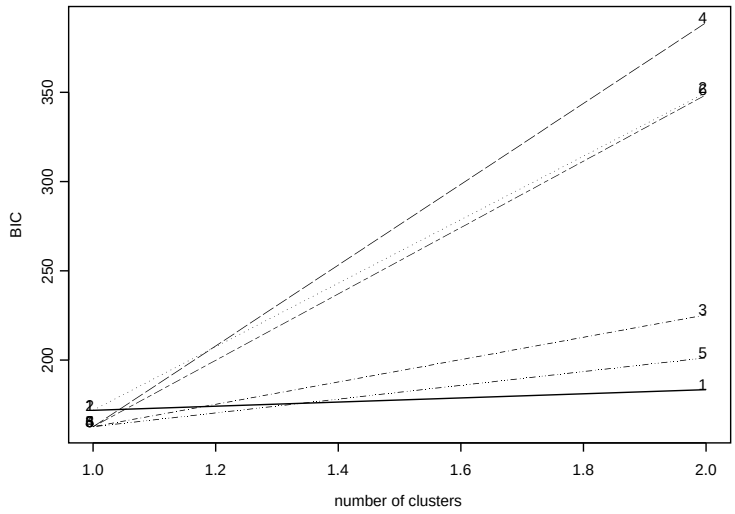

Figura 3.4: Representación de los valores BIC para los datos simulados

nndistFk (ruido)	Min	Q_{25}	Mediana	Media	Q_{25}	Max
$K = 1$	0.0008557	0.01575	0.0234	0.02528	0.03301	0.07655
$K = 3$	0.008291	0.03625	0.04668	0.04744	0.05698	0.104
$K = 5$	0.02069	0.05135	0.06182	0.06286	0.07294	0.1329
$K = 10$	0.03391	0.07633	0.08909	0.0903	0.1019	0.1731

La Figura (3.6) muestra histogramas referentes a las distancias de datos del retículo regular al (K=3)-ésimo vecino más cercano presentado en la Figura (3.7). En las tres gráficas de esta Figura, se toma como distancia los resultados de calcular el espacio que hay entre el punto central de cada cuadrícula del retículo y el tercer vecino más próximo.

Notar que el tercer histograma es el resumen de los datos anteriores y no se aprecia tanto la bimodalidad (que representaría el rasgo y el ruido), aunque al hacer la representación por separado, se observa claramente la diferencia. Así, en el primer histograma, el rango para la distancia es de 0.005 a 0.025 y observamos una moda clara. Al variar el rango de distancia de 0.02 a 0.1, observamos otra moda. La primera representa el rasgo, mientras que la segunda representa el ruido.

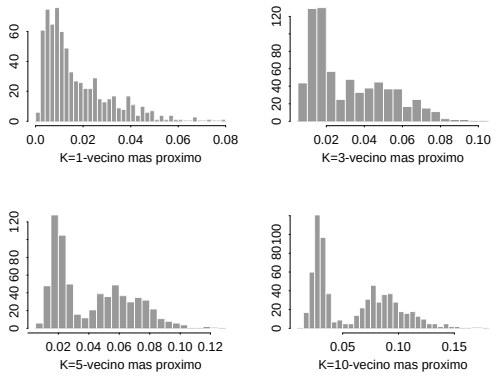

Figura 3.5: Histogramas para la distancia F_{D_K} con $K = 1, 3, 5, 10$

En la Figura (3.7) repetimos el proceso anterior pero para el (K=10)-ésimo vecino más próximo y en este caso, en la tercera gráfica, sí aparece una bimodalidad clara.

En tercer lugar, aplicamos la distancia al K-ésimo vecino más próximo para detectar, mediante el algoritmo EM, tanto el rasgo como el ruido.

Los resultados para $K = 5, 10, 15$ se presentan en las Figuras (3.8), (3.9) y (3.10). Un aumento del valor de K conlleva a una mejor clasificación. En cada gráfica se presentan los histogramas de las distancias, así como un mapa de probabilidad de pertenencia al rasgo.

Para reforzar este análisis, hemos simulado una situación en la que hay dos rasgos y ruido (ver Figuras 3.11-3.13). Nuevamente se observa que un aumento de K conduce a una mejor clasificación en la que podemos discernir claramente los rasgos del ruido.

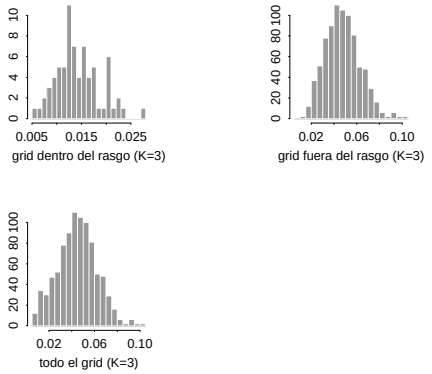

Figura 3.6: Histogramas para la distancia de celda regular al tercer vecino más próximo

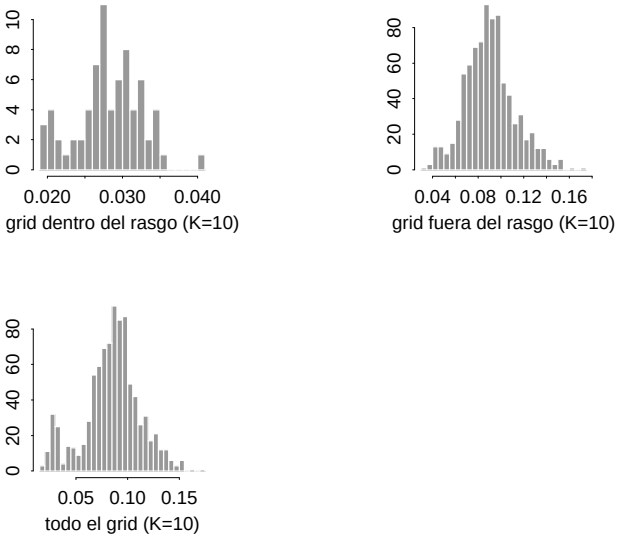

Figura 3.7: Histogramas para la distancia de celda regular al décimo vecino más próximo

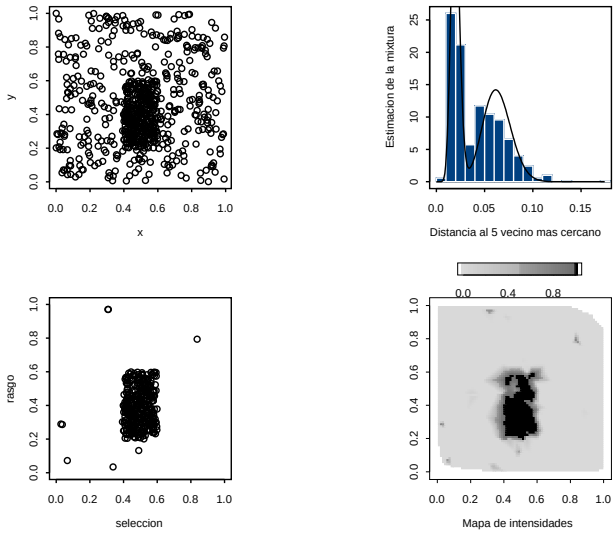

Figura 3.8: Detección de un rasgo con distancias al 5^o vecino más próximo

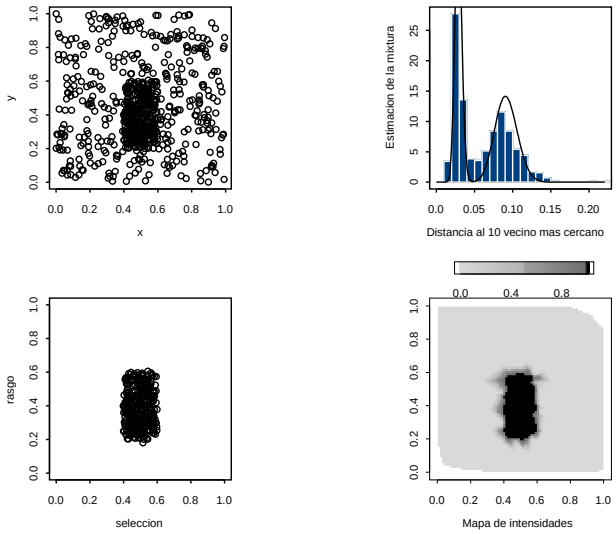

Figura 3.9: Detección de un rasgo con distancias al 10^o vecino más próximo

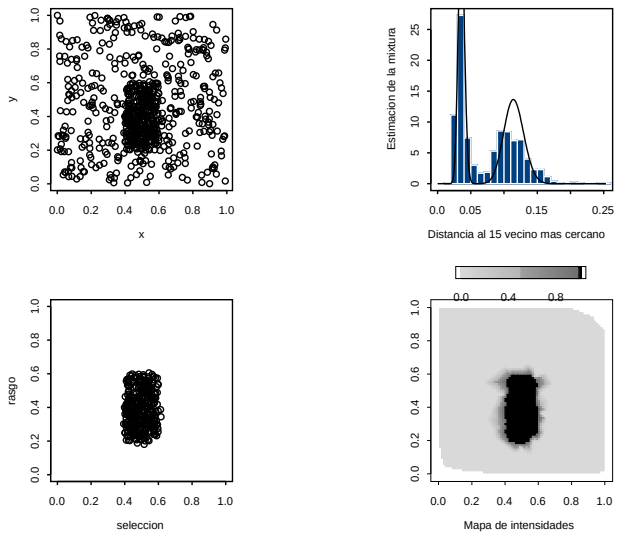

Figura 3.10: Detección de un rasgo con distancias al 15^o vecino más próximo

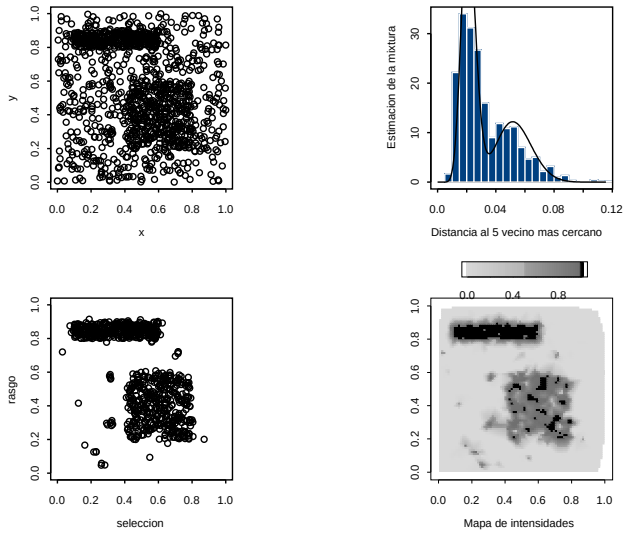

Figura 3.11: Detección de dos rasgos con distancias al 5^o vecino más próximo

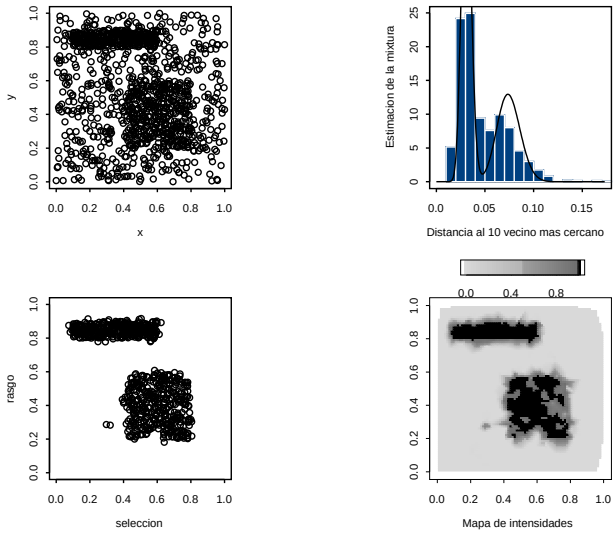

Figura 3.12: Detección de dos rasgos con distancias al 10^o vecino más próximo

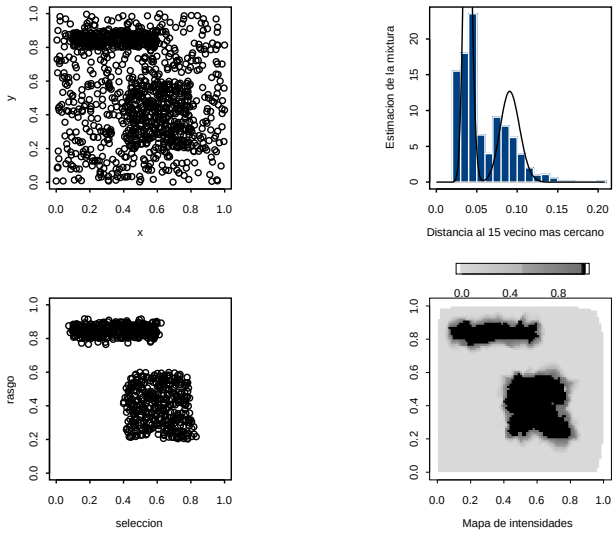

Figura 3.13: Detección de dos rasgos con distancias al 15^o vecino más próximo

Capítulo 4

Fundamentos teóricos sobre funciones LISA locales

Indice General

4.1. Introducción 95

4.2. Funciones individuales versus globales 96

4.3. Distribución teórica 99

4.1. Introducción

En la exposición del segundo capítulo, se exponen medidas globales para un proceso puntual. Estas medidas proporcionan información global de todo el patrón. La K función, la densidad producto de segundo orden, distribución de Palm y otras tienen este cometido.

Un paso más allá, sería buscar medidas de carácter individual que permitieran estudiar el comportamiento global del patrón a partir de características locales. Es en este capítulo, donde, además se van a estudiar estas medidas individuales para cada observación, dando el soporte teórico necesario para llevarlo a cabo.

La metodología que proponemos se basa en la construcción de funciones locales que recogen características de segundo orden alrededor de cada individuo. Son

las llamadas funciones LISA, *"Local Indicator of Spatial Assocation "* (Cressie & Brant, 1997 [27]; Cressie & Brant, 2001 [28]). A cada individuo se le asignará un vector, cuyas componentes, serán las funciones LISA individuales, evaluadas en los diferentes radios.

El objetivo es establecer medidas válidas que cuantifiquen la distancia entre las funciones LISA individuales, de forma que podamos distinguir cuáles de ellas están próximas entre sí. Agrupaciones de éstas se han de corresponder con agrupaciones de los individuos que las provocaron. Habrá que establecer pues, el mecanismo que aisle, en función de las agrupaciones de las funciones, las agrupaciones de los individuos (rasgos y ruido).

4.2. Funciones individuales versus globales

Consideramos un proceso puntual en $A \subseteq R^2$ con un área $\nu(A) = |A| = a$. Llamaremos $N(A) = \{\textit{número de puntos del proceso en } A\}$. Consideraremos la función K (Ripley, 1987 [87]), para un proceso puntual estacionario e isotrópico y $\lambda K(r) \equiv E[N(b(0,r)) \setminus \{0\} : 0 \in N]$, donde $b(0,r)$ es un disco de radio r centrado en 0, $r > 0$. $K(r)$ proporciona una medida de una estructura de dependencia espacial de un proceso puntual. De esta forma, $\lambda^2 a K(r)$ es el número esperado de pares ordenados de sucesos en la región A.

La K-función es una función acumulativa y con su derivada podemos estudiar el valor de:

$$\rho(r) \equiv \frac{\lambda^2 K'(r)}{2\pi r}; \quad r > 0 \tag{4.1}$$

expresión que corresponde a la función densidad producto. La densidad producto ya fue definida en el Capítulo 2 y denotada por $l^2_{N_0}(r)$. En esta sección, por sencillez de notación, $l^2_{N_0}(r)$ se denotará por $\rho(r)$.

En este apartado vamos a definir funciones locales indicadoras de asociación espacial (funciones LISA, Brant & Cressie, 1997 [17]; Stoyan, Kendall & Mecke, 1987 [99]). Cuando estimamos datos de un proceso puntual, $\{x_1, x_2, ..., x_{N(A)}\}$, la función densidad producto empírica, $\widehat{\rho}(r)$, describe la estructura de interacción del proceso. Por ejemplo, grandes valores de $\widehat{\rho}(r)$ y pequeños de r, indican gran abundancia de pequeños intervalos entre las distancias de puntos, es decir, los datos forman un cluster.

Consideramos las funciones LISA, de manera análoga a la "función local" descrita por Anselin (1995) [5] y llamada *local indicators of spatial association (LISA).* Una función densidad producto individual LISA $\widehat{\rho}^{(i)}(r)$ debe manifestar la contribución del elemento x_i en la estimación global $\widehat{\rho}(r)$.

Una estimación de la función densidad producto a partir de la estimación kernel de $\frac{\lambda^2 K'(r)}{2\pi r}$ se encuentra en Stoyan, (1987 [96]). Recordar que $\lambda^2 K'(r)$ es el número esperado (por unidad de área), de pares ordenados de elementos con distancia entre ellos menor o igual que r.

Una estimación kernel de la densidad $\lambda^2 K'(r)$ (ignorando el efecto de los bordes), viene dada por:

$$\widehat{\lambda^2 K'(r)} \equiv a^{-1} \sum_{i=1}^{n} \sum_{j \neq i} f_\varepsilon \left(\|s_i - s_j\| - r \right); r > \varepsilon > 0 \tag{4.2}$$

para $\{x_1, x_2, ..., x_n\}$ observados en la región $A \subseteq R^2$ de área $\nu(A) = |A| = a$, donde $\|s_i - s_j\|$ denota la distancia euclídea entre s_i y s_j. Así, f_ε es la función kernel análoga a las funciones kernel utilizadas para estimar la densidad de probabilidad. La función kernel proporciona pesos que determinan la contribución de cada distancia entre los elementos, a la distancia estimada a cualquier distancia r. Notar que se subindica con ε, para indicar la banda de confianza propuesta en la estimación de la función densidad. Para ser precisos: $f_\varepsilon(s) = \frac{1}{\varepsilon} f\left(\frac{s}{\varepsilon}\right)$.

A partir de ahora utilizaremos el kernel de Epanechnikov, por su uso extendido en la teoría de procesos puntuales (Stoyan, Kendall & Mecke, 1995 [99]):

$$f_\varepsilon(s) = \begin{cases} \frac{3}{4}(1 - s^2) & para \quad -1 \leq s \leq 1 \\ 0, & resto \end{cases} \tag{4.3}$$

Análogo a la construcción de histogramas para datos univariantes, la estimación de la densidad de la función kernel se construye por la asignación alta de pesos entre las distancias. Para la estimación del kernel de la densidad producto, tomaremos, para los puntos s_j los conjuntos $r - \varepsilon \leq \|s_i - s_j\| \leq r + \varepsilon$ y $r > \varepsilon$.

Ignorando los efectos de borde, un conjunto es justamente la región entre dos círculos de radios $r + \varepsilon$ y $r - \varepsilon$ centrados en s_i nombrado anteriormente y donde el área es $4\pi r\varepsilon$, tal y como se muestra en la Figura (4.1)

Así, una estimación empírica de la densidad producto global, viene dada por:

$$\begin{aligned} \widehat{\rho}_{\varepsilon}(r) &= \frac{1}{2\pi r a}\sum_{i=1}^{n}\sum_{j\neq i} f_{\varepsilon}\left(\|s_i - s_j\| - r\right) \\ con\ \ r &> \varepsilon > 0 \quad y \quad r-\varepsilon \leq \|s_i - s_j\| \leq r+\varepsilon \end{aligned} \tag{4.4}$$

Para introducir la función densidad producto local y por su relación con la K-función local, partimos de ésta para su definición. Sea un individuo cualquiera del patrón, $s_i \in N$. El número esperado de sucesos con distancia r del i-ésimo individuo y calculado respecto a la medida reducida de Palm (Stoyan, Kendall & Mecke, 1995 [99]), viene dado por:

$$\{\lambda K(r)\}^{(i)} \equiv E\left[N(b(s_i,r)\backslash\{s_i\})\,|s_i \in N\right]; r > 0 \tag{4.5}$$

Además:

$$\left\{\lambda\widehat{K'(r)}\right\}^{(i)} \equiv \sum_{j\neq i} f_{\varepsilon}(\|s_i - s_j\| - r); r > \varepsilon > 0 \tag{4.6}$$

Ahora, para un proceso de Poisson homogéneo, se obseva que $\frac{n-1}{a}$ proporciona un estimador insesgado para λ bajo un proceso reducido del Palm, $N^{!}_{s_i}$, dado que este proceso tiene la misma distribución de probabilidad que el proceso original, N (Stoyan, Kendall, & Mecke, 1995 [99]). En particular, $E_{!}[N(B)] \equiv E[N^{!}_{s_i}(B)] = \lambda\nu(B)$, para un conjunto Borel $B \subset A$, donde la notación $E_{!}[\cdot]$ indica la esperanza de la medida reducida de Palm. La insesgadez de $\frac{n-1}{a}$ para λ es consecuencia de:

$$\begin{aligned} E_{!}[N(A)-1] &= E_{!}\left[\left(\sum_{j\neq i} 1\right)\right] = E\left[\int_A N^{!}_{s_i}(dx)\right] = \\ &= \int_A E\left[N^{!}_{s_i}(dx)\right] = \int_A E\left[N(dx)\right] = \int_A \lambda dx = \lambda a \end{aligned} \tag{4.7}$$

Combinando la estimación del kernel con la estimación de λ, obtenemos:

$$\widehat{\lambda}\left\{\lambda\widehat{K'(t)}\right\}^{(i)} = (n-1)a^{-1}\sum_{j\neq i} f_{\varepsilon}(\|s_i - s_j\| - r); r > \varepsilon > 0 \tag{4.8}$$

que proporciona una estimación de la densidad del kernel, para la derivada $\{\lambda^2 K'(r)\}^{(i)}; t > 0$. Por analogía con la formación de la estimación de la densidad producto, una versión empírica de la función densidad producto local es,

ignorando los efectos de borde:

$$\widehat{\rho}_{\varepsilon}^{(i)}(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} f_{\varepsilon}(\|s_i - s_j\| - r)\,; r > \varepsilon > 0 \tag{4.9}$$

La estimación dada anteriormente, es análoga a las "funciones individuales" definidas por Stoyan & Stoyan (1994) [98]. Para un r fijo, $\widehat{\rho}_{\varepsilon}^{(i)}(r)$ cumple la definición que se requiere de una función LISA, es decir, la suma de las funciones densidad producto LISA individuales son proporcionales a la función global, así, fijando r, tenemos que:

$$\begin{aligned} \widehat{\rho}_{\varepsilon}(r) &= \frac{1}{2\pi r a} \sum_{i=1}^{n} \sum_{j \neq i} f_{\varepsilon}(\|s_i - s_j\| - r) = \\ &= \frac{1}{n-1} \sum_{i=1}^{n} \frac{n-1}{2\pi r a} \sum_{j \neq i} f_{\varepsilon}(\|s_i - s_j\| - r) = \\ &= \frac{1}{n-1} \sum_{i=1}^{n} \widehat{\rho}_{\varepsilon}^{(i)}(r) \end{aligned} \tag{4.10}$$

Por tanto, llegamos a que

$$\widehat{\rho}_{\varepsilon}^{(i)}(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} f_{\varepsilon}(.) \tag{4.11}$$

A la función $\widehat{\rho}_{\varepsilon}^{(i)}(r)$ se le llama función LISA y es la función clave, ya que es la base de los futuros desarrollos y aplicaciones.

4.3. Distribución teórica

En esta sección vamos a desarrollar las expresiones de la esperanza y la varianza de la función individual LISA (4.19), (4.22), así como la covarianza entre dos funciones LISA individuales (4.24).

Nótese que los cálculos de la esperanza y la varianza se hacen condicionados a la existencia de un punto s_i y se calculan sobre la medida de Palm reducida (se subindica con una admiración).

Cuando las expresiones se refieren a dos observaciones, la notación que se utiliza es doble admiración (4.26) y (4.27).

Ciertamente para un s_i situado cerca del borde de la región observada A del plano, la función densidad producto LISA, estimará la contribución del s_i a la densidad producto global de función (Anselin, 1995 [5]). Condicionando una observación al suceso s_i, podemos calcular la esperanza con respecto al proceso de Palm, para un $r > \varepsilon > 0$

$$\begin{aligned} E_! \left[\widehat{\rho}_\varepsilon^{(i)}(r)\right] &= \frac{1}{2\pi r a} E_! \left[(N(A)-1) \sum_{j \neq i} f_\varepsilon (\|s_i - s_j\| - r) \right] = \\ &= \frac{1}{2\pi r a} E_! \left[E \left\{ (n-1) \sum_{j \neq i} f_\varepsilon (\|s_i - s_j\| - r) N(A) = n \right\} \right] = \\ &= \frac{1}{2\pi r a} E_! \left[(N(A)-1)^2 \right] \frac{1}{a} \int_A f_\varepsilon (\|s_i - s\| - r)\, ds = \\ &= \frac{\lambda^2 + \frac{\lambda}{a}}{2\pi r} \int_{r-\varepsilon}^{r+\varepsilon} f_\varepsilon (t-r) \left| \delta b(s_i, t) \bigcap A \right| dt \end{aligned} \tag{4.12}$$

donde $|\delta b(s_i, t) \bigcap A|$ da una medida del perímetro del círculo de radio r, centrado en s_i que está contenido en la región A. Aquí usamos el hecho de que para un proceso de Poisson homogéneo, condicionar una observación de $N(A) = n$, s_i, a un conjunto de sucesos $s_1, ..., s_{i-1}, s_{i+1}, ..., s_n$ es independiente e idénticamente distribuido en A. Si sobre (4.10) utilizamos la corrección de borde global de Ripley (1987) [87] obtenemos la expresión:

$$\widehat{\rho}_\varepsilon (r) = \frac{1}{2\pi r a} \frac{2\pi \|s_i - s_j\|}{|\delta b(s_i, \|s_i - s_j\|) \bigcap A|} \sum_{i=1}^{n} \sum_{j \neq i} f_\varepsilon (\|s_i - s_j\| - r) \tag{4.13}$$

Y esta misma corrección y expresión, nos lleva a la ecuación de estimación de la densidad producto local o individual:

$$\widehat{\rho}_\varepsilon^{(i)}(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} \frac{2\pi \|s_i - s_j\|}{|\delta b(s_i, \|s_i - s_j\|) \bigcap A|} f_\varepsilon (\|s_i - s_j\| - t)\,; r > \varepsilon > 0 \tag{4.14}$$

Sin embargo, teniendo en cuenta otro tipo de correcciones tenemos otras posibles expresiones para las funciones LISA:

$$\widehat{\rho}_{\varepsilon}^{(i)}(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} \frac{f_{\varepsilon}(.)}{c_X(r)} \tag{4.15}$$

y

$$\widehat{\rho}_{\varepsilon}^{(i)}(r) = \frac{n-1}{2\pi r a} \sum_{j \neq i} \frac{f_{\varepsilon}(.)\, b_{ij}}{\nu\left(X^{\|s_j - s_i\|}\right)} \tag{4.16}$$

siendo las correcciones de borde utilizadas para estas expresiones:

$$c_X(r) = \sum_{j>i} \Phi(\|s_i - s_j\|; \theta) \tag{4.17}$$

$$\nu\left(X^{\|s_j - s_i\|}\right) = \frac{1}{2} \sum_{j \neq i} w_{ij}^{-1} \Phi(\|s_i - s_j\|; \theta) \tag{4.18}$$

Para $s_i \in A$ y $i = 1, ..., n$ y partiendo de (4.12) la esperanza de la función LISA toma la forma:

$$\begin{aligned} E_! \left[\widehat{\rho}_{\varepsilon}^{(i)}(r)\right] &= \frac{\lambda^2 + \frac{\lambda}{a}}{2\pi r} \int_{r-\varepsilon}^{r+\varepsilon} \frac{2\pi s}{|\delta b(s_i, s) \bigcap A|} f_{\varepsilon}(s-t) \, |\delta b(s_i, s)| \, ds = \\ &= \frac{\lambda^2 + \frac{\lambda}{a}}{2\pi r} \int_{r-\varepsilon}^{r+\varepsilon} 2\pi s ds f_{\varepsilon}(s-t) \, ds = \\ &= \lambda^2 + \frac{\lambda}{a} \end{aligned} \tag{4.19}$$

para un proceso homogéneo de Poisson, desarrollamos ahora la varianza de la densidad producto individual en su versión corregida de borde:

$$\begin{aligned} Var_! \left[\widehat{\rho}_{\varepsilon}^{(i)}(r)\right] &= E_! \left[\left\{\left(\widehat{\rho}_{\varepsilon}^{(i)}(r)\right)^2 | N(A) - n\right\}\right] - E_! \left[\widehat{\rho}_{\varepsilon}^{(i)}(r)\right]^2 = \\ &= \frac{1}{(2\pi r a)^2} E_! \left[E\left\{(n-1)^2 \left(\sum_{j \neq i} g(s_i, s_j, r)\right)^2 | N(A) = n\right\}\right] \\ &\quad - \left(\lambda^2 + \frac{\lambda}{a}\right)^2 \end{aligned} \tag{4.20}$$

Tomando $g(s, y, r) = \frac{2\pi \|s-y\|}{|\delta b(s, \|s-y\|) \bigcap A|} f_{\varepsilon}(\|s - y\| - r)$ y dado que

$$E_{!}\left[E\left\{(n-1)^2\left(\sum_{j\neq i} g\left(s_i,s_j,r\right)\right)^2 |N(A)=n\right\}\right]=$$

$$=E_{!}\left[E\left\{(n-1)^2\left(\begin{array}{c}\sum_{j\neq i} g\left(s_i,s_j,r\right)+\\ \sum_{j\neq}\sum_{\substack{k\neq i\\ k\neq j}} g\left(s_i,s_j,r\right)g\left(s_i,s_j,r\right)\end{array}\right)|N(A)=n\right\}\right]=$$

$$=E_{!}\left[(N(A)-1)^2\left\{\begin{array}{c}(N(A)-1)\,E[g\left(s_i,s,r\right)^2]\\ +(N(A)-1)(N(A)-2)(E[g\left(s_i,s,r\right)])^2\end{array}\right\}\right] \quad (4.21)$$

la expresión final de la varianza será

$$Var_{!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r\right)\right]=\frac{\lambda^3+3\lambda^2/a+\lambda/a^2}{2\pi r^2}\int_{r-\varepsilon}^{r+\varepsilon}\frac{2\pi s^2 f_{\varepsilon}^2\left(s-r\right)}{|\delta b\left(s_i,s\right)\bigcap A|}ds+3\left(\lambda^3/a+\lambda^2/a^2\right) \quad (4.22)$$

La covarianza para la función LISA, evaluada en dos distancias es:

$$\begin{aligned}Cov_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right),\widehat{\rho}_{\varepsilon}^{(i)}\left(r_2\right)\right] &= \frac{\lambda^3+3\lambda^2/a+\lambda/a^2}{\left(2\pi\right)^2 r_1 r_2}\int_A g\left(s_i,s,r_1\right)g\left(s_i,s,r_2\right)ds\\ &\quad +3\left(\lambda^3/a+\lambda^2/a^2\right)\end{aligned} \quad (4.23)$$

La covarianza para un estimador de la función densidad producto LISA, evaluada en dos distancias y para dos individuos es ($t_2>t_1>\varepsilon>0$):

$$\begin{aligned}Cov_{!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right),\widehat{\rho}_{\varepsilon}^{(i)}\left(r_2\right)\right] &= \frac{\lambda^3+3\lambda^2/a+\lambda/a^2}{\left(2\pi\right)^2 r_1 r_2}\int_A g\left(s_i,s,r_1\right)g\left(s_i,s,r_2\right)ds\\ &\quad +3\left(\lambda^3/a+\lambda^2/a^2\right)\end{aligned} \quad (4.24)$$

La covarianza entre estimadores de la función de densidad producto LISA para dos individuos distintos s_i y s_l, en dos distancias es:

$$\begin{aligned}&\sigma_{il}^{r_1 r_2}=Cov_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right),\widehat{\rho}_{\varepsilon}^{(l)}\left(r_2\right)\right]=\\ &= E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right)\cdot\widehat{\rho}_{\varepsilon}^{(l)}\left(r_2\right)\right]-E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right)\right]\cdot E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(l)}\left(r_2\right)\right]=(*)\end{aligned} \quad (4.25)$$

para dos observaciones s_i y s_l y distancias $s_1 > \varepsilon > 0$ y $s_2 > \varepsilon > 0$. Notar que para procesos de Poison homogéneos, $E_{!!}\left[N(A) - 2\right] = \lambda a$ y condicionando una observación a dos puntos del proceso, $\frac{(n-2)}{a}$ es un estimador insesgado para λ.

Por tanto, $\quad E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right)\right] = \lambda^2 + \frac{2\lambda}{a} + \frac{\lambda + \frac{1}{a}}{2\pi r_1} g\left(s_i, s_l, r_1\right)$

y una expresión similar se obtiene para $E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_2\right)\right]$.

Así,

$$\begin{aligned} E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right) \cdot \widehat{\rho}_{\varepsilon}^{(l)}\left(r_2\right)\right] &= \lambda^4 + 7\lambda^3/a + 9\lambda^2/a^2 + \frac{\lambda^3 + 3\lambda/a + 1/a^2}{\left(2\pi\right)^2 r_1 r_2} g\left(s_i, s_l, r_1\right) \\ & \quad g\left(s_l, s_i, r_2\right) + \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{2\pi} \\ & \quad \left[\frac{g\left(s_i, s_l, r_1\right)}{r_1} + \frac{g\left(s_l, s_i, r_2\right)}{r_2}\right] + \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{\left(2\pi\right)^2 r_1 r_2} \\ & \quad \int_A g\left(s_i, s, r_1\right) g\left(s_l, s, r_2\right) ds \end{aligned} \tag{4.26}$$

la covarianza será

$$\begin{aligned} (*) &= \frac{\frac{\lambda}{a}}{\left(2\pi\right)^2 r_1 r_2} g\left(s_i, s, r_1\right) g\left(s_l, s, r_2\right) + \frac{\frac{\lambda^2}{a}}{\pi}\left[\frac{g\left(s_i, s, r_1\right)}{r_1} + \frac{g\left(s_l, s, r_2\right)}{r_2}\right] + \\ &+ \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{\left(2\pi\right)^2 r_1 r_2} \int_A g\left(s_i, s, r_1\right) g\left(s_l, s, r_2\right) ds + 3\lambda^3/a + 5\lambda^2/a^2 \end{aligned} \tag{4.27}$$

La covarianza, para un elemento x_i, será:

$$\begin{aligned} \sigma_{ii}^{t_1 t_2} &= Cov_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right), \widehat{\rho}_{\varepsilon}^{(i)}\left(r_2\right)\right] = \\ &= E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right) \cdot \widehat{\rho}_{\varepsilon}^{(i)}\left(r_2\right)\right] - E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_1\right)\right] \cdot E_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}\left(r_2\right)\right] = \\ &= \frac{\frac{\lambda}{a}}{\left(2\pi\right)^2 r_1 r_2} g\left(s_i, s, r_1\right) g\left(s_i, s, r_2\right) + \frac{\frac{\lambda^2}{a}}{\pi}\left[\frac{g\left(s_i, s, r_1\right)}{r_1} + \frac{g\left(s_i, s, r_2\right)}{r_2}\right] \\ &+ \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{\left(2\pi\right)^2 r_1 r_2} \int_A g\left(s_i, s, r_1\right) g\left(s_i, s, r_2\right) ds + 3\lambda^3/a + 5\lambda^2/a^2 \end{aligned} \tag{4.28}$$

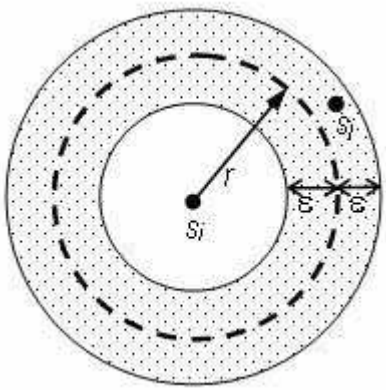

Figura 4.1: Esquema de las componentes asociadas al cálculo de las funciones LISA

Capítulo 5

Detección de rasgos en presencia de ruido. Distancias entre funciones LISA individuales

Indice General

5.1. Introducción 105

5.2. Estudio descriptivo de funciones individuales para un patrón puntual 107

5.3. Clasificaciones de rasgos en presencia de ruido. Ejemplos simulados 129

5.4. Buscando la cantidad de rasgo y ruido simulado . . . 148

5.5. Clasificaciones de rasgos en presencia de ruido. Imágenes reales 151

5.6. Análisis comparativo entre el algoritmo EM y el algoritmo SEM 154

5.1. Introducción

El objetivo de este capítulo es la detección de rasgos en imágenes cubiertas de ruidos. Para ello hacemos uso de la potencialidad de las funciones LISA introducidas en el capítulo anterior. Para cada patrón puntual, evaluamos la correspon-

diente función LISA, calculada sobre un vector de distancias. Supondremos, sin pérdida de generalidad, que la colección de puntos se da en el cuadrado unidad y por lo tanto el vector de distancias tomado es el que tiene por límites los valores 0.005 y 0.5, con un total de 100 elementos.

Conocemos el valor esperado de la función LISA bajo aleatoriedad (ruido en nuestro contexto), por tanto, el comportamiento y la forma de estas funciones puede identificar el tipo de patrón y efectivamente, distinguir entre ruido y rasgo. Recordemos que rasgo viene identificado por una colección de puntos agrupados en una pequeña región del espacio.

En primer lugar, se muestra la evidencia de un comportamiento diferente de las funciones LISA individuales de ruido y rasgo, mediante ejemplos simulados. Aleatoriamente distribuimos puntos de ruido en el cuadrado unidad y los puntos de rasgo los definimos también de forma aleatoria en pequeñas regiones con tres formas: cuadrado, círculo y forma L.

Parece lógico que las funciones LISA de puntos de ruido permanecerán cerca del valor esperado, mientras que las del rasgo tendrán forma distinta y presentarán grandes distancias frente al valor esperado.

La segunda parte del estudio de simulación aquí presentada, clasifica mediante el algoritmo SEM y dadas las distancias entre estas funciones LISA individuales, los rasgos frente al ruido.

Tras los ejemplos simulados presentamos algunos casos reales de dos campos totalmente diferentes de la ciencia, como pueden ser la medicina y la geografía. Nuestro procedimiento aporta solución al problema planteado en cada uno de ellos, intuyendo así las múltiples aplicaciones que puede tener. En medicina, aplicaremos el algoritmo a fotos digitalizadas de células, cedidas por el Hospital de St. Joan de Déu de Barcelona, en mayo de 2001. Realizamos un tratamiento previo con el programa MATROX, para obtener una imagen binaria de negros y blancos. Este programa de tratamiento de imágenes, junto con otros de la marca MATROX, ha sido comercializado en los últimos 25 años por la empresa MATROX, cuya sede principal se encuentra en Dorval en la isla de Montreal, teniendo oficinas de ventas en América del Norte, Europa y Asia, constituyendo una herramienta potente en campos como la digitalización de imágenes en medicina; encontraremos más información en Matrox, 1991 [69]. Los puntos negros

serán ahora el rasgo y el ruido. El blanco indicará ausencia de puntos. El objetivo que nos planteamos con este tipo de imágenes es terapéutico, proporcionando una herramienta más en la diagnosis.

En geografía, aplicaremos el algoritmo a localizaciones de epicentros de terremotos, concretamente de la costa de California y cuyos datos son de libre acceso, perteneciendo a Peter Guttorp, del Departamento de Estadística de la Universidad de Washington, Seattle, Washington. En este caso, el objetivo es proporcionar información válida al estudio sísmico de la forma de las placas tectónicas, concretamente de la falla de San Andrés que se encuentra en esta zona.

5.2. Estudio descriptivo de funciones individuales para un patrón puntual

En esta sección vamos a realizar un estudio de simulación para describir el comportamiento de las funciones LISA de un patrón puntual espacial. Para ello hemos tenido en cuenta varias variables: cantidad de puntos de ruido, de rasgo, forma de los rasgos y tipos de distancias utilizadas.

Respecto de la cantidad de puntos de ruido que se han distribuido en el cuadrado unidad, hemos elegido tres cantidades 300, 400 y 500 puntos. Las cantidades de puntos que se han distribuido en los rasgos han sido 200, 400 y 600 puntos. Las formas elegidas de los rasgos han sido tres, forma cuadrada, círculo y forma en L. Las distancias utilizadas entre las funciones LISA individuales han sido también tres L1, L2 y divergencia de Kullback.

Una última variable que se ha tenido en cuenta, ha sido la intensidad de puntos entre los rasgos, cuando de éstos había más de uno. Con diferente intensidad nos referimos a diferente número de puntos por unidad de área en un rasgo. El objetivo de utilizar esta variable es intentar aproximar, en la medida de lo posible, el estudio de simulación a la casuística que podemos encontrar en la realidad y es evidente que en una foto real, los rasgos que pueden aparecer son heterogéneos en intensidad de puntos, es decir, hay unos rasgos con mayor cantidad de puntos que otros.

El total de los casos trabajados, haciendo todas las combinaciones posibles entre las variables contempladas, ha sido 99, de los cuales comentaremos los más relevantes. La línea de presentación de los casos, es creciente en complejidad. En este sentido, comenzamos con ejemplos en los que sólo hay un rasgo, primero de forma cuadrada, luego forma L y por último forma circular. Estudiamos diferentes intensidades para cada uno de ellos. Posteriormente pasamos a dos rasgos, combinando las tres posibilidades de forma y de intensidad y por último finalizamos el estudio de simulación con los casos de tres rasgos.

Cada una de las imágenes que se van a mostrar en este apartado consta de nueve ventanas y comenzamos el estudio con la Figura (5.1).

Recorriendo de arriba a abajo y de izquierda a derecha, la primera de las imágenes de esta Figura, corresponde a la representación de los datos con los que estamos trabajando, que en este caso serían 500 puntos de ruido distribuidos aleatoriamente en el cuadrado unidad y 600 puntos de rasgo, que se encuentran localizados en el cuadrado de vértices (0.4,0.2),(0.6,0.2),(0.4,0.6),(0.6,0.6). Las siguientes dos ventanas, corresponden a estos mismos datos pero por separado, la primera es el ruido y la segunda es el rasgo. Cabe notar que al estar el ruido aleatoriamente distribuido, puntos de éste se encuentran situados en la región de rasgo y por tanto en la primera ventana es imposible distinguirlos.

La siguiente ventana muestra las representaciones de los vectores de funciones LISA individuales para cada punto, tanto rasgo (color verde) como ruido (color negro), además de la representación del valor esperado de la función LISA individual para un punto de ruido (color rojo). Formalmente tenemos los siguientes elementos:

Los vectores de funciones LISA individuales para cada punto (consideramos un total de n puntos) vienen dados por:

$$Y_i = \left[\widehat{\rho}_{\varepsilon}^{(i)}(r_1), \widehat{\rho}_{\varepsilon}^{(i)}(r_2), ..., \widehat{\rho}_{\varepsilon}^{(i)}(r_k)\right]^T; \quad i = 1, ..., n; \quad k \in N$$

y cada componente tiene la siguiente expresión

$$\widehat{\rho}_{\varepsilon}^{(i)}(r_t) = \frac{n-1}{2\pi r_t} \sum_{j \neq i} f_{\varepsilon}(\|s_i - s_j\| - r_t); r_t > \varepsilon > 0, \qquad t = 1, ..., k$$

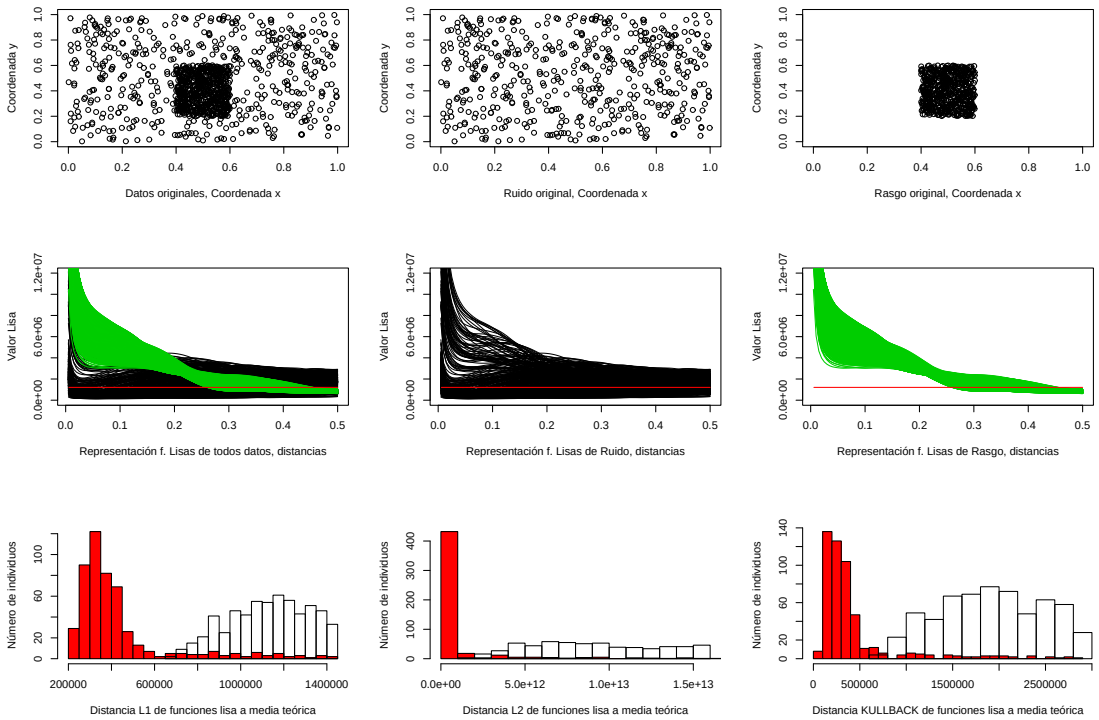

Figura 5.1: Simulación de rasgo cuadrado de 600 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

siendo f_ε el kernel de Epanechnikov

$$f_\varepsilon(s) = \begin{cases} \frac{3}{4}(1-s^2) & para \quad -1 \leq s \leq 1 \\ 0, & resto \end{cases}$$

El vector de distancias tomado es el que tiene por límites los valores 0.005 y 0.5, con un total de 100 elementos.

El valor esperado de la función LISA para un punto de ruido de un patrón de Poisson viene dado por:

$$E_! \left[\widehat{\rho}_\varepsilon^{(i)}(r)\right] = \lambda^2 + \frac{\lambda}{a}$$

siendo $\frac{n-1}{a}$ un estimador insesgado de λ, donde a representa el área, que en nuestro caso es 1, por encontrarnos en el cuadrado unidad.

Los vectores LISA correspondientes a los puntos de rasgo muestran una agrupación en la que la primera parte de los vectores se encuentran muy por encima de la media teórica. Esta parte corresponde a los radios que definen distancias cortas. Si nos fijamos en el comportamiento de los vectores, pero ahora correspondientes a los puntos del ruido, se observa que desde el principio, la representación de la función está alrededor de este valor esperado, es decir, alrededor de la línea roja. Este hecho se repetirá en todos los ejemplos que veremos y, por tanto, ya nos puede dar idea, que si el comportamiento es diferente, éste será un elemento que podremos utilizar para diferenciar rasgo de ruido en una imagen real.

Para calcular las distancias de cada función LISA al valor esperado de la media teórica hemos utilizado tres tipos de distancias diferentes: distancia L1, distancia L2 y divergencia de Kullback. Las expresiones implementadas son las siguientes:

Para L1 :

$$d(f_1, f_2) = |f_1 - f_2| = \int_{-\infty}^{+\infty} |f_1(x) - f_2(x)|\, dx \tag{5.1}$$

Para L2 :

$$d(f_1, f_2) = \int_{-\infty}^{+\infty} (f_1(x) - f_2(x))^2\, dx \tag{5.2}$$

Para Kullback (Basseville, 1989 [9]) se ha utilizado la divergencia simétrica siguiente:

$$d(f_1, f_2) = \int_{-\infty}^{+\infty} |f_1(x) - f_2(x)| \left|\log \frac{f_1(x)}{f_2(x)}\right| dx \tag{5.3}$$

En cada uno de los histogramas se muestran las frecuencias de las distancias de los vectores de funciones LISA de todos los puntos con respecto a la media teórica. En color rojo se han representado las columnas de los puntos de ruido, por tanto más cerca del cero, porque las distancias son menores y en color blanco, se tienen las columnas de los puntos del rasgo, más distanciadas de la media teórica y por tanto, más alejadas del cero.

Se observa una clara bimodalidad en el histograma, en las tres distancias utilizadas, que se corresponde con una bimodalidad en las distancias de los vectores LISA. Una moda es para las distancias correspondientes a puntos de ruido y otra para las distancias correspondientes a puntos de rasgo. También se observa, en las columnas de rasgo, que en la parte inferior, hay columnas de mucha menos frecuencia, de color rojo, que corresponderían a ruido. Este comportamiento obedece al hecho de distribuir aleatoriamente ruido en el cuadrado unidad y por tanto, inevitablemente, puntos de ruido son considerados como rasgo por pertenecer a la misma localización espacial. Estos histogramas vienen a confirmar la hipótesis de clasificación de rasgo y ruido, de una forma más cuantificable (ya se ha visto de forma gráfica el comportamiento diferente de los vectores de funciones LISA). Estas distancias, al agruparse las columnas de rasgo por una parte y las de ruido por otra, indican que pueden ser un buen elemento para la clasificación.

Se observa también, que tanto al utilizar la distancia L1 como la de Kullback, los histogramas están más suavizados con respecto a los de la distancia L2. Este hecho tiene relación con la expresión de cada una de estas distancias. Si se utiliza L2, al elevar al cuadrado, los valores de la distancia crecen rápidamente. Ésto permite apreciar con más claridad la bimodalidad en los casos L1 y Kullback mientras que en L2 es más difícil de observar.

Las variaciones de la imagen anterior, Figura (5.1) que consta de un rasgo cuadrado, han sido considerar rasgo de 200, 400 y 600 puntos en la misma localización espacial y ruido de 300, 400 y 500 puntos, en el cuadrado unidad. Combinando estas posibilidades tenemos 9 imágenes y alguna de estas posibilidades se muestran en las Figuras (5.2) y (5.3).

El siguiente paso que nos planteamos es constatar que la diferencia de forma en los clusters o rasgos, no influye en la forma particular de los vectores de funciones LISA y por tanto, no es necesario tener en cuenta particularidades al respecto.

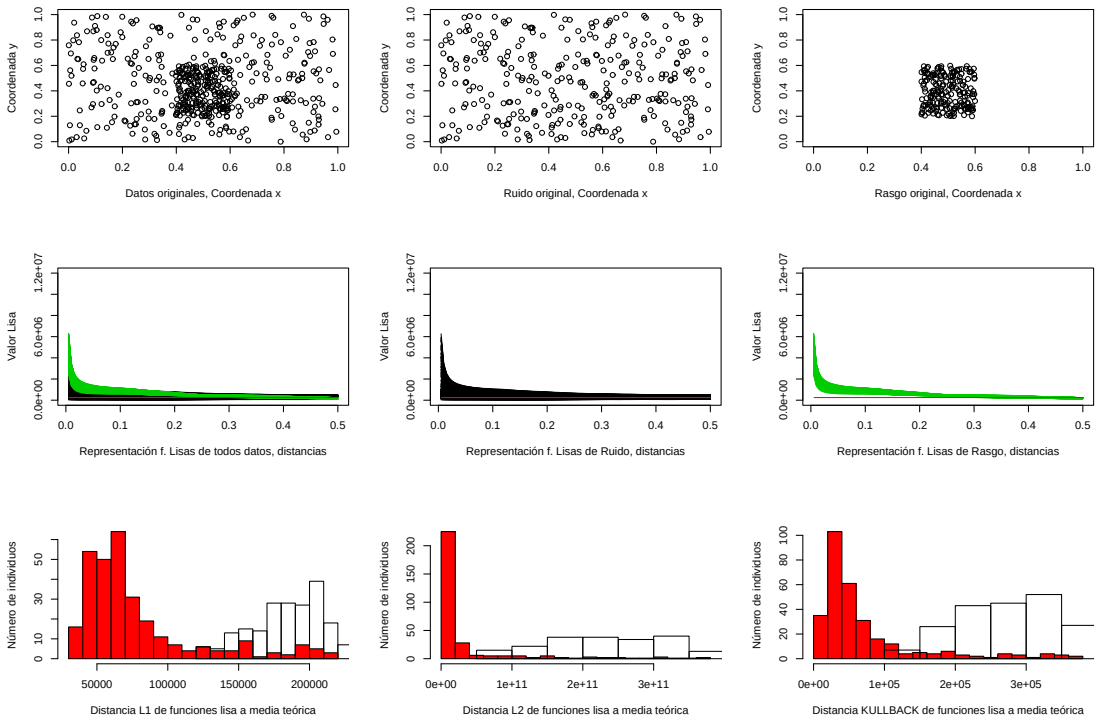

Figura 5.2: Simulación de rasgo cuadrado de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

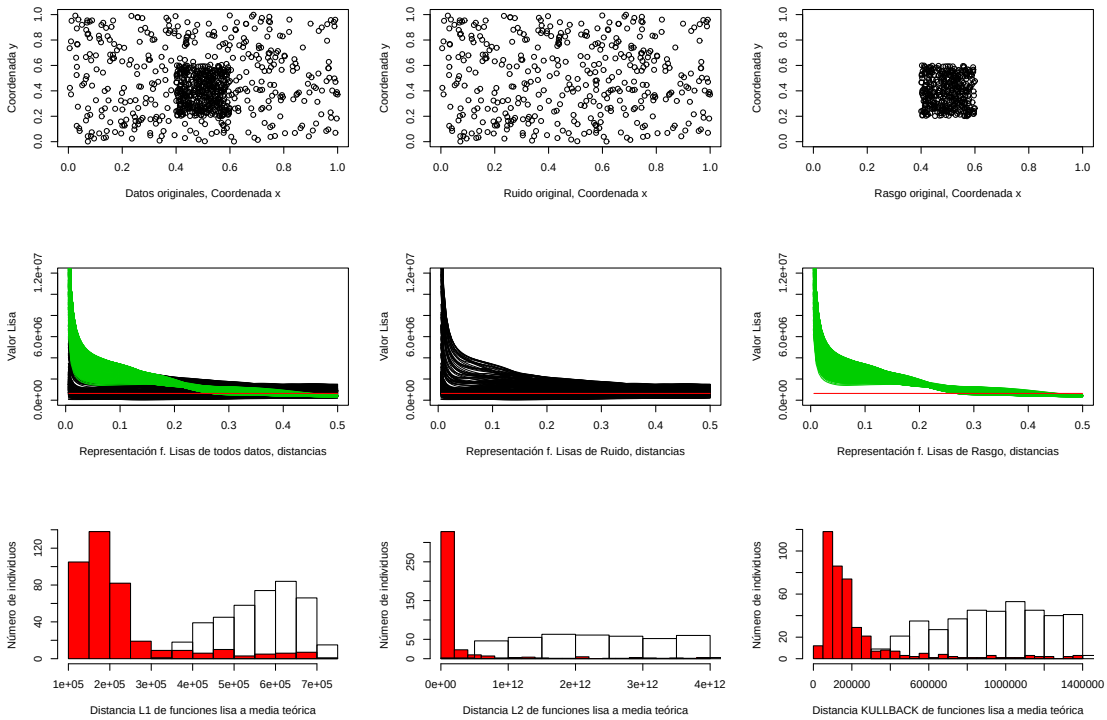

Figura 5.3: Simulación de rasgo cuadrado de 400 puntos sobre ruido de 400 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

De no ser así, una imagen con clusters irregulares, llevaría asociada una gran complicación para detectarlos.

Probamos, en primer lugar, con rasgos de forma no regular, concretamente forma de L, atendiendo a las posibilidades de rasgos que tengan recodos o esquinas. Con esta forma, se introduce la misma batería de posibilidades en cuanto al número de puntos en rasgo y en ruido, que en el caso del cuadrado. Para su análisis tomamos los casos que se muestran en las Figuras (5.4), (5.5) y (5.6).

Las tres primeras ventanas, al igual que en el caso del cuadrado, son los datos originales, el ruido sólo y el rasgo sólo. En las tres siguientes ventanas, las que corresponden a las representaciones gráficas de los vectores de las funciones LISA individuales para cada punto, se observa que el comportamiento es igual que en el caso del rasgo cuadrado. En distancias cortas las representaciones correspondientes a los puntos de rasgo, se mantienen por encima del valor esperado de la media teórica y las representaciones de los puntos del ruido, se agrupan alrededor de este valor esperado. Para acabar, las tres últimas ventanas corresponden a los histogramas para las tres distancias señaladas, en los que también se puede volver a observar la clara bimodalidad, tanto para rasgo como para ruido. Es evidente pues, que el comportamiento de los dos rasgos, es independiente de la forma.

Finalmente, consideramos una tercera forma de rasgo, la forma circular o redondeada, para disipar dudas respecto a la relación entre forma y comportamiento de los vectores de funciones LISA. Con el siguiente grupo de ejemplos consideramos que ya tenemos pruebas suficientes para afirmar y defender la independencia de estas dos características. Analicemos las Figuras (5.7), (5.8) y (5.9).

Estos ejemplos pertenecen a un grupo de 9 y en cada uno de ellos, la cantidad de puntos que aparece en el rasgo circular es distinta. La cantidad de puntos del rasgo con forma de círculo se aproxima distribuyendo puntos en el cuadrado unidad y rechazando aquellos que no se encuentran en la zona circular diseñada. Éste es el motivo de no controlar directamente la cantidad de puntos que quedarán en el rasgo, puesto que esta cantidad varía. Hay que jugar con una cantidad mayor de puntos que son los que sí que podemos predeterminar, de manera que cuando obtengamos la zona donde se encuentran esos puntos que forman el círculo, los que hayan quedado sean, aproximadamente, los que se deseaba tener. En cualquier

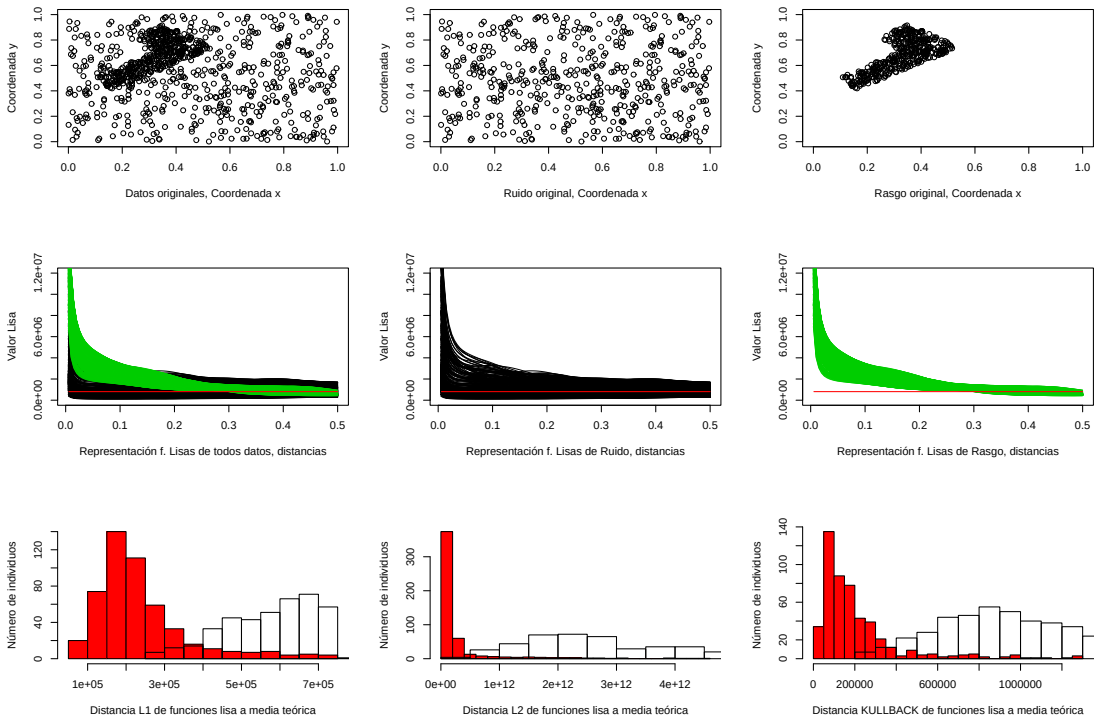

Figura 5.4: Simulación de rasgo forma L de 400 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

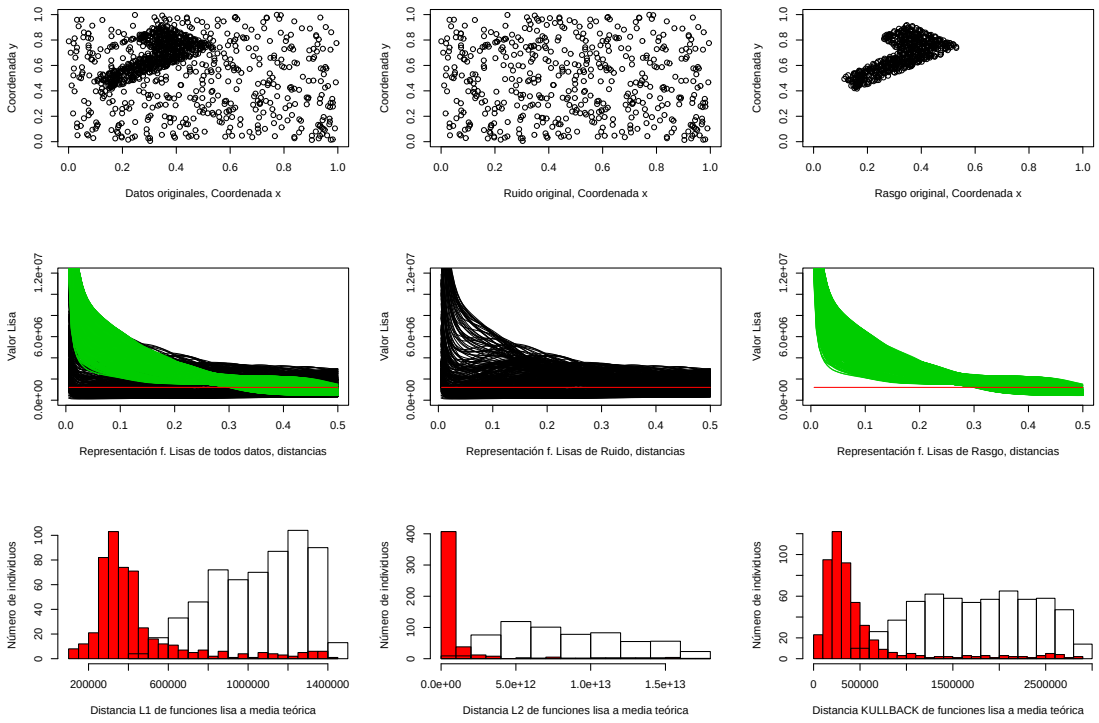

Figura 5.5: Simulación de rasgo forma L de 600 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

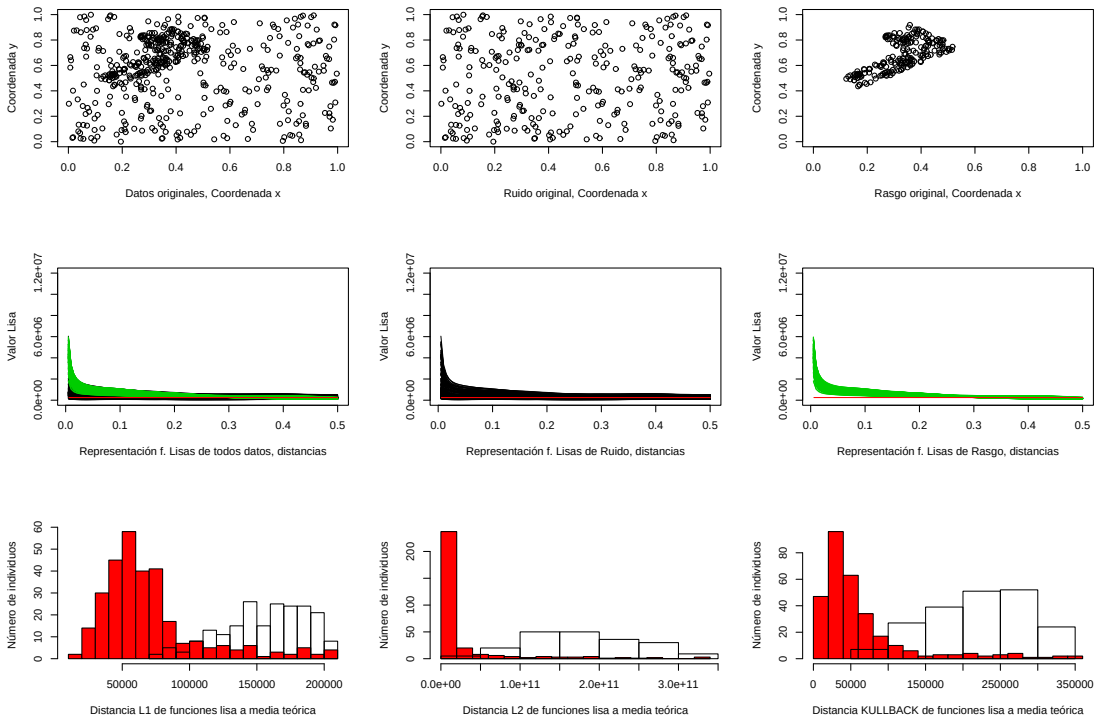

Figura 5.6: Simulación de rasgo forma L de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

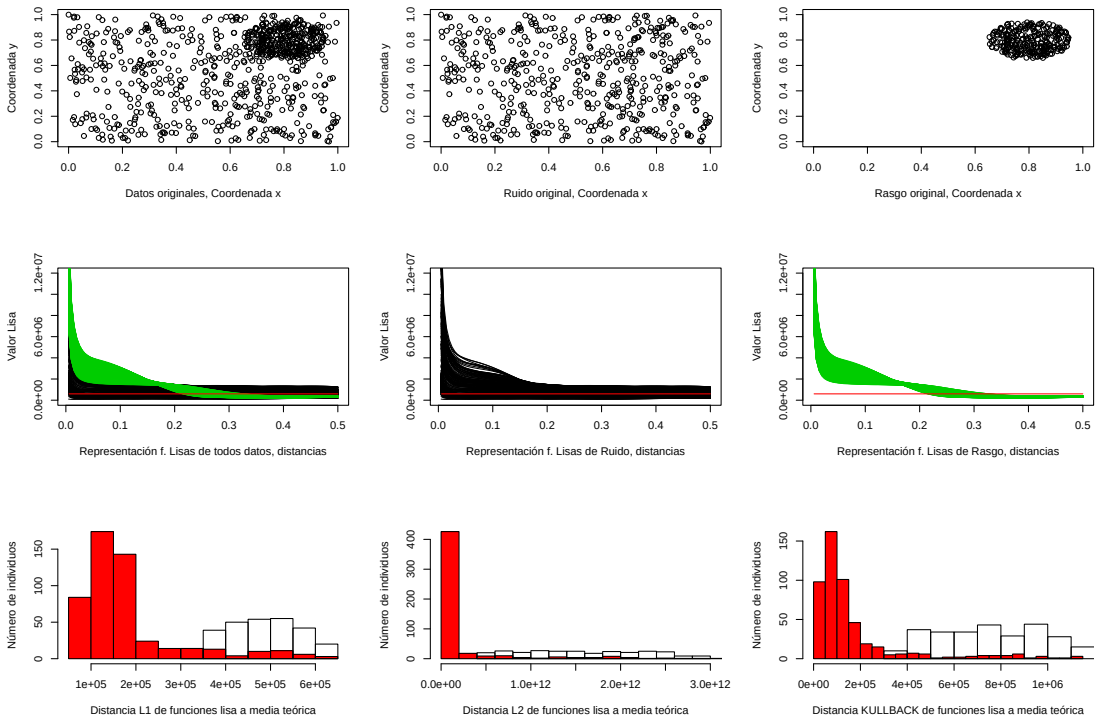

Figura 5.7: Simulación de rasgo con forma de círculo de 274 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

caso, las tres primeras ventanas corresponden a los datos simulados y las tres siguientes a las representaciones de los vectores de las funciones LISA individuales para cada punto. Observamos nuevamente la forma y constatamos que vuelve a ser análoga a los casos anteriores de rasgo cuadrado y forma L. Ocurre de manera similar con los histogramas.

Concluimos, pues, que en el caso de un sólo rasgo, la forma que presente, no afecta al comportamiento de los vectores LISA individuales para cada punto del patrón puntual. Además, constatamos que del conjunto de puntos, los que forman un rasgo o agrupación, se comportan de forma parecida y diferenciada de los puntos que forman parte del ruido. De la misma manera, los histogramas de las distancias de cada uno de los vectores de funciones LISA al valor esperado de una función LISA de un punto de ruido, muestran una clara bimodalidad, una de ellas para los puntos de rasgo y otra para los puntos de ruido.

Abordamos, pues, el caso en que haya más rasgos y pensamos casos con dos. Nos interesa saber si este hecho afecta a la forma o a las distancias entre estos vectores de funciones LISA. Incluso nos cuestionamos más, ¿un número diferente de puntos en cada uno de los rasgos, por tanto diferente intensidad de puntos, afectaría a los dos parámetros con los que estamos trabajando, es decir representación e histogramas?. Con los tres rasgos presentados en los ejemplos anteriores y utilizando todas las combinaciones de forma, cantidad de puntos de rasgo, cantidad de puntos de ruido, misma intensidad entre los dos rasgos y diferente intensidad entre ellos, obtenemos 54 casos. Por organización, precisión y comodidad a la hora de comentar y estudiar casos, los distribuimos en dos grupos. Aquellos que tienen la misma intensidad de puntos en los dos rasgos (27 casos) y los que la tienen diferente, que serían el resto (27 casos). Dentro de cada uno de ellos tenemos tres grupos, combinando cuadrado y círculo, cuadrado y forma L y por último, forma L y círculo y en cada uno de estos tres grupos, los 9 que han ido apareciendo en el caso de un sólo rasgo, es decir combinar 300, 400 y 500 puntos de ruido, con 200, 400 y 600 puntos de rasgo (aproximadamente a estas cifras en el caso del rasgo circular). De este primer grupo, mostramos tres ejemplos, las Figuras (5.10), (5.11) y (5.12), representativas de cada combinación de forma y de distintas cantidades de puntos en rasgo y ruido.

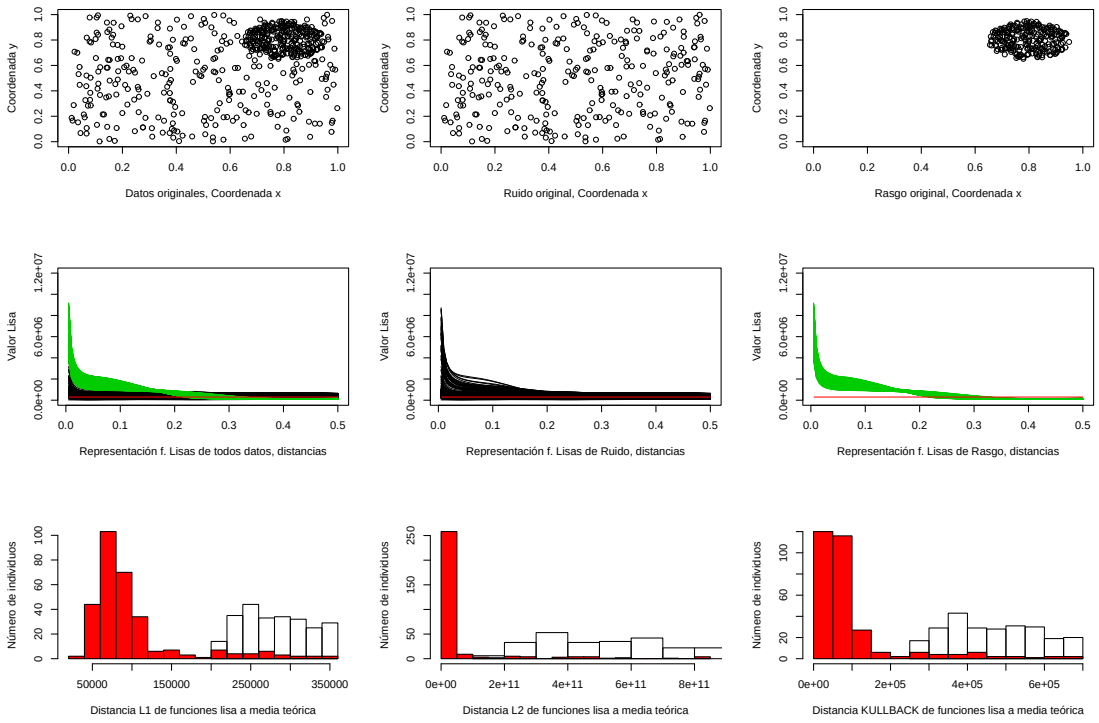

Figura 5.8: Simulación de rasgo con forma de círculo de 246 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

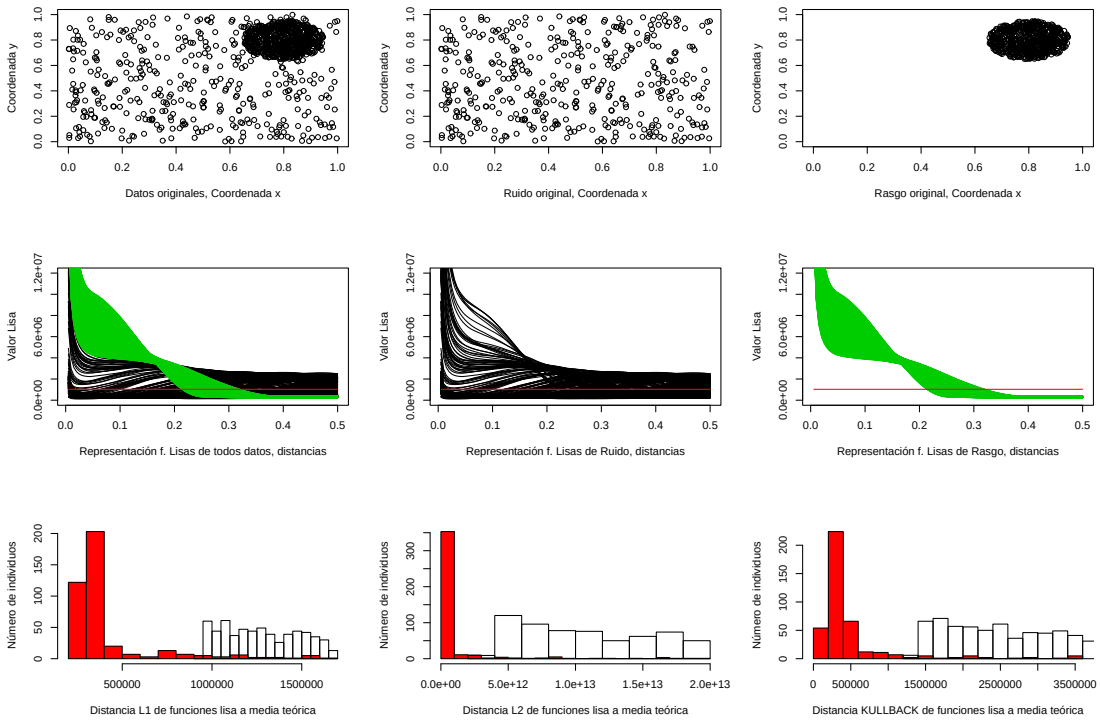

Figura 5.9: Simulación de rasgo con forma de círculo de 615 puntos sobre ruido de 400 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

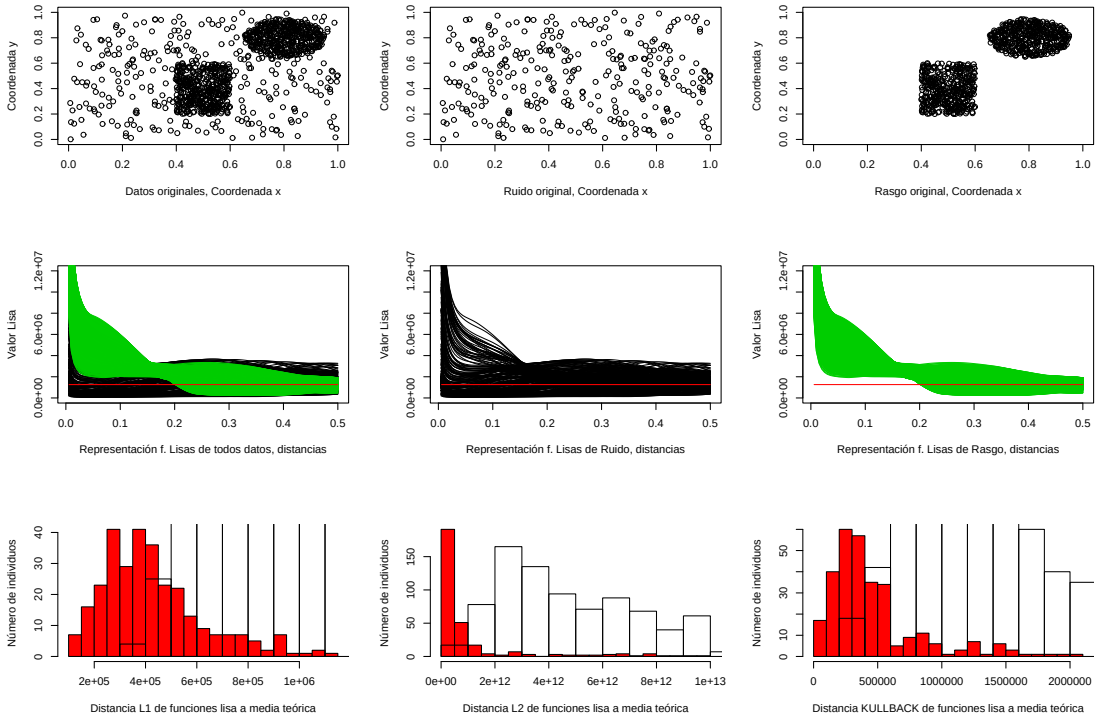

Figura 5.10: Simulación de dos rasgos, uno con forma cuadrada de 400 puntos y forma de círculo de 404 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

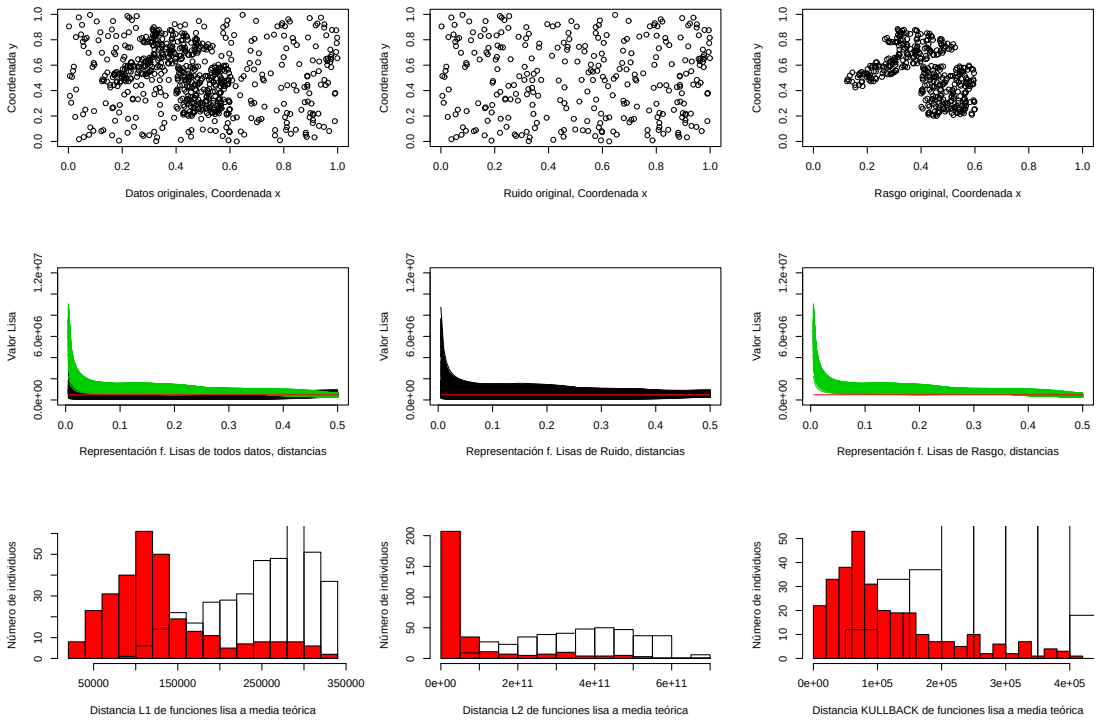

Figura 5.11: Simulación de dos rasgos, uno con forma cuadrada de 200 puntos y forma L de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

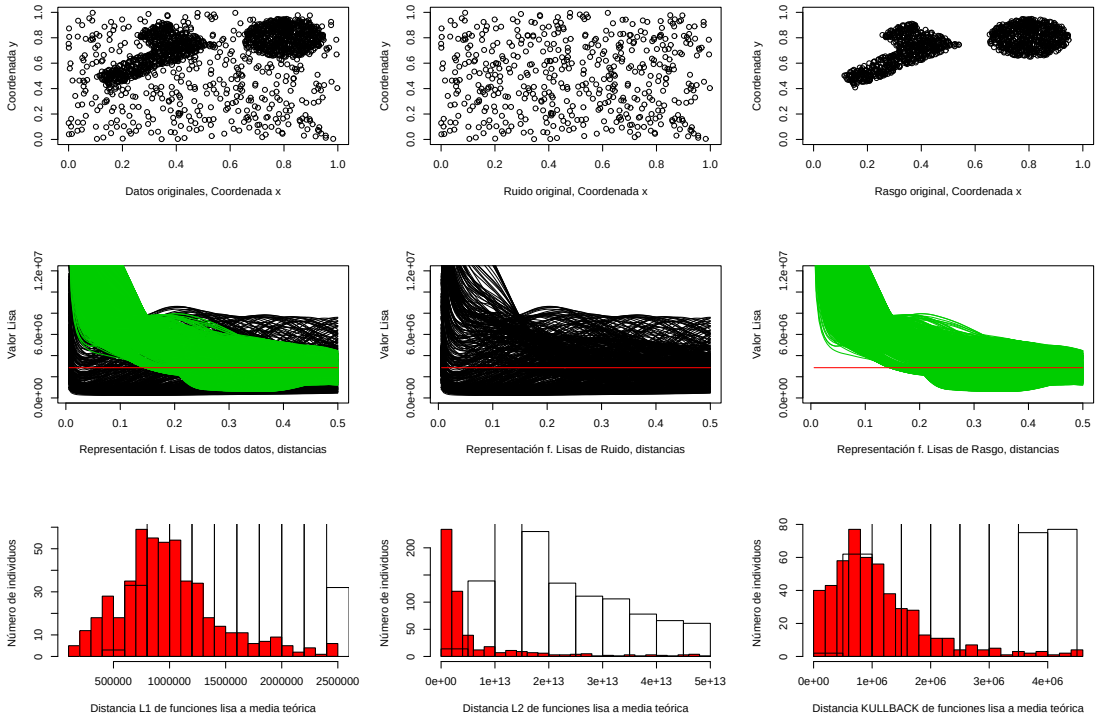

Figura 5.12: Simulación de dos rasgos, uno con forma L de 600 y forma de círculo de 591 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

En todos ellos se aprecia que el comportamiento de las funciones LISA es similar al caso de un rasgo. Estas funciones se mantienen por encima, a distancias cortas, del valor esperado y alrededor de este valor en el caso de puntos de ruido.

Cambiamos la intensidad de los rasgos, comprobando que no hay diferencias significativas entre los casos anteriormente analizados y aquellos en los que en uno de los dos rasgos del patrón puntual la intensidad del número de puntos es mayor y en otro menor. Éste sería el segundo de los grupos comentados anteriormente. Mostramos algunos de los resultados en las Figuras (5.13), (5.14).

De todos ellos, podemos destacar que cuando la intensidad de rasgo es menor, las representaciones de los vectores de funciones LISA individuales, se acercan más al valor esperado de la función LISA para un punto de ruido. Esta situación es fácil de analizar, puesto que se trata de una situación intermedia entre un rasgo, con una fuerte cercanía entre los puntos y una situación de ruido. En cualquier caso las evidencias mantenidas anteriormente se hacen patentes.

La siguiente y última situación que nos planteamos, es el caso de tres rasgos: cuadrado, forma L y círculo. Hay un total de 18 casos, distribuidos en dos grupos, el primero, donde los rasgos tienen la misma intensidad de puntos y el segundo donde ésta es diferente. Dentro de cada grupo y como es habitual, tomamos variaciones del número de puntos de ruido, es decir, 300, 400 y 500 puntos y variaciones del número de puntos en los rasgos de 200, 400 y 600, aproximadamente.

Para el caso de la misma intensidad de puntos, cuando ésta es baja o se acerca a la intensidad del ruido, la información se muestra confusa. Tomamos un ejemplo, en el caso de 300 puntos de ruido, 100 en el rasgo cuadrado, 200 en el rasgo en forma de L y 216 en círculo, como se muestra en la Figura (5.15), y comentamos este hecho.

Se puede apreciar en este gráfico, similar a los comentados en los anteriores ejemplos, que no hay diferencias significativas entre las representaciones de los vectores de funciones LISA de puntos de rasgo y ruido. Incluso los histogramas se solapan, las columnas se superponen y no podemos estudiarlas con claridad.

En una realización más detallada, donde se distingan las representaciones de los vectores de las funciones LISA por rasgos en gráficos diferentes, es decir, los

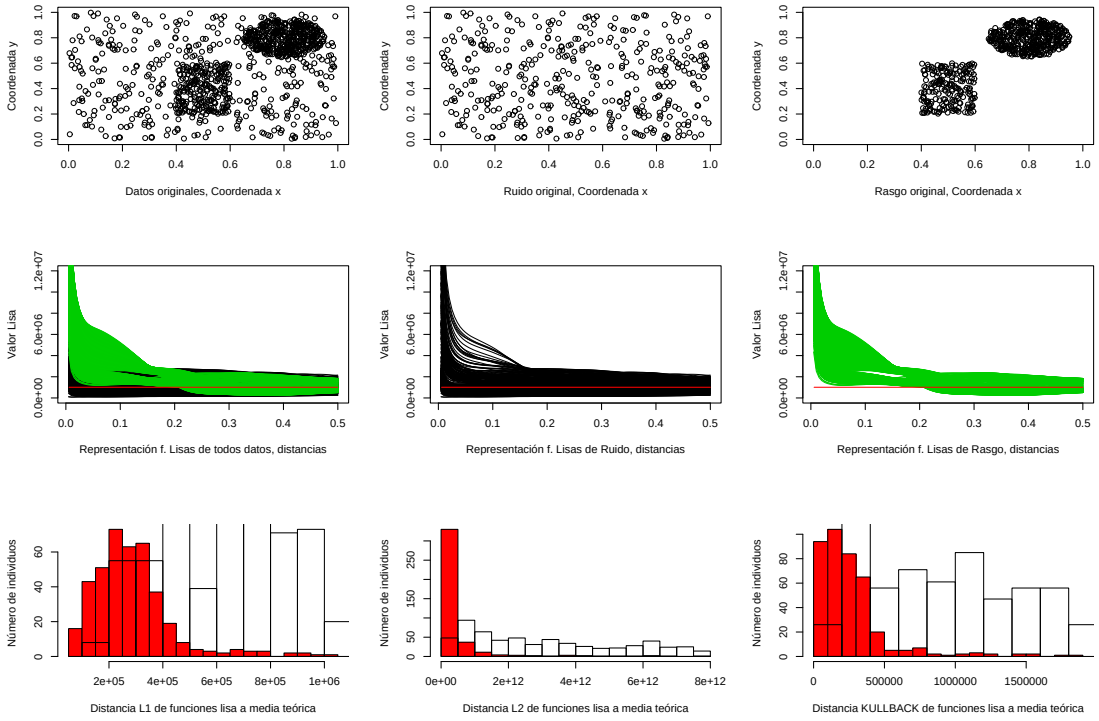

Figura 5.13: Simulación de dos rasgos, uno cuadrado de 200 puntos y forma de círculo de 505 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

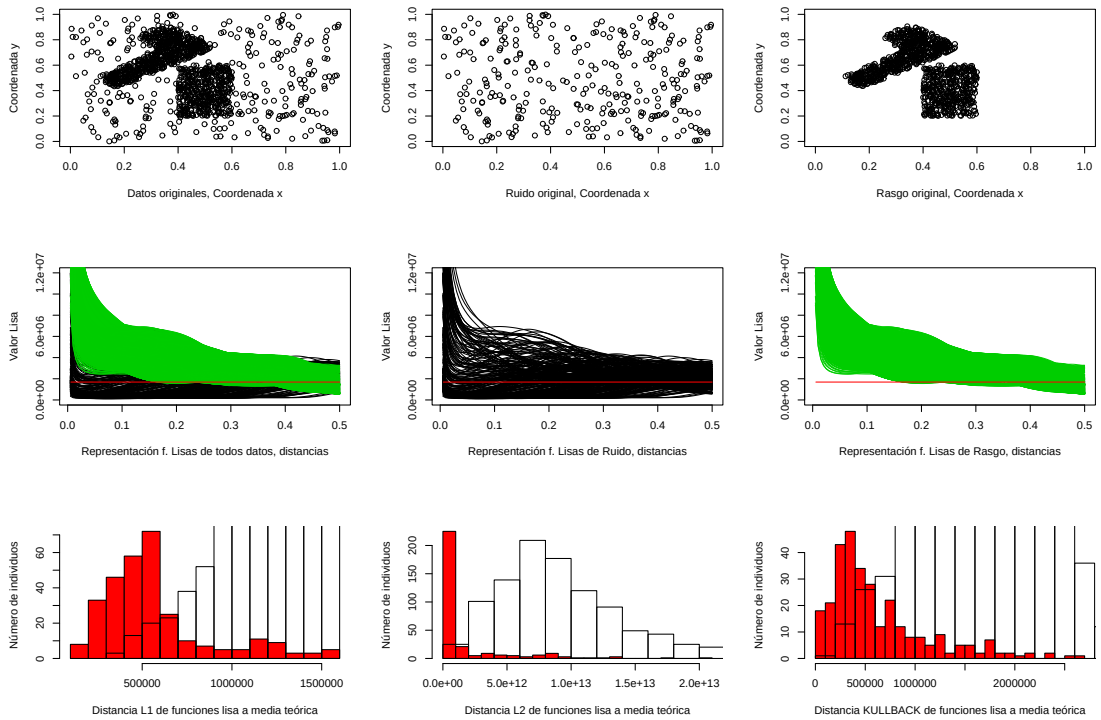

Figura 5.14: Simulación de dos rasgos, uno cuadrado de 400 puntos y forma de círculo de 600 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

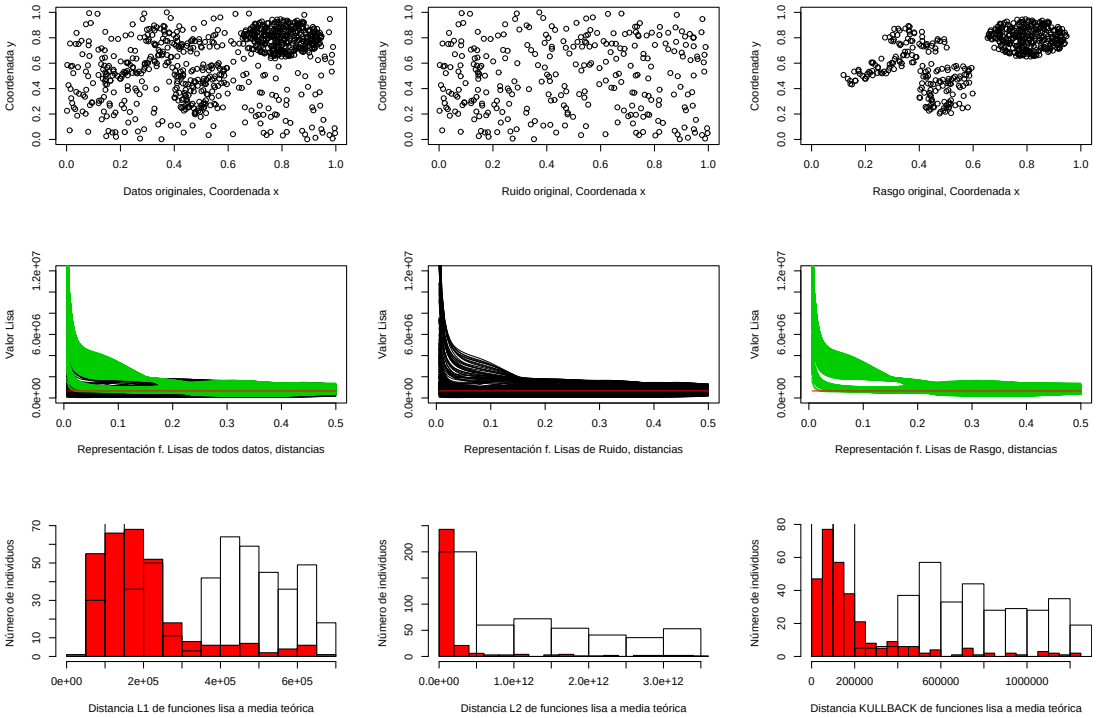

Figura 5.15: Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 200 puntos y forma de círculo de 216 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

del cuadrado, los del rasgo forma L y los del círculo, se aprecia que según la intensidad de éstos, las funciones tienen comportamientos más diferenciados y ciertamente, como cabía esperar, donde el rasgo tiene una intensidad de puntos más baja, como es el caso del cuadrado, las funciones presentan menor diferencia a las del ruido. Para el círculo, el comportamiento es claramente diferenciado, tal y como se muestra en la Figura (5.16).

Esta característica se acusa más cuando la diferencia de intensidades entre los rasgos es más clara, tal y como ocurre en el siguiente ejemplo, Figura (5.17).

En este caso, el cuadrado tiene 100 puntos, el rasgo en forma de L 400 y 575 puntos el círculo y se puede observar que además de lo que se ha comentado para el ejemplo anterior, la confusión entre rasgo y ruido es mayor. Dada la gran cantidad de vectores de funciones LISA, representamos las funciones por separado. Se muestra en la Figura (5.18).

En ella apreciamos más la diferencia de comportamiento entre las representaciones de los vectores de las funciones LISA individuales, correspondientes a los puntos del rasgo cuadrado (poca intensidad), de las de los puntos del rasgo circular (más intensidad) y, obviamente, este comportamiento se ha de traducir en la clasificación. La clasificación será más clara en aquellos rasgos donde la intensidad sea mayor o diferente de la del ruido, mientras que en aquellos rasgos con una intensidad más débil respecto de la intensidad del ruido, la clasificación será más confusa.

5.3. Clasificaciones de rasgos en presencia de ruido. Ejemplos simulados

En el apartado anterior hemos justificado que el uso de las funciones LISA es adecuado para la clasificación. Por ello, en este apartado y siguientes, nos proponemos clasificar y detectar rasgos mediante funciones LISA a través de un algoritmo SEM y mediante distancias al K-ésimo vecino más cercano basado en un algoritmo EM. La aplicación del algoritmo EM está vinculado a hipótesis de distribuciones de probabilidad tanto de las localizaciones de los puntos, como de

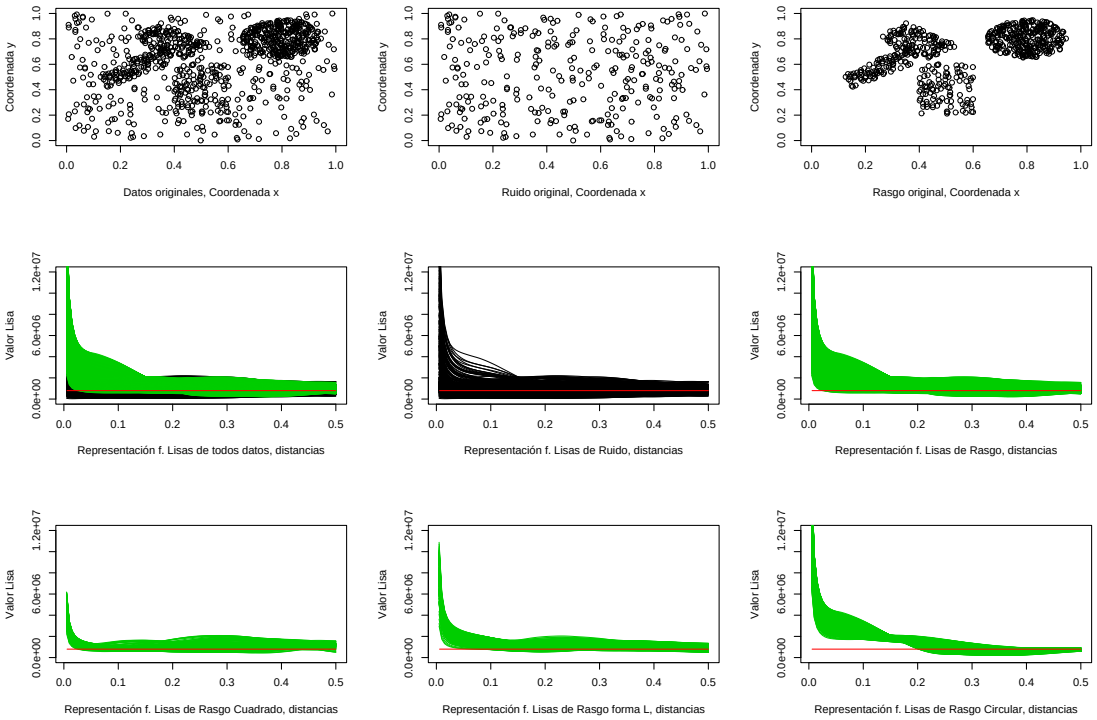

Figura 5.16: Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 200 puntos y forma de círculo de 293 puntos sobre ruido de 300 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Representaciones de funciones LISA individuales, (izquierda) rasgo cuadrado, (medio) rasgo forma L, (derecha) rasgo circular.

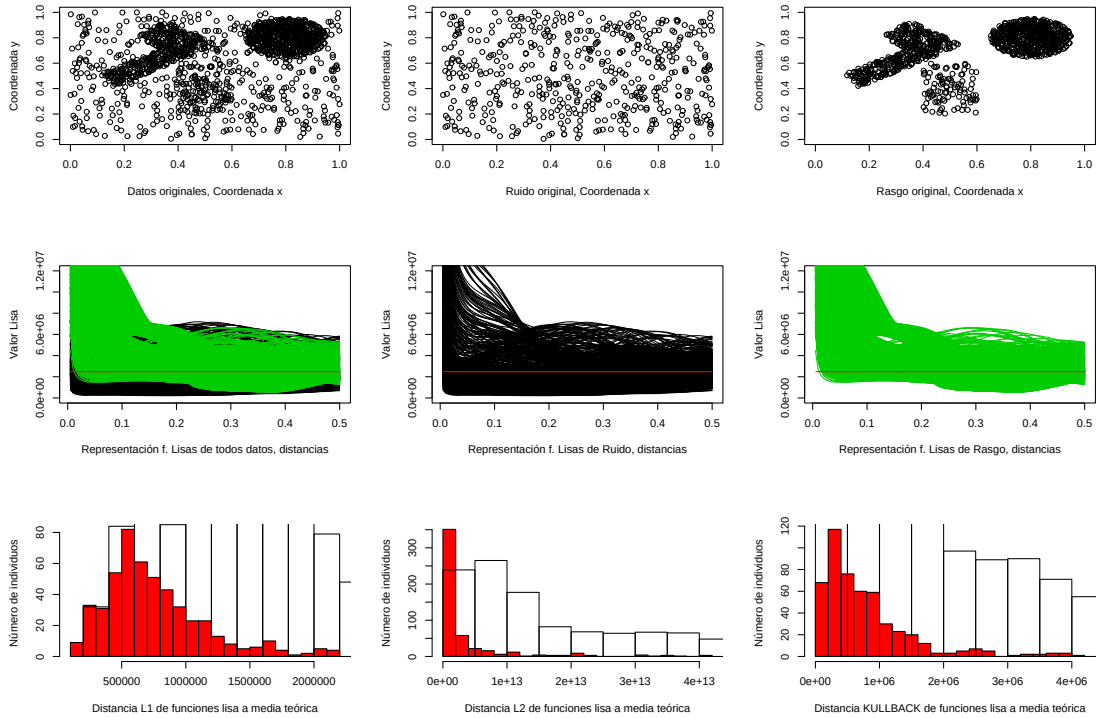

Figura 5.17: Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 400 puntos y forma de círculo de 575 puntos sobre ruido de 500 puntos en el cuadrado unidad. *Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Histogramas que muestran las frecuencias de las distancias de las funciones LISA individuales a la esperanza teórica, utilizando distancias L1, L2 y Kullback.

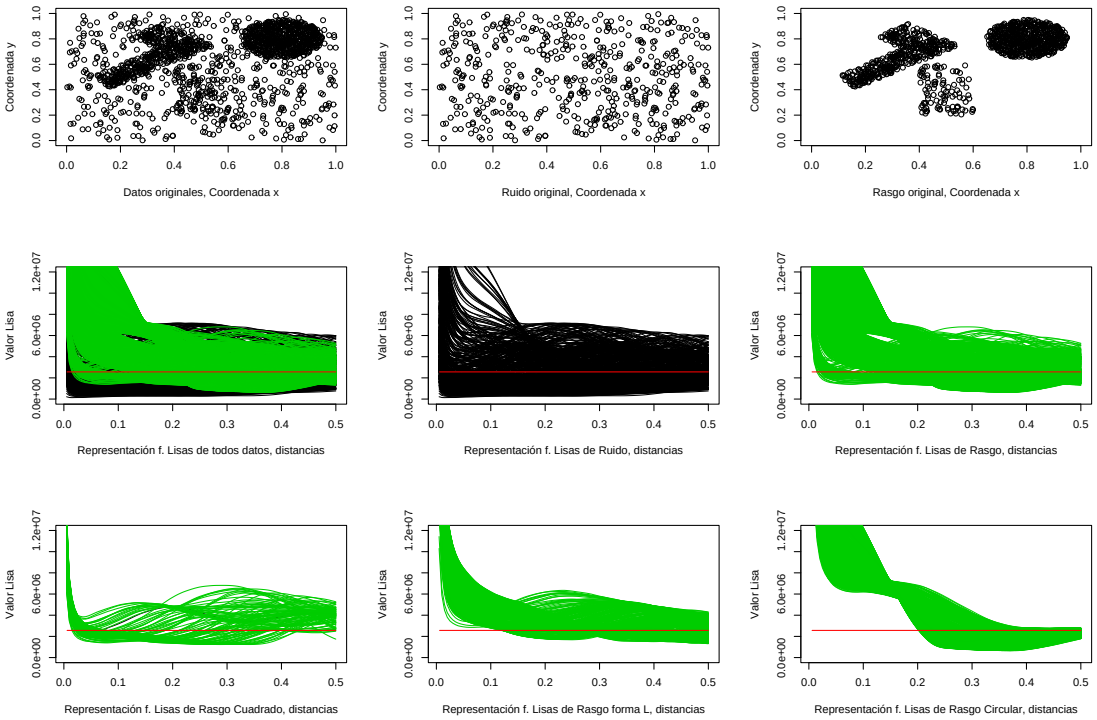

Figura 5.18: Simulación de tres rasgos, uno cuadrado de 100 puntos, uno forma L de 400 puntos y forma de círculo de 602 puntos sobre ruido de 500 puntos en el cuadrado unidad.*Primera fila:* Representación de los datos originales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Segunda fila:* Representaciones de funciones LISA individuales, (izquierda) ruido y rasgo conjuntamente, (medio) sólo ruido, (derecha) sólo rasgo. *Tercera fila:* Representaciones de funciones LISA individuales, (izquierda) rasgo cuadrado, (medio) rasgo forma L, (derecha) rasgo circular.

las distancias calculadas entre ellos. Básicamente, este algoritmo tiene dos pasos, en el primero, se obtiene la probabilidad de que un punto pertenezca a rasgo, mediante la siguiente expresión:

$$P(\lambda_1|d_i,\delta_i)=\frac{\widehat{p}\,(t)\,f_{D_k}(d_i;\widehat{\lambda}_1\,(t))}{\widehat{p}\,(t)\,f_{D_k}(d_i;\widehat{\lambda}_1\,(t))+(1-\widehat{p}\,(t))\,f_{D_k}(d_i;\widehat{\lambda}_2\,(t))},i=1,...,n$$

donde las distancias d_i tienen una distribución de probabilidad conocida, la Gamma generalizada, por tanto, una función de densidad también conocida con la que podemos operar. El valor $p(t)$ será la proporción de puntos de rasgo sobre el total (ruido y rasgo), $\lambda_1\,(t)$ y $\lambda_2\,(t)$ las intensidades de puntos en el rasgo y en el ruido, respectivamente. Notar que las localizaciones de los puntos siguen una distribución de probabilidad basada en un proceso de Poisson homogéneo restringido al cuadrado unidad en el caso de los puntos de ruido y a la forma del rasgo, en el caso de los puntos de rasgo. Por tanto, los parámetros en ambos casos son $\lambda_1\,(t)$ y $\lambda_2\,(t)$.

Será en el segundo paso del algoritmo, donde se obtendrán los estimadores $\widehat{p}\,(t)$, $\widehat{\lambda}_1\,(t)$, $\widehat{\lambda}_2\,(t)$.

En el caso de las funciones LISA, las distancias d_i son las distancias entre la función LISA de un individuo y la esperada bajo la hipótesis de aleatoriedad. Para el cálculo de estas distancias, igual que en el apartado anterior, se utilizan las distancias L1, L2 y divergencia simétrica de Kullback. Dada la definición de la función LISA, no tenemos una distribución de probabilidad de estas distancias, por tanto, no se puede aplicar el algoritmo EM, en las mismas circunstancias que en el Capítulo 3. Hay que modificarlo, para poder obtener la probabilidad anterior. La modificación que se ha introducido, es una adaptación del EM, que le denominamos Stochastic EM (SEM), donde la función de densidad de las distancias, evaluada en el punto correspondiente, se estima a partir de los histogramas de las distancias de las funciones LISA, correspondientes a puntos de rasgo y ruido simulados.

La situación es la siguiente. El algoritmo EM tiene dos pasos. En el primero precisa calcular una probabilidad y en este cálculo es necesario evaluar la función de densidad de las distancias. Esta función de densidad ha de proporcionar dos valores, según se esté evaluando respecto al rasgo o respecto al ruido. Para el caso

de trabajar con las distancias al K-vecino más próximo, que es el contexto donde se aplicaba este algoritmo (Banfield & Raftery, 1992 [7]), se disponía del modelo de probabilidad para estas distancias, por tanto se operaba sin ningún problema.

En nuestro contexto, donde las distancias se calculan entre funciones LISA individuales, no hay modelo de probabilidad para esta variable y por tanto, no hay función de densidad para utilizar en el cálculo de la probabilidad. Éste es el hecho que hace necesario un nuevo paso que solucione el problema, el Stochastic (S-step), que se introduce entre el E-step y el M-step y que nos proporcione esos dos valores. La forma de solucionarlo es estimar la función de densidad de estas distancias mediante su estimador usual, el histograma. El valor que debería obtenerse mediante la función de densidad, tanto para rasgo como para ruido, $f_{D_K}(d_i; \widehat{\lambda}_1(t))$ y $f_{D_K}(d_i; \widehat{\lambda}_2(t))$, queda sustituido y estimado por la frecuencia que los histogramas proporcionen, al situar en una de sus columnas, la distancia real del individuo.

Los histogramas creados en este paso, son dos, uno que recoge las frecuencias de las distancias de las funciones LISA individuales al valor esperado bajo aleatoriedad, de simulaciones de puntos de ruido y el correspondiente para simulaciones de puntos de rasgo. La cantidad de individuos simulados de rasgo y ruido es determinada previamente.

Con estos dos valores estimados ya podemos calcular en cada iteración del SEM, para el E-step, el valor de esta probabilidad, ya que el resto de la expresión es conocida. Esta probabilidad la redondearemos a 0 ó 1, dependiendo del valor al que se aproxime. Aquellos individuos a los que se les asigne una probabilidad de 1 entrarán a formar parte del rasgo en la iteración correspondiente. Análogamente, para el caso de tener un cero, la asignación es para el ruido. En el M-step, se estima el valor de $p(t)$ únicamente, ya que los valores de $\lambda_1(t)$ y $\lambda_2(t)$ no tienen sentido en nuestro contexto. Las iteraciones del algoritmo van en función de un criterio de convergencia.

La batería de ejemplos que hemos utilizado en este estudio de simulación para clasificar rasgos en presencia de ruido, tienen en cuenta las variables y los resultados del estudio descriptivo (cantidad de puntos de ruido, de rasgo, forma de los rasgos, tipos de distancias utilizadas, intensidad entre los puntos de los rasgos) y además añade otra variable, que es la cantidad de rasgo y de ruido simulado para el SEM. Ésta es muy importante, ya que la buena o mala clasificación que

proporcione el método dependerá de las cantidades de puntos, tanto en rasgo como en ruido, que utilicemos en las simulaciones.

La cantidad de puntos de ruido que se han distribuido en el cuadrado unidad es de 300, 400 y 500 puntos. Las cantidades de puntos que se han distribuido en los rasgos han sido 200, 400 y 600 puntos. Las formas de los rasgos han sido tres, forma cuadrada, circular y forma en L. Las distancias utilizadas entre las funciones LISA individuales han sido también tres L1, L2 y divergencia de Kullback. Se ha tenido en cuenta la intensidad de puntos entre los rasgos y hay ejemplos con la misma intensidad y otros con diferente intensidad. Todos los ejemplos anteriores, se han repetido tres veces, variando el rasgo simulado (80, 50 y 30 puntos) en un rasgo cuadrado de lado 0.1 y unidos respectivamente al ruido simulado (100, 700 y 500 puntos) en el cuadrado unidad. La variabilidad que resulta de todas las combinaciones es grande. Simplificando posibilidades, para no multiplicar ejemplos innecesarios, el total de casos analizados ha sido de 135.

En todos ellos, según la cantidad de rasgo y de ruido simulado que se ha utilizado, los resultados varían. Se han elegido unas cantidades que funcionan bien y otras que funcionan peor, para mostrar esta variabilidad. Las cantidades anteriormente mencionadas, se han tomado después de haber pasado una fase de pruebas.

Igual que en la sección anterior, la complejidad será creciente, analizando el caso de uno, dos y tres rasgos.

En el primer ejemplo, Figura (5.19), los parámetros se han fijado a 300 puntos de ruido y en él, un cuadrado de 200 puntos. El ruido que se ha simulado para el SEM es de 500 puntos y el rasgo simulado de 30 puntos (notar que las dimensiones del rasgo original y las del rasgo simulado son diferentes, aunque la intensidad de puntos es parecida; ésto se aprecia en la imagen de la Figura a la que se alude en este párrafo). El cálculo de las distancias se ha realizado con L1.

En la primera ventana aparece el rasgo original, sin ruido. A la derecha, el rasgo clasificado. Se han clasificado 265 puntos del total, cuando realmente había 200 puntos. Además de los que se ven en la imagen claramente, que están alrededor y que no son rasgo, hay que tener en cuenta que todos los puntos del ruido que hayan caido aleatoriamente en el recinto del rasgo, los va a clasificar como rasgo.

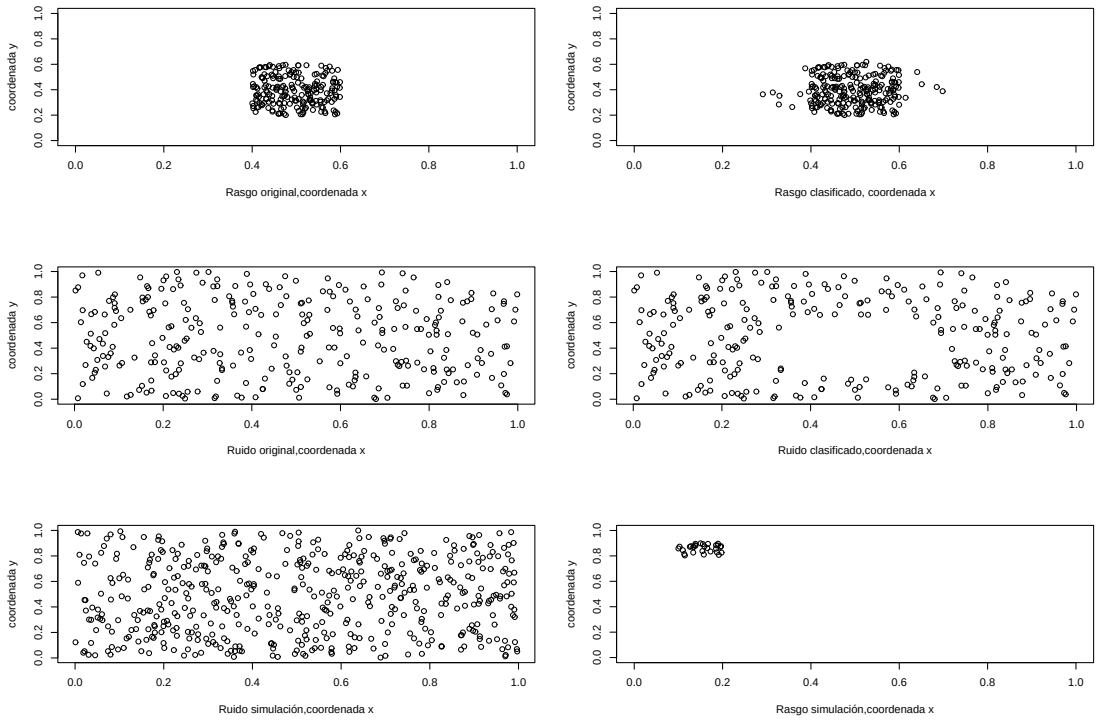

Figura 5.19: Clasificación de un rasgo cuadrado de 200 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 30 puntos en el rasgo simulado y 500 puntos en el ruido simulado. La distancia utilizada es L1. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

Este fenómeno se explica de una manera más detallada en las imágenes centrales. En ellas se puede observar que el ruido clasificado tiene una zona sin puntos, que son los que se han incluido en el rasgo. En las últimas imágenes, hemos incluido el ruido y el rasgo simulado que se ha utilizado para la parte estocástica del algoritmo EM.

Con las cantidades de ruido y rasgo simulado elegidas, la clasificación de puntos de rasgo original ha sido al 100 % satisfactoria, es decir, todos los puntos que originalmente son rasgo, se clasifican como rasgo y un 7 % de puntos de ruido son clasificados como rasgo, por tanto mal clasificados. Pero en última instancia, lo que nos interesa es clasificar rasgo, no clasificar ruido y el objetivo está cumplido.

Este resultado, que consideramos bueno, se repite, al utilizar otras distancias, L2 y Kullback, se muestra un ejemplo en la Figura (5.20). En este caso, utilizamos la divergencia de Kullback y un rasgo original, en forma de L. Las cantidades utilizadas son: ruido original de 300 puntos, rasgo original de 600 puntos, ruido simulado de 1000 puntos y rasgo simulado de 80 puntos. El porcentaje de buena clasificación de puntos de rasgo es de un 96 %, ya que 24 de los 600 puntos del rasgo, no se clasifican, además de algún punto de ruido que se clasifica como rasgo. Esta cifra así como la imagen mostrada, la consideramos bastante satisfactoria.

Un ejemplo de clasificación engañosa, aún teniendo el 100 % de rasgo clasificado, sería el ejemplo que se muestra en la Figura (5.21). En él podemos observar como una cantidad grande de ruido queda clasificado como rasgo y por tanto, el resultado no es bueno. El motivo es no utilizar cantidades de rasgo y ruido simulado, adecuadas.

Por último podemos observar el caso de nula detección, es decir, todo clasificado como ruido. Se muestra en la Figura (5.22).

Al aumentar el ruido original, detectamos que hay más casos, donde la clasificación es peor, pero, eligiendo convenientemente las cantidades de rasgo y de ruido simulados, tenemos buenos resultados, como el que mostramos en la Figura (5.23), con 400 puntos de ruido original, 400 de rasgo cuadrado, un ruido simulado de 700 puntos y un rasgo simulado de 50 puntos. El porcentaje de buena clasificación es del 99.5 %, utilizando la distancia L2.

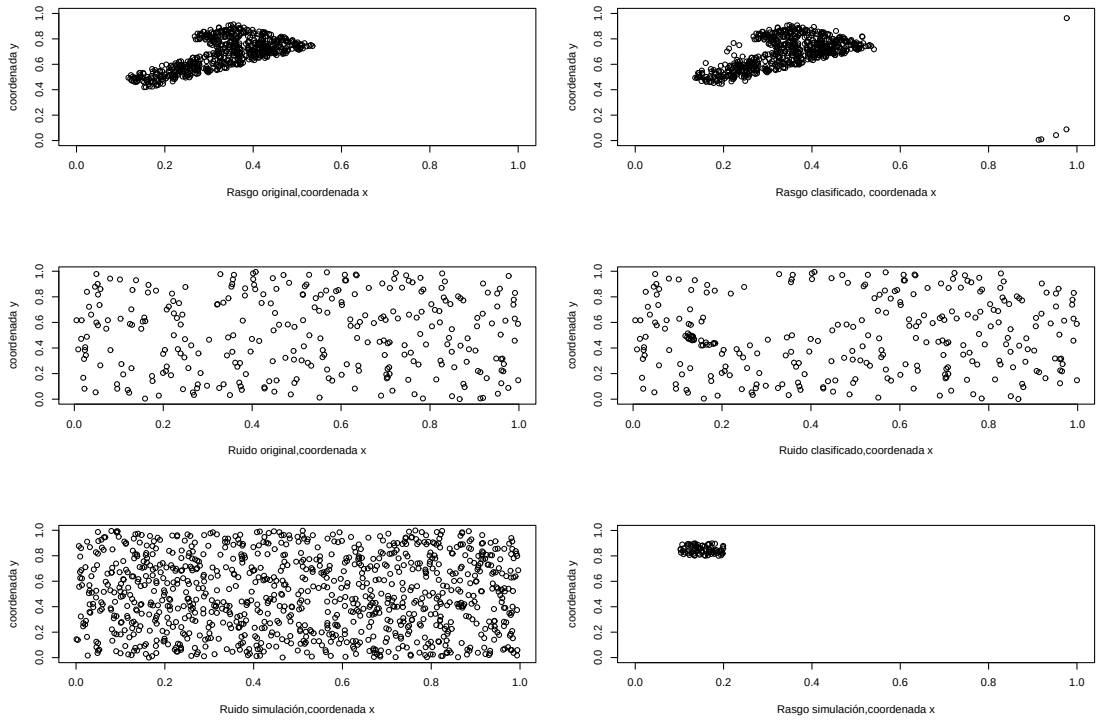

Figura 5.20: Clasificación de un rasgo forma L de 600 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es Kullback. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

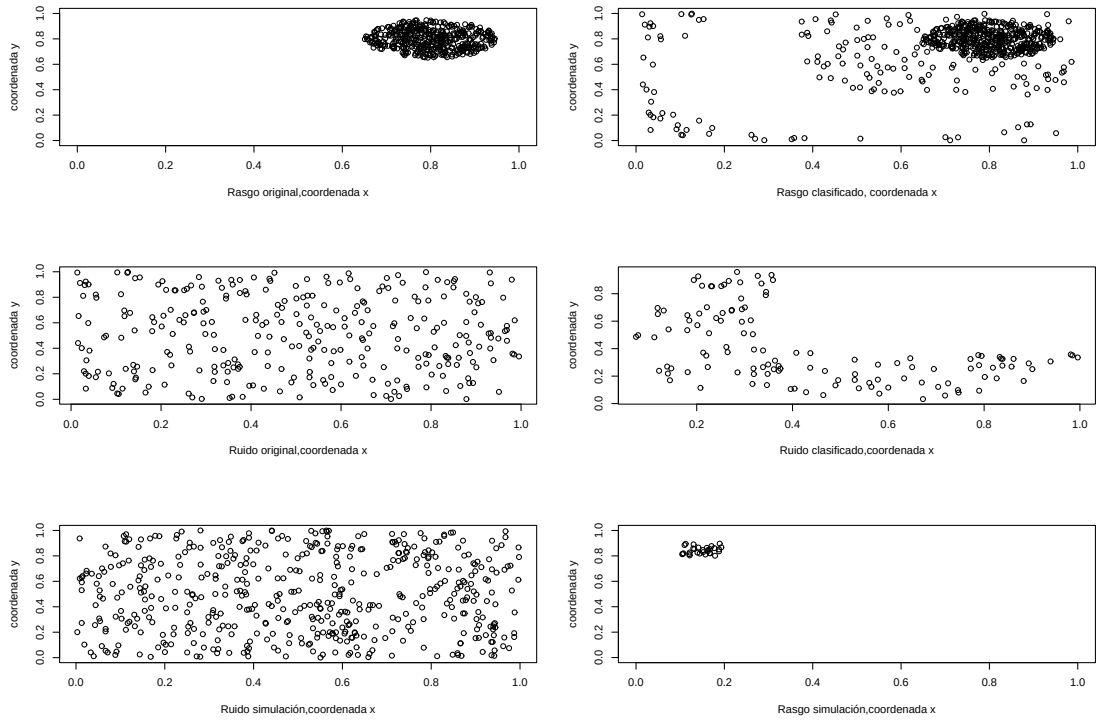

Figura 5.21: Clasificación de un rasgo forma circular de 417 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 30 puntos en el rasgo simulado y 500 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

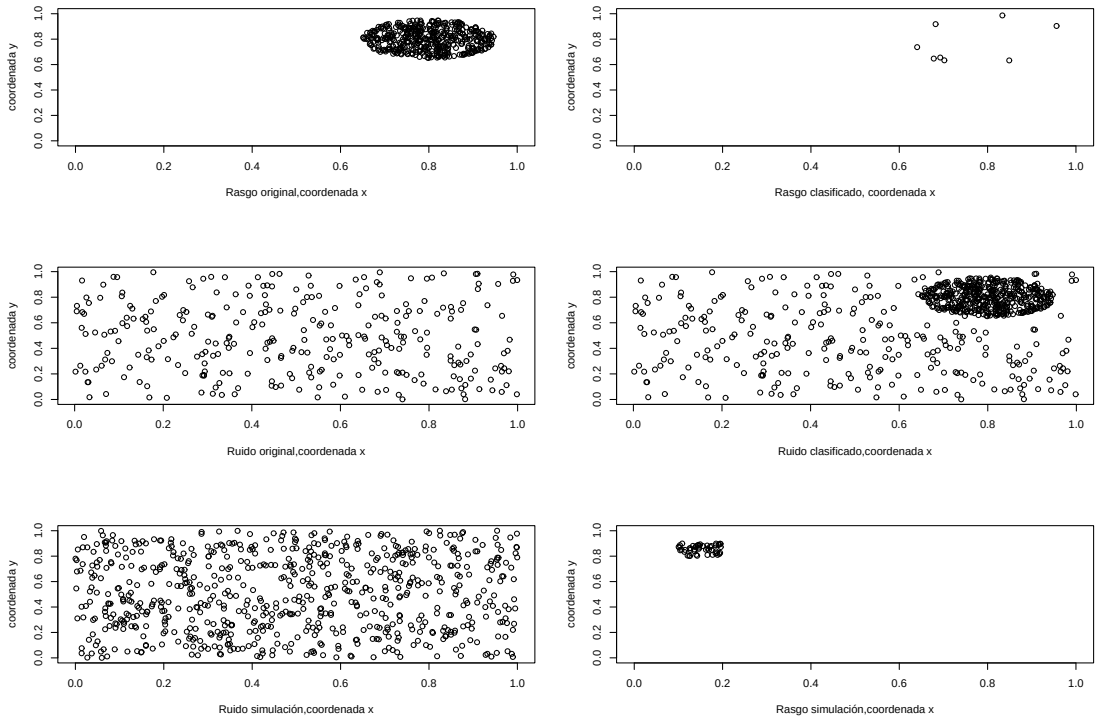

Figura 5.22: Clasificación de un rasgo forma circular de 432 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

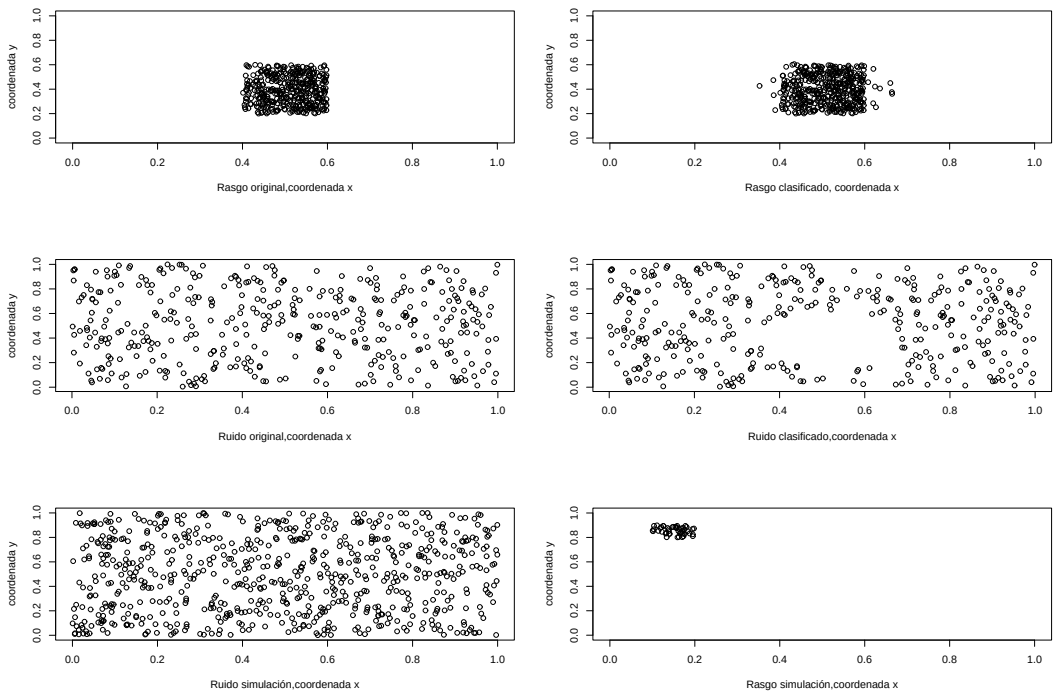

Figura 5.23: Clasificación de un rasgo cuadrado de 400 puntos sobre ruido de 400 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

En el ejemplo de la Figura (5.24), con una clasificación del 97.66 %, se ha clasificado un rasgo en forma de L de 400 puntos, en presencia de un ruido de 500 puntos, con un ruido simulado de 1000 puntos y un rasgo simulado de 80 puntos. La distancia utilizada en este caso ha sido, también, L2.

Pasamos a clasificar dos rasgos. En primer lugar utilizaremos la misma intensidad entre los dos rasgos. En el ejemplo de la Figura (5.25), tenemos un caso de 500 puntos de ruido, 200 de rasgo en forma cuadrada y 200 puntos en forma L. El ruido simulado es de 700 puntos y el rasgo simulado es de 50 puntos. Se puede observar que no es una clasificación perfecta, pero aún así, el 88 % del rasgo ha sido clasificado satisfactoriamente en tan sólo 20 iteraciones del SEM.

Cuando las cantidades de rasgo son superiores, la clasificación mejora, como apuntaba en su momento el estudio descriptivo. En el siguiente ejemplo, Figura (5.26), el rasgo cuadrado es de 400 puntos, el rasgo forma L de 600 puntos y el ruido es de 300 puntos. El ruido simulado es de 1000 puntos y el rasgo simulado es de 80 puntos. La distancia utilizada es L2.

Si la intensidad de los rasgos es diferente ocurre, como es esperable, que el rasgo donde la intensidad es mayor, queda mejor clasificado, que aquel donde puede ser una intensidad parecida a la del ruido. Para los casos de tres rasgos que hemos realizado, 54 casos en total, con forma cuadrada, forma L y forma redondeada, mostramos algún ejemplo, Figura (5.27), que ilustra este tipo de clasificación, teniendo en cuenta, que aunque parece una clasificación no muy buena, el total de puntos clasificados es del 85.64 %.

Para este caso, que cuenta con 300 puntos de ruido, 100 el rasgo cuadrado, 416 el rasgo circular y 250 el rasgo forma L, utilizando un ruido simulado de 700 puntos y un rasgo simulado de 50 puntos, se observa que el rasgo cuadrado, prácticamente no se clasifica. Al aumentar el ruido simulado, a 1000 puntos, la clasificación se restringe prácticamente al rasgo con forma circular, que es el que mayor intensidad tiene. Esto se repite en varios ejemplos más. El ejemplo de la Figura (5.28) que pasamos a analizar tiene 300 puntos de ruido, 100 puntos de rasgo con forma cuadrada, 428 de rasgo con forma circular y 250 puntos en rasgo con forma L. El ruido simulado es de 700 puntos con un rasgo simulado de 50 puntos.

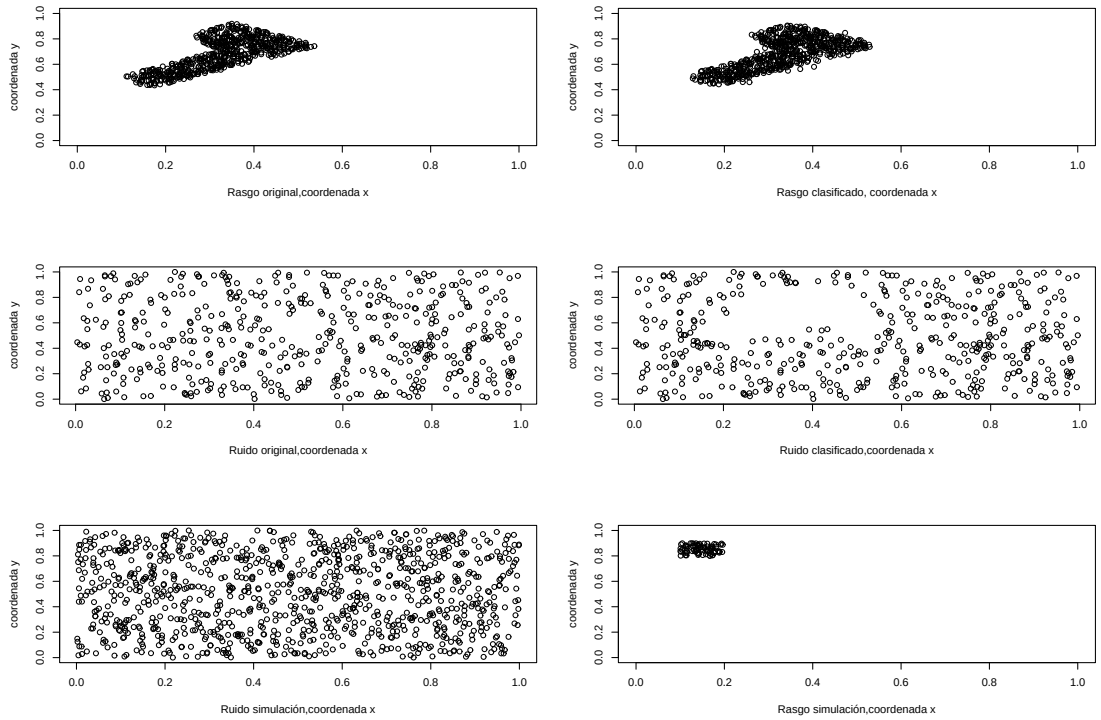

Figura 5.24: Clasificación de un rasgo forma L de 400 puntos sobre ruido de 500 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

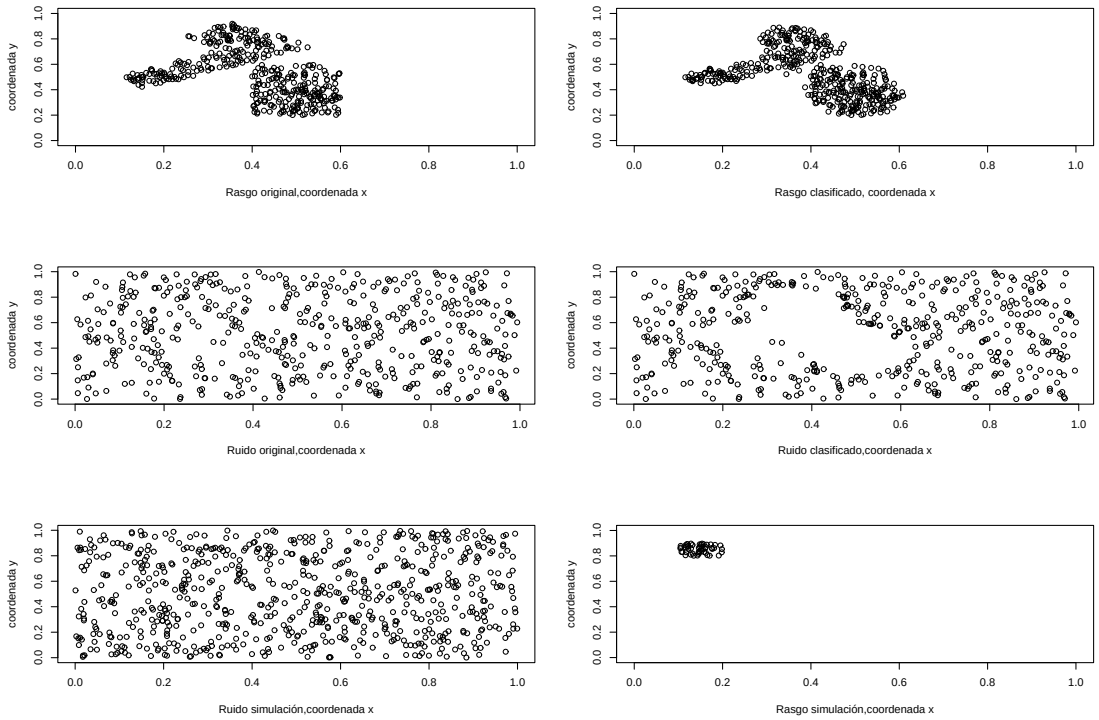

Figura 5.25: Clasificación de dos rasgos. Uno cuadrado de 200 puntos y otro forma L de 200 puntos sobre ruido de 500 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

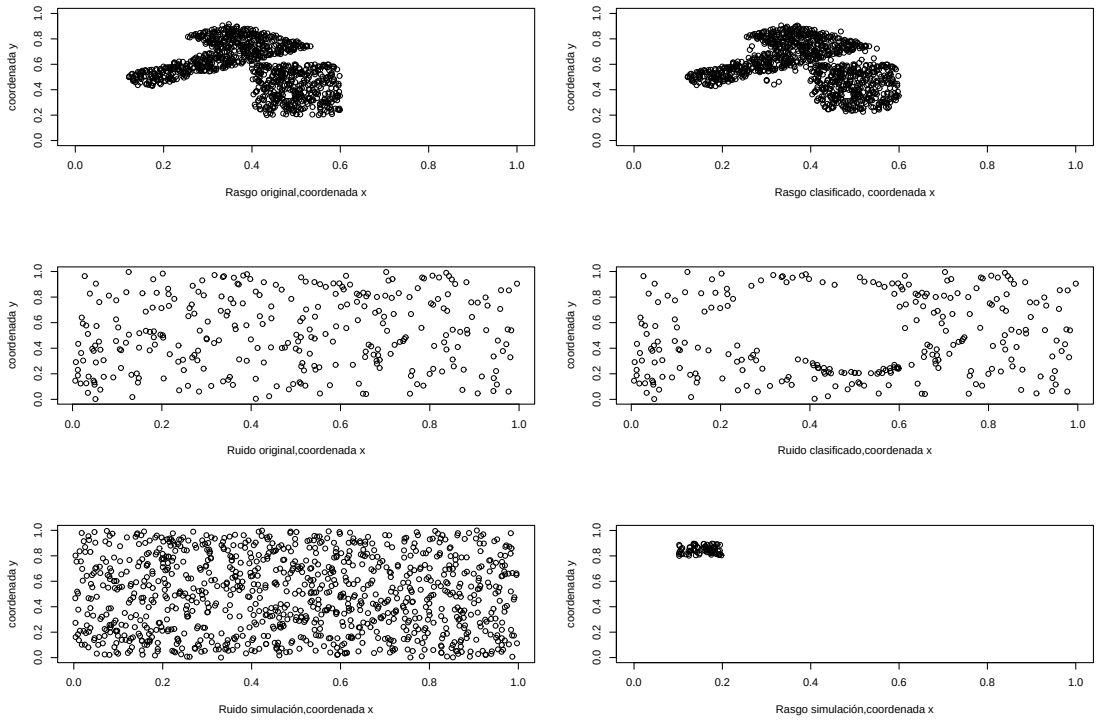

Figura 5.26: Clasificación de dos rasgos. Uno cuadrado de 400 puntos y otro forma L de 600 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

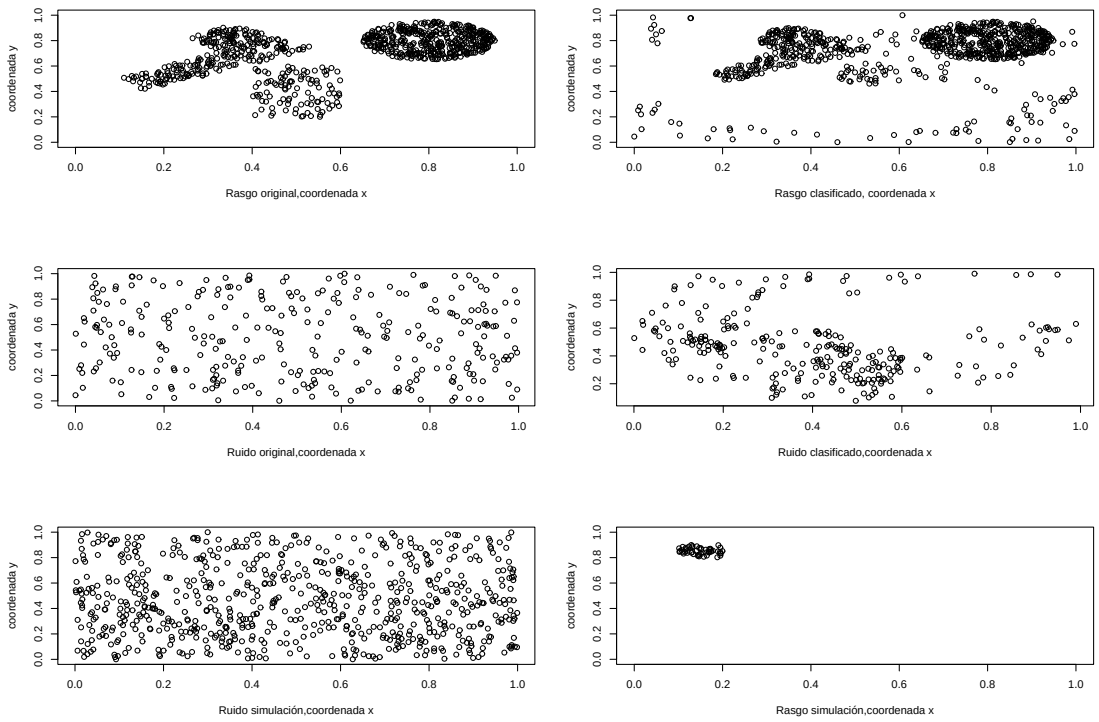

Figura 5.27: Clasificación de tres rasgos. Uno cuadrado de 100 puntos, otro forma L de 250 puntos y otro de forma circular de 416 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

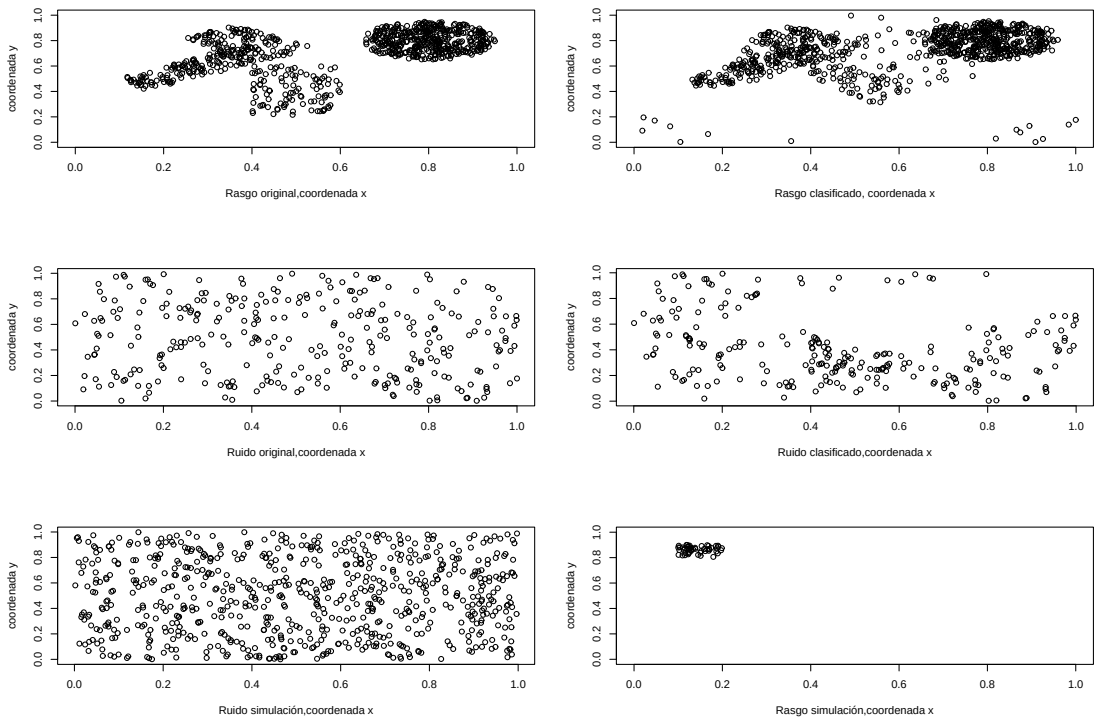

Figura 5.28: Clasificación de tres rasgos. Uno cuadrado de 100 puntos, otro forma L de 250 puntos y otro de forma circular de 428 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 50 puntos en el rasgo simulado y 700 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

Finalmente queda comentar el caso de tres rasgos, con la misma intensidad entre ellos. De todos los casos estudiados mostramos aquellos más representativos, Figuras (5.29) y (5.30), con un 93.12 % y 96.05 % de puntos clasificados, respectivamente.

5.4. Buscando la cantidad de rasgo y ruido simulado

Cierto es que hay ejemplos en los que, al no tener de manera adecuada las cantidades de rasgo y ruido simulado, los resultados no son todo lo satisfactorios que desearíamos. Y es por eso que sería bueno obtener una relación entre las cantidades de rasgo reales y las cantidades de rasgo simuladas en los ejemplos que funcionan bien. Es decir, tener un índice con el que ajustar rápidamente el rasgo simulado en un caso real. Reduciremos la región a estudio. Consideraremos una región 10000 veces más pequeña que la imagen, es decir un cuadrado de 0.01x0.01, de forma que un rasgo cualquiera que nos pudiera interesar no fuera más pequeño que este tamaño. Veamos cuántos puntos caben en este recinto (es muy posible que cuando nos encontremos en rasgo, toda esta región fuera negra, por tanto todos los puntos que cupieran serían rasgo). En el siguiente apartado las fotos digitalizadas que utilizamos tienen 741x555 píxeles y en un cuadrado de las dimensiones mencionadas, habrá 7.41x5.55 puntos de luz, que son 41.07 puntos aproximadamente.

Buscamos ahora la relación de nuestro ejemplo, donde conocemos las cantidades que funcionan bien y trasladaremos la proporción. El rasgo, tiene 300 puntos en un cuadrado de 0.2x0.4 de dimensión. Haciendo el cálculo para reducir este tamaño obtenemos un valor de $\frac{300}{20*40}$ =0.375. Esta sería la cantidad de rasgo original, que hay en una región cuadrada de 0.01x0.01. Si para esta cantidad funciona bien, un rasgo simulado de 80 puntos, situados en un cuadrado de 0.1x0.1, al reducirlo cien veces, para obtener la intensidad de puntos en un cuadrado 0.01x0.01, obtenemos un valor de $\frac{80}{100}$ =0.8. Finalmente, calculando la relación entre las cantidades $\frac{0.8}{0.375}$ =2.13 observamos que el valor obtenido es, prácticamente, el resultado de duplicar la intensidad de puntos en el rasgo simulado respecto del rasgo

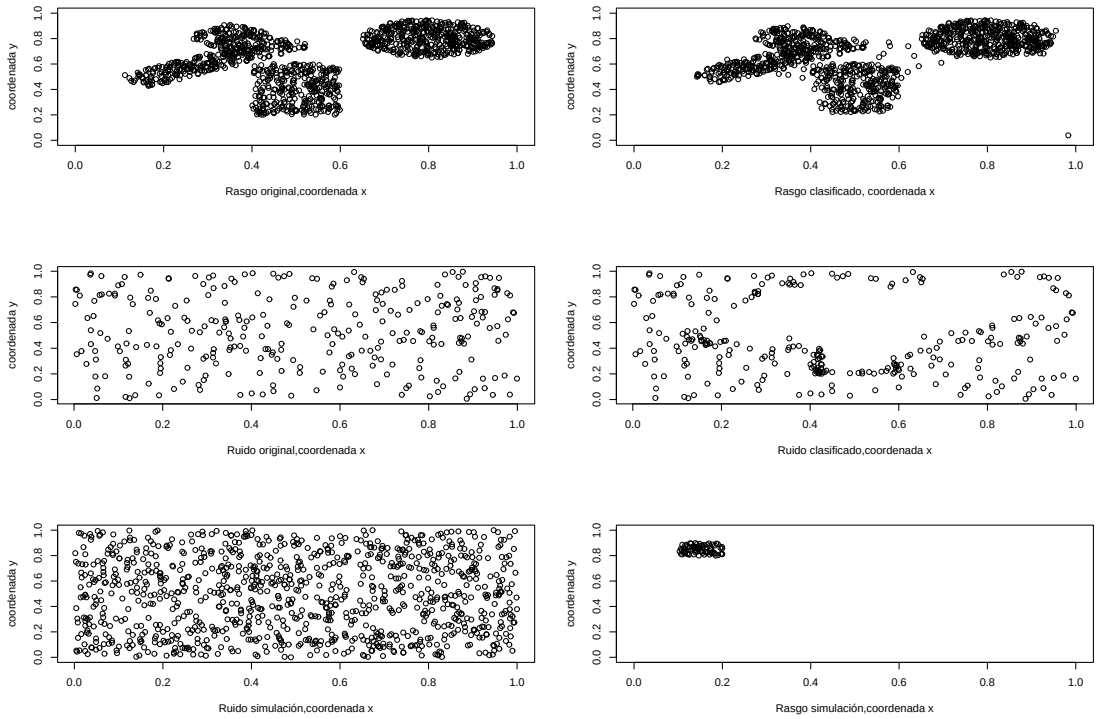

Figura 5.29: Clasificación de tres rasgos. Uno cuadrado de 300 puntos, otro forma L de 400 puntos y otro de forma circular de 434 puntos sobre ruido de 300 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

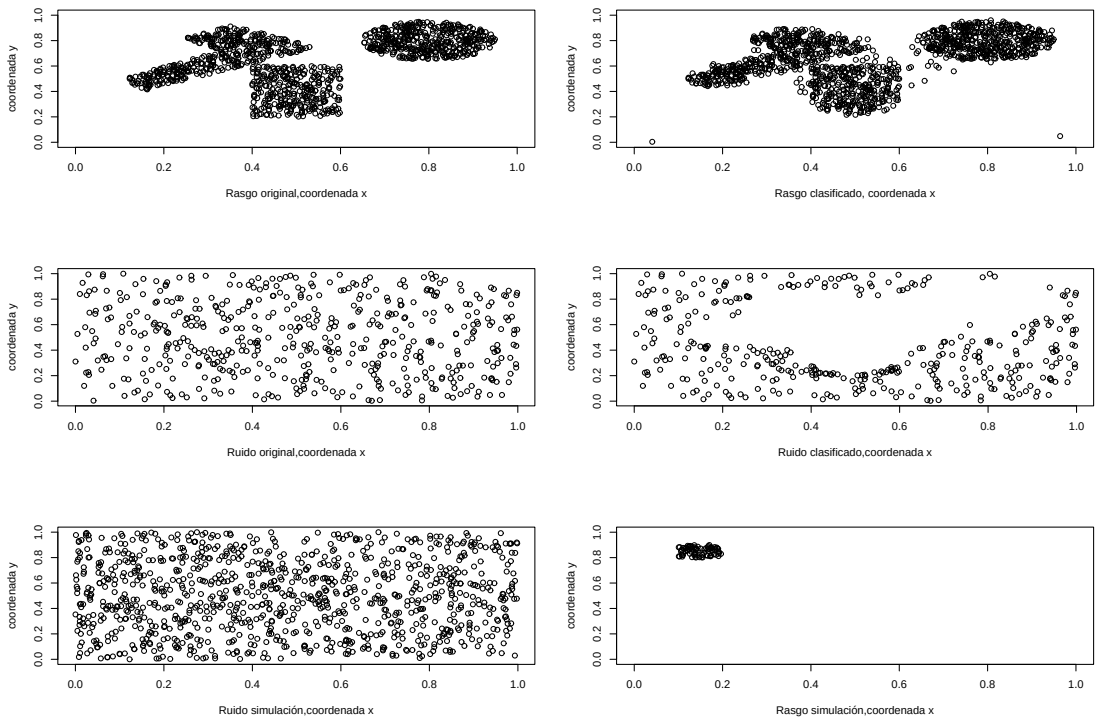

Figura 5.30: Clasificación de tres rasgos. Uno cuadrado de 300 puntos, otro forma L de 400 puntos y otro de forma circular de 426 puntos sobre ruido de 500 puntos en el cuadrado unidad, utilizando 80 puntos en el rasgo simulado y 1000 puntos en el ruido simulado. La distancia utilizada es L2. *Primera fila:* Representación de datos originales y clasificados, (izquierda) Rasgo original, (derecha) Rasgo clasificado. *Segunda fila:* Representación de datos originales y clasificados, (izquierda) Ruido original, (derecha) Ruido clasificado. *Tercera fila:* Representaciones de rasgo y ruido simulados en fase de SEM, (izquierda) Ruido simulado, (derecha) Rasgo simulado.

original. Calculamos estas cantidades para las imágenes que vamos a utilizar en la siguiente sección: $\frac{x}{41.07}$ =2.13, por tanto x =41.07x2.13=87.47. Esta sería la intensidad que ha de contener el rasgo simulado de dimensión 0.01x0.01.

5.5. Clasificaciones de rasgos en presencia de ruido. Imágenes reales

En esta sección utilizaremos imágenes reales que nos permitirán aplicar el método propuesto. Son dos los tipos de imágenes que se van a utilizar y el objetivo a conseguir es el mismo, detectar aquellas zonas que de alguna manera la concentración de puntos es mayor y por tanto puede ser de utilidad aislarlas. A estas zonas son las que les hemos ido denominando rasgo.

En primer lugar utilizamos imágenes médicas, cedidas por el Hospital St. Joan de Déu de Barcelona, correspondiente a "Tejido Conectivo Blando" del Aparato Digestivo. Este tejido se encuentra en la mayoría de los órganos y aparatos del cuerpo humano y en él se pueden distinguir varios tipos de células, entre ellas fibroblastos, células plasmáticas, mastocitos y linfocitos. Para diagnosticar y tratar afecciones de este tipo de tejido, se utilizan tinciones según diferentes técnicas. En este caso ha sido Hematoxilina-Eosina, que tiñe de color más intenso el núcleo celular, de color más pálido el citoplasma (resto de la célula) y no tiñe los espacios intercelulares que en este tipo de tejidos son muy amplios. De todas las estudiadas, elegimos la Figura (5.31) por ser la más representativa. Observamos la tinción y queda claro que los núcleos celulares son el objetivo de nuestro problema. Consideramos estas zonas teñidas el rasgo y el resto de la foto el ruido.

En el campo de la Anatomía Patológica, que estudia las diferentes enfermedades en los tejidos, es interesante la aplicación de técnicas que separen células normales de células patológicas, es decir, afectadas por alguna alteración o anomalía. En este contexto situamos la aplicación de nuestro estudio.

Como ya hemos comentado, las zonas de color más oscuro son el objetivo, queremos aislar este rasgo del resto de puntos que consideraríamos ruido, es decir clasificar los puntos en rasgo y ruido. Para ello, de cada foto, la información que

necesitamos conocer es: coordenadas espaciales de cada punto o píxel de la foto y una etiqueta, que nos indique qué tipo de píxel es, si de color negro (por tanto candidato a ser clasificado como rasgo o ruido) o de color blanco, que de entrada, quedaría descartado, porque el blanco lo consideramos ausencia de puntos del proceso puntual.

Hay que hacer la observación siguiente, y es que una imagen es una señal continua, es decir, está definida en un espacio continuo. El proceso de capturar esta imagen, para un posterior análisis de ella, se denomina digitalización de imágenes. Este proceso consiste en el paso del espacio continuo en que está definida a uno discreto, en particular del espacio R^2 al espacio Z^2, lo cual se consigue mediante la división de la imagen en pequeñas áreas. Esas áreas son denominadas píxeles (de la abreviatura de la denominación inglesa pixel de picture element). A cada píxel se le asigna el valor medio registrado en el área a la que representa. Siempre vamos a considerar imágenes digitales discretas y así los píxeles sólo van a tomar valores en un conjunto discreto. En el caso de las imágenes binarias los valores posibles están en el conjunto (0,1).

Más formalmente, una imagen binaria f es una aplicación de un subconjunto D de Z^2 llamado el dominio de definición de f en el conjunto (0,1).

Utilizamos varias imágenes del mismo tipo pero con formas y colores diferentes. Sometemos al mismo proceso a todas, considerando las tres distancias para calcular el espacio entre los vectores de funciones LISA y el valor esperado de la media teórica, divergencia de Kullback, distancia L1 o distancia L2. Seguimos la proporción calculada para aproximar la cantidad de ruido y rasgo simulado.

Para este primer tratamiento de la imagen, hacemos uso del programa MATROX y la salida binaria que nos ofrece es la que vamos a utilizar. Con estos datos, lanzaremos nuestro programa de clasificación y el resultado se muestra en la Figura (5.32).

Esta imagen consta de seis gráficos, el primero de ellos es la función original, tal como la obtenemos del MATROX, la segunda es el histograma de las distancias de los vectores de funciones LISA individuales a la media teórica, en esta ocasión hemos utilizado la distancia L2. En ella se aprecia una bimodalidad muy suavizada dado que el ruido, que está claramente definido, queda agrupado en la primera columna. El siguiente gráfico muestra la clasificación a través de funciones LISA,

donde quedan aisladas totalmente las zonas más oscuras, aún siendo cierto que hay algún punto alrededor que se clasifica como rasgo sin serlo. Hay que considerar que estamos con una imagen de 14910 puntos, que es muy superior en cantidad, a cualquiera de los ejemplos que hemos estado barajando en la fase de simulación, por tanto los resultados los consideramos satisfactorios. El siguiente gráfico es el ruido clasificado. Las dos últimas son el ruido simulado y el rasgo simulado en la zona de dimensión 0.01x0.1.

Cabe mencionar un último comentario de esta imagen respecto del coste computacional. La clasificación completa, de una cantidad de puntos como esta, tarda aproximadamente 2 horas, en cualquiera de los casos de las imágenes. Para cada uno de los 14910 puntos se calculan vectores de 100 componentes de funciones LISA individuales, distancias entre ellos y todo eso forma parte de un bucle, donde se van calculando, en cada iteración, las clasificaciones de rasgo y ruido, hasta cumplir la convergencia, por tanto, en cada iteración, funciones LISA de rasgo y ruido simulado, distancias entre ellas, histogramas, etc, etc.

Una segunda aplicación de nuestro método, en un ámbito muy diferente a la medicina, ha sido en geografía.

En la Figura (5.33), se muestra una imagen de la costa de California. Esta costa está afectada por numerosos y famosos terremotos, como el de San Francisco y otros. El choque de dos placas tectónicas, producen la falla de San Andrés y es por este motivo que en esa zona se acumulan fenómenos de movimientos sísmicos continuos. Debido a que es una zona de gran densidad de población, hay muchos estudios realizados al respecto y gran parte de los presupuestos de los gobiernos locales y centrales de EEUU se destinan a controlar, prevenir y paliar los efectos.

Controlar al máximo el trazado interno de la falla, con los datos anteriores de los que se dispongan, contribuye a tener más localizados los epicentros de futuros terremotos y ayuda a los esfuerzos gubernamentales.

En este sentido y a partir de una imagen de epicentros de terremotos de la costa de California, Byers & Raftery (1998) [18], con el método del K-vecino más próximo (algoritmo EM), proporcionan una clasificación en rasgo y ruido. La aportación de su trabajo fue distinguir la falla de San Andrés al clasificar el

rasgo, dado que había más acumulación de epicentros de terremotos. Este método está comentado en el Capítulo 3 y es uno de los puntos de partida sobre los que se sustenta este trabajo. Es por este motivo que se dedica una sección de este capítulo a un análisis comparativo entre el algoritmo EM y el algoritmo SEM.

La imagen es de satélite, por tanto digitalizada y de ella conocemos las localizaciones en coordenadas de los puntos de color negro, que son las mismas que se utilizaron en el mencionado trabajo.

Utilizando estos mismos datos, dado que son públicos y de libre acceso y con ayuda de las funciones LISA individuales, obtenemos los siguientes resultados, que se pueden observar en la Figura (5.34).

Al igual que en la imagen anterior, en primer lugar, aparece el gráfico de las localizaciones originales de los terremotos, seguido del histograma de las distancias de los vectores de funciones LISA a la media teórica, utilizando en esta ocasión la divergencia de Kullback. Queda muy clara la bimodalidad del histograma, no tan suavizada como en la imagen anterior, ya que en este caso estamos hablando de 2049 datos. Los siguientes dos gráficos son las clasificaciones de rasgo y ruido, donde el rasgo es muy similar al encontrado por Byers & Raftery (1998) [18], con una delimitación clara de la falla. Por último, las dos imágenes referentes al ruido y al rasgo simulado.

Con estos resultados, nos parece evidente, pues, la aplicación y el buen funcionamiento de las funciones LISA individuales a contextos tan diferentes como son la medicina y la geografía, con resultados igualmente satisfactorios.

5.6. Análisis comparativo entre el algoritmo EM y el algoritmo SEM

A lo largo de los capítulos y secciones anteriores, hemos presentado dos métodos estadísticos, el basado en la distancia al K-ésimo vecino más cercano y el basado en las distancias entre funciones LISA, para la clasificación y detección de rasgos en el contexto de procesos puntuales espaciales. El desarrollo del método de distancias al K-ésimo vecino más cercano pasa por el algoritmo EM (Banfield

& Raftery, 1992 [7]), mientras que el método de distancias entre funciones LISA se lleva a la práctica mediante un SEM (Cressie & Brant, 1997 [27]).

En esta sección, se pretende comparar ambos métodos con el fin de evaluar las diferencias y semejanzas de ambas teorías, analizar qué método funciona mejor y bajo qué circunstancias y en particular, con el objetivo de determinar si el método basado en LISA y SEM mejora el propuesto en Banfield & Raftery, 1992 [7]) basado en distancias al K-ésimo vecino más cercano y EM. Esta comparación se hará bajo unos determinados elementos que cubren una casuística suficientemente general. En particular, presentamos en las Figuras (5.35) y (5.36), las dos situaciones que utilizamos para la comparación, una basada en un único rasgo y la otra en tres rasgos con formas cuadrada, circular y forma de L. Las Tablas (5.1) y (5.2) representan las combinaciones analizadas y el número de repeticiones asociadas a cada combinación. Para cada uno de los algoritmos tomamos una variabilidad de repeticiones, tal y como se muestra en las Tablas (5.1) y (5.2), en función de los parámetros que se pueden controlar.

	K=5	**K=10**	**K=15**
Sin corrección de borde	100 repeticiones	100 repeticiones	100 repeticiones
Con corrección de borde	100 repeticiones	100 repeticiones	100 repeticiones

Tabla 5.1: Resumen de número de repeticiones realizadas en el estudio de simulación bajo el algoritmo EM, para K=5, 10 y 15 vecino más próximo, sin utilizar y utilizando corrección de borde en el cuadrado unidad.

En la Tabla (5.1), los valores K=5, K=10 y K=15 recogen la distancia al 5-vecino más próximo, 10-vecino más próximo y 15-vecino más próximo, respectivamente. La consideración de "sin corrección de borde" o "con correccción de borde" se refiere a la corrección de borde toroidal, según la cual, al calcular distancias de individuos próximos a algún extremo de la región del plano, esta corrección hace coincidir el límite referido con el opuesto del mismo plano, consiguiendo que el espacio considerado sea continuo. Al tomar una región cuadrada, con la coincidencia de los límites opuestos se obtiene un toro, de ahí el

nombre de corrección de borde toroidal. En nuestro caso, hemos tomado las dos posibilidades, considerar dicha corrección o no.

	L1	**L2**	**Kullback**
Sin corrección de borde	100 repeticiones	100 repeticiones	100 repeticiones
Con corrección de borde	100 repeticiones	100 repeticiones	100 repeticiones

Tabla 5.2: Cuadro resumen de número de repeticiones realizadas en el estudio de simulación bajo el algoritmo SEM, utilizando las distancias L1, L2 y Kullback, sin utilizar y utilizando corrección de borde en el cuadrado unidad.

En la Tabla (5.2), L1, L2 y Kullback son las distancias que hemos utilizado entre las funciones LISA individuales y el valor esperado de la función LISA bajo aleatoriedad. Del mismo modo, se utilizan las consideraciones de corrección de borde toroidal. En las Tablas (5.3), (5.4) (5.5) y (5.6), la condición "con corrección de borde" la denotaremos por CCB y la condición "sin corrección de borde" la denotaremos por SCB.

Para cada conjunto de 100 repeticiones realizadas calculamos la media sobre una serie de ítems que nos sirven de comparación. En el caso del algoritmo EM, los ítems sobre los que aportaremos información, son: *Rasgo clasificado,* n^o de puntos que el algoritmo EM clasifica como rasgo del total; *Ruido clasificado,* n^o de puntos que el algoritmo EM clasifica como ruido del total; *Buena clasificación,* n^o de puntos correctamente clasificados, es decir, eran inicialmente rasgo y han sido clasificados como rasgo o eran inicialmente ruido y los ha clasificado como ruido; *Falsa clasificación,* n^o de puntos clasificados como rasgo o ruido, que originalmente no lo eran; *% buena clasificación,* porcentaje de buena clasificación; *% falsa clasificación,* porcentaje de falsa clasificación.

Para el algoritmo SEM, la información utilizada es: *n^o Iteraciones,* donde se proporciona el número de iteraciones que el algoritmo SEM utiliza para la realización de la clasificación; *Rasgo clasificado,* n^o de puntos que el algoritmo SEM clasifica como rasgo del total; *Ruido clasificado,* n^o de puntos que el algoritmo SEM clasifica como ruido del total; *Buena clasificación,* n^o de puntos correctamente clasificados, es decir, eran inicialmente rasgo y los ha clasificado como rasgo

o eran inicialmente ruido y los ha clasificado como ruido; *Falsa clasificación,* nº de puntos clasificados como rasgo o ruido, que originalmente no lo eran; *% buena clasificación,* porcentaje de buena clasificación; *% falsa clasificación,* porcentaje de falsa clasificación.

El primero de los ejemplos analizados para este estudio, ver Figura (5.35), consta de un rasgo en el que aleatoriamente se distribuyen 400 puntos en una región cuadrada, sobre un ruido de 400 puntos en el cuadrado unidad. En las Tablas (5.3) y (5.4), se muestran los resultados obtenidos. Los valores que aparecen en cada celda son la media aritmética de las correspondientes a las 100 repeticiones.

	K=5	**K=10**	**K=15**
SCB	Rasgo clasif.: 437.6 Ruido clasif.: 362.4 Buena Clasif.: 397.7 Falsa Clasif.: 2.3 % buena clasif.: **99.4** % falsa clasif.: **0.6**	Rasgo clasif.: 437.8 Ruido clasif.: 362.2 Buena Clasif.: 399.4 Falsa Clasif.: 0.6 % buena clasif.: **99.8** % falsa clasif.: **0.2**	Rasgo clasif.: 480.2 Ruido clasif.: 319.8 Buena Clasificac.: 398.6 Falsa Clasif.: 1.4 % buena clasif.: **99.7** % falsa clasif.: **0.3**
CCB	Rasgo clasif.: 435.2 Ruido clasif.: 364.8 Buena Clasif.: 395.2 Falsa Clasif.: 4.8 % buena clasif.: **89** % falsa clasif.: **11**	Rasgo clasif.: 474.8 Ruido clasif.: 325.2 Buena Clasif.: 397.9 Falsa Clasif.: 2. % buena clasif.: **99.5** % falsa clasif.: **0.5**	Rasgo clasif.: 480 Ruido clasif.: 320 Buena Clasif.: 398.6 Falsa Clasif.: 1.4 % buena clasif.: **99.65** % falsa clasif.: **0.35**

Tabla 5.3: Cuadro resumen de resultados obtenidos para 100 repeticiones, en media aritmética, para un estudio de simulación bajo un proceso puntual de 400 puntos en un rasgo cuadrado sobre 400 puntos de ruido en el cuadrado unidad, utilizando el algoritmo EM para 5, 10 y 15 vecino más próximo y considerando SCB: "sin corrección de borde" y CCB: "con corrección de borde".

Como ya se apunta en el artículo de Banfield & Raftery, 1992 [7], encontrar el valor de K que haga óptima la clasificación, utilizando el algoritmo EM, es la clave del éxito del método. En este trabajo aludido se propone una intervención

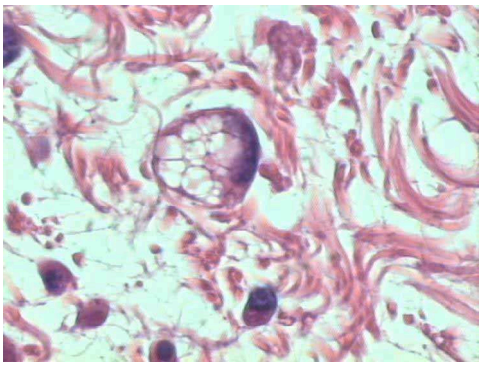

Figura 5.31: Tejido conectivo de órgano humano

	L1	**L2**	**Kullback**
SCB	Núm. de it.: 56.4 Rasgo clasif.: 431.4 Ruido clasif.: 368.6 Buena Clasif.: 389.9 Falsa Clasif.: 10.1 % buena clasif.: **97.8** % falsa clasif.: **2.2**	Núm. de it.: 28.1 Rasgo clasif.: 434.6 Ruido clasif.: 365.4 Buena Clasif.: 398.4 Falsa Clasif.: 1.6 % buena clasif.: **99.6** % falsa clasif.: **0.4**	Núm. de it.: 54 Rasgo clasif.: 482.7 Ruido clasif.: 317.3 Buena Clasif.: 400 Falsa Clasif.: 0 % buena clasif.: **100** % falsa clasif.: **0**
CCB	Núm. de it.: 43.9 Rasgo clasif.: 482.7 Ruido clasif.: 317.3 Buena Clasif.: 400 Falsa Clasif.: 0 % buena clasif.: **100** % falsa clasif.: **0**	Núm. de it.: 17.8 Rasgo clasif.: 439.3 Ruido clasif.: 360.7 Buena Clasif.: 392.6 Falsa Clasifi.: 7.4. % buena clasif.: **98.15** % falsa clasif.: **1.85**	Núm. de it.: 56.5 Rasgo clasif.: 470.8 Ruido clasif.: 329.2 Buena Clasif.: 396.6 Falsa Clasif.: 3.4 % buena clasif.: **99.2** % falsa clasif.: **0.8**

Tabla 5.4: Cuadro resumen de resultados obtenidos para 100 repeticiones, en media aritmética, para un estudio de simulación bajo un proceso puntual de 400 puntos en un rasgo cuadrado sobre 400 puntos de ruido en el cuadrado unidad, utilizando el algoritmo SEM para los tres tipos de distancias L1, L2 y Kullback y considerando SCB: "sin corrección de borde" y CCB: "con corrección de borde"

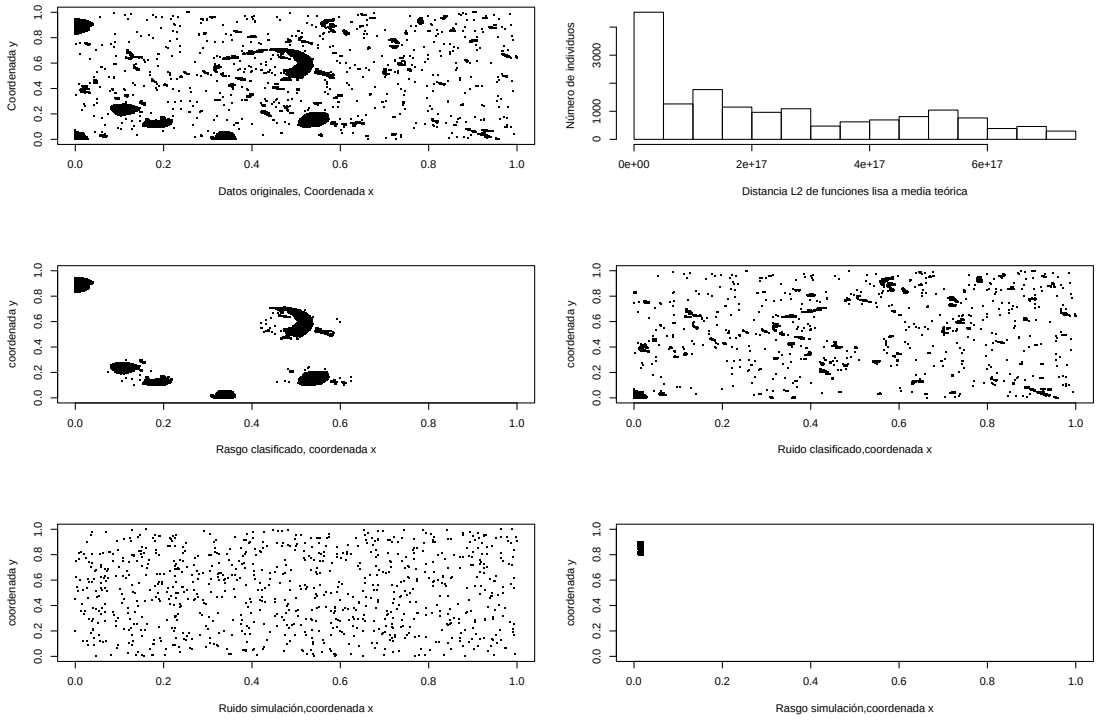

Figura 5.32: Clasificación de imagen médica de la Figura (5.31). *Primera fila:* (izquierda) salida del programa MATROX, (derecha) histograma de las distancias de las funciones LISA, utilizando L2. *Segunda fila:* (izquierda) rasgo clasificado, (derecha) ruido clasificado. *Tercera fila:* (izquierda) ruido simulado, (derecha) rasgo simulado.

Figura 5.33: Costa de California

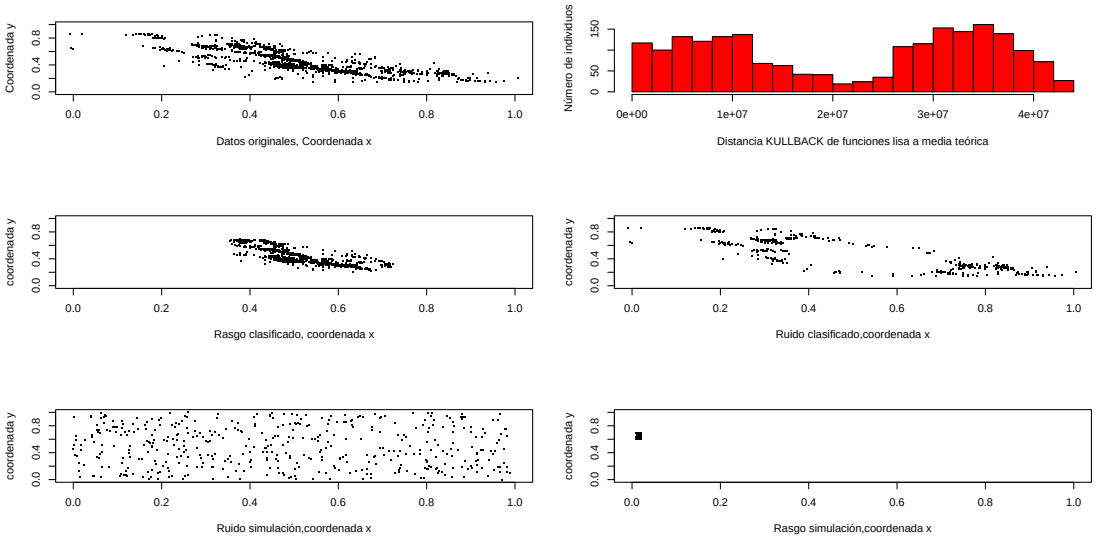

Figura 5.34: Clasificación de las posiciones de los epicentros de los terremotos en California, utilizando funciones LISA y divergencia de Kullback. *Primera fila:* (izquierda) representación de los epicentros de los terremotos, en California, (derecha) histograma de las distancias de las funciones LISA, utilizando Divergencia de Kullback. *Segunda fila:* (izquierda) rasgo clasificado, (derecha) ruido clasificado. *Tercera fila:* (izquierda) ruido simulado, (derecha) rasgo simulado.

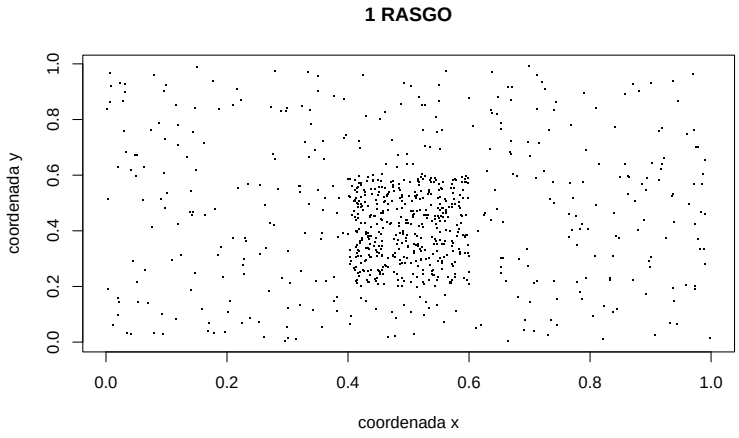

Figura 5.35: Rasgo cuadrado de 400 puntos sobre ruido de 400 puntos en el cuadrado unidad.

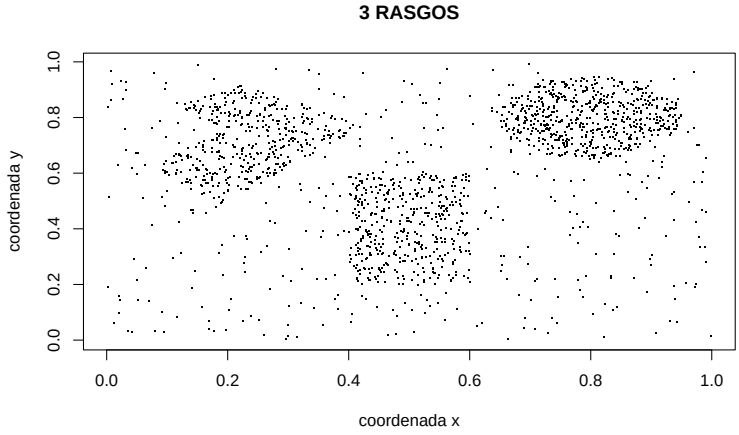

Figura 5.36: Tres rasgos, cuadrado, forma L y circular de 300, 400 y 434 puntos respectivamente, sobre ruido de 300 puntos en el cuadrado unidad.

directa, de forma que el investigador que va a utilizar el algoritmo, va aumentando el valor de K sucesivamente. El proceso se repite hasta que los resultados sean satisfactorios o la mejoría entre el valor de K utilizado en dos simulaciones del algoritmo sucesivas, sea despreciable. Es evidente que el criterio puede variar de un investigador a otro y por tanto, nos encontramos ante una desventaja de este método frente al algoritmo SEM que proponemos, ya que el algoritmo *para* cuando cumple un criterio de convergencia. En él también se puede inferir, haciendo que sea más o menos exigente este criterio de convergencia, pero es un hecho más objetivo, cuanto menos. El número medio de iteraciones utilizadas en las tres posibilidades de distancias, no excede en ningún caso de 15 minutos de tiempo computacional y el criterio de convergencia impuesto arroja resultados que no bajan del 97.8 % de porcentaje de buena clasificación. En este sentido, encontramos un 89 % de porcentaje de buena clasificación, como dato más bajo, para el algoritmo EM, en el caso de K=5, utilizando corrección de borde.

El segundo de los ejemplos elegidos para esta comparación, ver Figura (5.36), consta de tres rasgos en el que aleatoriamente se distribuyen 300, 400 y 434 puntos en rasgos con forma cuadrada, forma L y circular, respectivamente, sobre un ruido de 300 puntos en el cuadrado unidad. Las Tablas (5.5) y (5.6) muestran los resultados obtenidos.

Podemos observar que la clasificación en esta situación, presenta mejores resultados con el algoritmo EM que el SEM, pero analizando los datos, se trata de un porcentaje de diferencia a favor del EM, de un 9 % o un 10 %.

Hay que hacer notar la complicación metodológica de la utilización de funciones LISA individuales y la carga teórica que lleva asociada. Involucrar características de segundo orden referentes a la agregación del patrón puntual, hace que sea un método más completo que la utilización del algoritmo EM, que únicamente utiliza distancias al K-ésimo vecino más cercano. A pesar de ello detectamos mejoras del algoritmo SEM frente a EM a niveles de porcentaje de buena clasificación, si cabe, dado que eran ya muy altos utilizando distancias al K-ésimo vecino más cercano, en el ejemplo de detección de un rasgo. Sin embargo, en la detección de tres rasgos, teniendo en cuenta la complejidad de este caso, por las combinaciones de puntos de ruido y rasgo y la forma de ellos, la detección basada en funciones LISA se encuentra en un porcentaje del 10 % por debajo, en términos de clasificación correcta, respecto al K-ésimo vecino más próximo. Mencionar

	K=5	**K=10**	**K=15**
SCB	Rasgo clasif.: 1182.7 Ruido clasif.: 251.3 Buena Clasif.: 1124.6 Falsa Clasif.: 9.4 % buena clasif.: **99.48** % falsa clasif.: **0.52**	Rasgo clasif.: 1312 Ruido clasif.o: 122 Buena Clasif.: 1127.5 Falsa Clasif.: 6.5 % buena clasif.: **99.43** % falsa clasif.: **0.57**	Rasgo clasif.: 1329.2 Ruido clasif.: 104.8 Buena Clasif.: 1130.5 Falsa Clasif.: 3.5 % buena clasif.: **99.67** % falsa clasif.: **0.33**
CCB	Rasgo clasif.: 1191.4 Ruido clasif.: 242.6 Buena Clasif.: 1124.5 Falsa Clasif.: 9.5 % buena clasif.: **96.9** % falsa clasif.: **3.1**	Rasgo clasif.: 1309.7 Ruido clasif.: 124.9 Buena Clasif.: 1126.5 Falsa Clasif.: 7.5 % buena clasif.: **99.33** % falsa clasif.: **0.67**	Rasgo clasif.: 1329.3 Ruido clasif.: 104.7 Buena Clasif.: 1128.8 Falsa Clasif.: 5.2 % buena clasif.: **99.53** % falsa clasif.: **0.47**

Tabla 5.5: Cuadro resumen de resultados obtenidos para 100 repeticiones, en media aritmética, para un estudio de simulación bajo un proceso puntual de 300 puntos en un rasgo cuadrado, 400 puntos en un rasgo forma L y 434 puntos sobre un rasgo circular, sobre 400 puntos de ruido en el cuadrado unidad, utilizando el algoritmo EM para 5, 10 y 15 vecino más próximo y considerando SCB: "sin corrección de borde" y CCB: "con corrección de borde"

	L1	**L2**	**Kullback**
SCB	Núm. de it.: 86.1 Rasgo clasif.: 1116.3 Ruido clasif.: 317.7 Buena Clasif.: 1055 Falsa Clasif.: 79 % buena clasif.: **88.57** % falsa clasif.: **11.43**	Núm. de it.: 74.6 Rasgo clasif.: 1090 Ruido clasif.: 343 Buena Clasif.: 1040.56 Falsa Clasif.: 93.44 % buena clasif.: **91.17** % falsa clasif.: **8.83**	Núm. de it.: 122.1 Rasgo clasif.: 1079.3 Ruido clasif.: 354.7 Buena Clasif.: 1052.2 Falsa Clasif.: 81.8 % buena clasif.: **91.99** % falsa clasif.: **8.01**
CCB	Núm. de it.: 92 Rasgo clasif.: 1239.6 Ruido clasif.: 194.4 Buena Clasif.: 1061.3 Falsa Clasif.: 72.7 % buena clasif.: **93.52** % falsa clasif.: **6.48**	Núm. de it.: 53 Rasgo clasif.: 1006.2 Ruido clasif.: 427.8 Buena Clasif.: 1040.7 Falsa Clasif.: 93.3 % buena clasif.: **92.23** % falsa clasif.: **7.77**	Núm. de it.: 116.7 Rasgo clasif.: 1153.7 Ruido clasif.: 280.3 Buena Clasif.: 1003.3 Falsa Clasif.: 130.7 % buena clasif.: **88.43** % falsa clasif.: **11.57**

Tabla 5.6: Cuadro resumen de resultados obtenidos para 100 repeticiones, en media aritmética, para un estudio de simulación bajo un proceso puntual de 300 puntos en un rasgo cuadrado, 400 puntos en un rasgo forma L y 434 puntos sobre un rasgo circular, sobre 400 puntos de ruido en el cuadrado unidad, utilizando el algoritmo SEM para los tres tipos de distancias L1, L2 y Kullback y considerando SCB: "sin corrección de borde" y CCB: "con corrección de borde"

que, en los términos en los que nos movemos, un porcentaje del 10 % no es un descenso dramático y de hecho, en las secciones anteriores hemos comprobado que la clasificación es absolutamente aceptable. Por tanto, a la luz de estos resultados comparativos, podemos considerar que el método de funciones LISA corroborado con el SEM es un método competitivo y fiable para la detección de rasgos.

Capítulo 6

Detección de rasgo en presencia de ruido. Análisis Cluster y Escalamiento multidimensional

Indice General

6.1. Introducción 167

6.2. Distribución teórica de las funciones LISA 169

6.3. Análisis cluster jerárquico 171

6.4. Escalamiento multidimensional 172

6.5. Clasificaciones de rasgos en presencia de ruido. Ejemplos simulados 174

6.6. Clasificaciones de rasgos en presencia de ruido. Imágenes reales 187

6.1. Introducción

En el contexto en el que se enmarca nuestro problema, detección de rasgos en presencia de ruido, abordamos en este capítulo una técnica diferente para la clasificación. El punto de partida es el mismo, un patrón puntual del que conocemos coordenadas cartesianas de las localizaciones de los puntos respecto

del cuadrado unidad (nuevamente suponer el cuadrado unidad no es inconveniente ninguno, ni resta generalidad a los resultados).

Calculamos el mismo vector para cada localización, cuyas componentes serán las funciones LISA individuales y aunque en esencia comparte la raíz con el capítulo anterior (en él se utilizaba la distancia entre vectores de funciones LISA individuales al valor esperado de una función LISA bajo aleatoriedad), en esta ocasión, la distancia, no se calculará del mismo modo.

Calcularemos ahora la distancia de Mahalanobis entre todos los vectores de funciones LISA individuales correspondientes a cada uno de los puntos, obteniendo una matriz D de distancias. Cada una de las entradas de esta matriz, será la distancia entre un par de vectores de funciones LISA individuales, correspondientes a las observaciones de los datos.

Cressie & Brant (1997) [27], proponen aplicar escalamiento multidimensional (Multidimesional Scaling) a esta matriz de distancias, de forma que este análisis clásico de obtención de clusters, detecte los grupos en los datos. En el caso en el que ellos lo aplicaron, cada "individuo" considerado era ya un grupo de individuos y establecían relaciones entre grupos. En nuestro contexto, la diferencia se encuentra en que se va a aplicar a todos los puntos de manera individual y los grupos que se detectan, son el rasgo o rasgos y el ruido. La importancia de la técnica, al igual que en el capítulo anterior, estriba en la no utilización de modelos probabilísticos que den estructura teórica y "conocida" para la obtención de la clasificación.

Utilizaremos pues, la técnica propuesta por Cressie, escalamiento multidimensional y una técnica conocida, Cluster Jerárquico para la clasificación, comentando ventajas e inconvenientes en el estudio de simulación. En la parte última del capítulo hay aplicaciones a casos reales, concretamente en un campo de la geoestadística espacial: prevención de incendios. Los datos que vamos a utilizar, son 412 posiciones de incendios de la Comunidad Valenciana (España), en los años 2000-2001. El objetivo es detectar zonas de mayor intensidad de concentración de incendios y comparar el resultado con el mapa físico de la Comunidad Valenciana, con el fin de detectar las zonas de mayor número de incidencias.

6.2. Distribución teórica de las funciones LISA

Sea el vector asociado a la observación i-ésima, s_i, formado por el valor de la función LISA evaluada en m distancias,

$$Y_i = \left[\widehat{\rho}_{\varepsilon}^{(i)}(r_1), \widehat{\rho}_{\varepsilon}^{(i)}(r_2), ..., \widehat{\rho}_{\varepsilon}^{(i)}(r_m)\right]^T; \quad i = 1, ..., n$$

donde cada una de las componentes del vector, son:

$$\widehat{\rho}_{\varepsilon}^{(i)}(r_t) = \frac{n-1}{2\pi r_t} \sum_{j \neq i} f_{\varepsilon}(\|s_i - s_j\| - r_t); r_t > \varepsilon > 0, \qquad t = 1, ..., m$$

Una agrupación de funciones LISA, consistirá en un conjunto de vectores definidos anteriormente (un vector para cada coordenada u observación) de la forma:

$$\{Y_i : i = 1, ..., n\}$$

El objetivo es demostrar que detectar y aislar estas agrupaciones de los vectores LISA, haciendo corresponder cada uno de los vectores a los puntos originales, por tanto, detectar y aislar estos conjuntos de puntos, dará como resultado una correcta clasificación en rasgo y ruido.

Podemos establecer diversas medidas de distancia entre estos vectores. Descartamos la distancia Euclídea, que aunque es la más utilizada en muchos contextos, no aporta información acerca de la correlación que pueda existir entre las variables.

En nuestro caso, que podemos llegar a conocer la varianza de los vectores LISA y la covarianza entre ellos, por tanto, la matriz de varianzas-covarianzas y dado que asumimos una fuerte correlación entre los vectores de funciones LISA individuales que se encuentren en rasgo o ruido, optamos por la modificación de la distancia Euclídea, introduciendo la inversa de la matriz comentada, dando lugar a la utilización de la distancia de Mahalanobis.

Dada esta medida de distancia entre dos cualesquiera de estos vectores, podemos crear una matriz simétrica de distancias cuyas entradas serían:

$$d_{il} = [Y_i - Y_l]^T \Sigma_{il}^{-1} [Y_i - Y_l] \;\; i, l \in \{1, ..., n\} \tag{6.1}$$

donde Σ_{il} es la matriz de varianzas-covarianzas del vector $[Y_i - Y_l]$ cuyos valores, ya comentados en el Capítulo 4 son:

$$\begin{aligned} Var_{!!}\left[\widehat{\rho}_{\varepsilon}^{(i)}(r_k) - \widehat{\rho}_{\varepsilon}^{(l)}(r_m)\right] &= \\ &= \frac{\frac{\lambda}{a}}{(2\pi r_k)^2}\left[g(s_i, s_l, r_k)\, g(s_l, s_i, r_k)\right]^2 + \\ &+ \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{(2\pi r_k)^2}\int_A \left[g(s_i, s, r_k)\, g(s_i, s, r_m)\right]^2 ds \end{aligned}$$

$$\begin{aligned} &Cov_{!!}\left[\left(\widehat{\rho}_{\varepsilon}^{(i)}(r_k) - \widehat{\rho}_{\varepsilon}^{(l)}(r_k)\right), \left(\widehat{\rho}_{\varepsilon}^{(i)}(r_m) - \widehat{\rho}_{\varepsilon}^{(l)}(r_m)\right)\right] = \\ = \;& \frac{\frac{\lambda}{a}}{(2\pi)^2 r_k r_m}\left[g(s_i, s_l, r_k)\, g(s_l, s_i, r_k)\right]\left[g(s_i, s_l, r_m)\, g(s_l, s_i, r_m)\right] \\ + \;& \frac{\lambda^3 + 5\lambda^2/a + 2\lambda/a^2}{(2\pi)^2 r_k r_m}\int_A \left[g(s_i, s, r_k) - g(s_l, s, r_k)\right]\left[g(s_i, s_l, r_m)\, g(s_l, s_i, r_m)\right] ds \end{aligned}$$

Usamos la estimación de λ del proceso de Poisson como:

$$\widehat{\lambda} = \frac{n-2}{a}$$

Utilizaremos estas expresiones para procesos puntuales estacionarios e isotrópicos (pero no necesariamente homogéneos de Poisson).

Dos sucesos cercanos s_i y s_l con características parecidas, deben mostrar una correlación fuerte entre Y_i y Y_l y como consecuencia, una distancia de Mahalanobis reducida. Además, conjeturamos que los puntos de los rasgos tendrán funciones LISA más parecidas entre ellos y diferentes de las funciones LISA de los sucesos del ruido.

6.3. Análisis cluster jerárquico

Análisis "cluster", o de conglomerados, es un nombre genérico utilizado para identificar una gran variedad de procedimientos usados para establecer una clasificación. Estos métodos parten de un conjunto de datos con información relevante acerca de una serie de casos (para nuestro problema localizaciones espaciales de puntos y a partir de ellos, vectores de funciones LISA individuales) e intentan reorganizarlos en grupos homogéneos, en base a esta información. Así, el análisis cluster permite la utilización de diferentes algoritmos de clasificación. La mayoría de ellos son procedimientos relativamente simples y en muchas ocasiones no están basados en un cuerpo de razonamiento estadístico extenso. Un análisis cluster siempre situará a los elementos en distintos grupos, que además arrojarán resultados distintos dependiendo del método concreto de agrupación utilizado, por lo que la clave para aplicar con propiedad el análisis cluster es reconocer cuando estos grupos son reales y no meras imposiciones del método a los datos. Es por esta razón que para la validación estadística de los resultados obtenidos, además de para evaluar la capacidad de discriminación entre los grupos que tienen las distintas variables, hayamos realizado un análisis de escalamiento multidimensional.

El método jerárquico de clasificación comienza de manera que cada caso forma un grupo. En el primer paso, la distancia entre casos (que en nuestro contexto son distancias de Mahalanobis) viene dada por la distancia Euclídea, pero cuando más adelante, se trata de unir clusters, hay que especificar qué regla de agregación se va a utilizar. Hemos utilizado el método de Ward, cuya idea básica trata de ir agrupando de forma jerárquica elementos de modo que se minimice una determinada función objetivo. Esta función objetivo perseguirá la minimización de la Variación Intra Grupal de la estructura formada. Utiliza una aproximación al análisis de la varianza para evaluar la distancia entre clusters, intentando minimizar la suma de los cuadrados de los residuos de cada dos hipotéticos clusters que se pueden formar en cada paso. Su representación gráfica se llama dendrograma.

Dado que el objetivo es crear grupos relativamente homogéneos, en el análisis de un dendrograma resultante de una agrupación jerárquica, donde se sugieren agrupaciones crecientes en número de casos, la cuestión es dónde "cortar" el gráfico de manera que se obtenga el número óptimo de grupos.

Gran parte de la dificultad para resolver este problema es la ausencia de una hipótesis nula, debido a la falta de una definición consistente de la estructura y contenido de un cluster. Una posible hipótesis nula sería: "no estructura" en un conjunto de datos, pero ésta está lejos de ser una proposición clara y fácilmente manejable.

En cualquier caso, los procedimientos heurísticos son los métodos más utilizados para decidir el nivel de "corte" en un dendrograma, mediante la inspección subjetiva de los diferentes niveles del árbol.

Por último, hay que distinguir entre la utilización de este método para Métodos Clustering Jerárquico en Modelos Probabilísticos MBHC, del Capítulo 3, que como indica la denominación, utiliza modelos de probabilidad para la distribución de las distancias entre las localizaciones de los individuos, y la utilización de este método en el presente Capítulo donde las distancias entre las funciones LISA individuales no siguen una distribución de probabilidad conocida.

6.4. Escalamiento multidimensional

La técnica que hoy en día conocemos como escalamiento multidimensional, se desarrolla a finales del siglo XIX en el ámbito de la Psicología, concretamente, en la Psicofísica, con la pretensión de relacionar estímulos aplicados a individuos, con la sensación subjetiva que les producía y la novedad que suponía, introducir por primera vez, graduación o escalamiento en las sensaciones subjetivas y no un único atributo para denominarlas, como hasta el momento.

La extensión de estos modelos de escalamiento multidimensional hizo que el ámbito de aplicación saliera de la Psicología, para adentrarse en el campo del análisis de datos, de aplicación universal.

En la actualidad, el escalamiento multidimensional, puede considerarse como una familia muy amplia de procedimientos de análisis multivariado, aptos para tratar gran cantidad de tipos diferentes de datos de entrada (tablas de contingencia, matrices de proximidad, correlaciones, datos de perfil, etc.) y se ha incorporado en los programas de análisis estadístico disponibles hoy en día, manuales, etc.

El escalamiento multidimensional es una técnica de análisis multivariante que permite representar las proximidades entre un conjunto de objetos como distancias en un espacio de baja dimensionalidad (generalmente de 2 ó 3 dimensiones). Partiendo de una matriz de proximidades, nos proporciona un nuevo gráfico con unas nuevas coordenadas para los valores. Estas nuevas coordenadas, sufren transformaciones admisibles como rotación, traslación, escala y reflexión. Comentando brevemente la última de las transformaciones, por ser la menos conocida, cabría decir que la imagen especular de una configuración de puntos tampoco altera las posiciones relativas de los mismos. Aunque en el caso de unas distancias Euclídeas esto produce una representación difícilmente aceptable, podemos volver a aplicar una refexión a la misma y nos permitirá recuperar el mapa original. Dado que todas las transformaciones son admisibles, también lo es utilizar varias de ellas simultáneamente. Así, por ejemplo, rotar una configuración 45ºa la derecha, reflejarla horizontalmente y duplicar luego su tamaño, no altera las distancias relativas entre los objetos. Por último mencionar, que en la nueva localización, en el eje X, se coloca la distancia mayor de los objetos y la siguiente en el eje Y. Por tanto las combinaciones posteriores van colocándose en el espacio del plano. Así pues, las mayores diferencias encontradas en la matriz de proximidades quedan representadas en los ejes ortogonales.

De un modo más formal y general, podemos decir que el escalamiento multidimensional toma como entrada habitual una matriz cuadrada de proximidades, que podemos llamarle D, de tamaño nxn, donde n es el número de observaciones. Cada entrada d_{ij} de la matriz D, representa la proximidad entre las observaciones i y j. A partir de esta matriz, el análisis de escalamiento muldimensional, nos proporciona una matriz rectangular X, de tamaño nxm, donde n es el número de observaciones y m es el número de dimensiones (en nuestro caso $m = 2$). Cada valor x_{ia} corresponde a la coordenada del estímulo i en la dimensión a (en nuestro caso, por filas corresponderá la coordenada x e y del nuevo eje de coordenadas bidimensionales que creará el programa).

Hay una gran familia de procedimientos de análisis multidimensional, aplicables a múltiples tipos de datos y diferentes niveles de medida y que tienen distintos supuestos de partida y diversas finalidades. La distinción entre las principales variantes de escalamiento multidimensional, puede hacerse en función de unos cuantos factores, los más relevantes, serían el tipo de datos de entrada, número de matrices de proximidad empleado, modelo de escalamiento empleado...

El caso en el que los datos de entrada sean distancias, sólo haya una matriz de proximidades (distancias) y ésta sea simétrica, constituye lo que se conoce como el escalamiento multidimensional clásico o CMDS, técnica que vamos a utilizar en el estudio de simulación, dado que partimos de la matriz D de distancias de Mahalanobis, entre vectores de funciones LISA individuales.

6.5. Clasificaciones de rasgos en presencia de ruido. Ejemplos simulados

Consideramos, al igual que en el Capítulo anterior, una casuística suficientemente general para cubrir una amplia gama de posibles casos reales. Así, hemos considerado tres posibilidades para el ruido 50, 100 y 150 puntos, dos intensidades para el rasgo 50 y 150 puntos y tres formas de rasgos, cuadrado, forma L y circular. Hay que tener en cuenta las dimensiones de las matrices de distancias calculadas, que en el menor de los casos simulados es de $100x100$. Cada una de las entradas necesita unos cálculos, que llevan el producto de tres matrices, siendo una de ellas la inversa de la matriz de varianzas-covarianzas, de dimensión "r", que es la cantidad de distancias tomadas en el cálculo de los vectores de las funciones LISA. Para todos los ejemplos el vector de funciones LISA tendrá 25 componentes, por tanto $r = 25$, correspondientes a las distancias tomadas entre 0.005 y 0.5. El tiempo computacional es alto pero compensa con la calidad de los resultados.

Para cada combinación de casos analizados, calculamos la matriz de distancias D, sobre la que aplicaremos Cluster jerárquico y Escalamiento multidimensional. La representación gráfica del Cluster jerárquico se basa en los dendrogramas, y la del Escalamiento multidimensional en ejes coordenados ortogonales donde las nuevas coordenadas de las observaciones son diferentes de las originales. Tanto en un método como en el otro, obtenemos agrupaciones de diferencias de vectores de funciones LISA individuales. Tales agrupaciones las identificamos con los valores originales de las localizaciones espaciales de puntos y devolvemos la imagen final de rasgo y ruido clasificado.

Para todos y cada uno de los ejemplos considerados, que por su interés se vayan a incluir en este trabajo, se presentarán dos tipos de Figuras, la primera

referente al resultado del Cluster jerárquico y la segunda referente al resultado del Escalamiento multidimensional.

El primer ejemplo que vamos a analizar tiene 50 puntos de ruido, aleatoriamente distribuidos en el cuadrado unidad y 50 puntos de rasgo, aleatoriamente distribuidos en la región cuadrada de vértices (0.4,0.2), (0.6,0.2), (0.4,0.6), (0.6,0.6). Observando la primera de las imágenes tanto en la Figura (6.1), como en la Figura (6.2), parece difícil que se pueda clasificar en dos grupos, dada la poca diferencia de cantidad de puntos que hay en uno y otro, de hecho ésta es una clara dificultad para el método propuesto en el Capítulo 5. Se calcula la matriz de las distancias de Mahalanobis, entre las funciones LISA individuales de todos los puntos del patrón puntual y aplicamos los dos algoritmos de clasificación expuestos anteriormente, Cluster jerárquico y Escalamiento multidimensional.

Para el primero de los métodos, el Cluster jerárquico, Figura (6.1), el porcentaje de puntos clasificados como rasgo de los puntos de rasgo originales es del 90 %. Es un porcentaje muy alto, atendiendo a la intensidad de puntos del rasgo y ruido, dado que a simple vista, como se ha comentado, es difícil de distinguir. Seguidamente se presenta el dendrograma del Cluster jerárquico. A grandes rasgos, podemos observar que se han clasificado los puntos en dos grandes grupos. Las reducidas dimensiones de la imagen hace imposible la distinción del número de cada observación, por tanto la distinción visual de los individuos de cada grupo. Esta situación se repetirá en todos los ejemplos que vamos a analizar. No es esto lo importante. Mostramos esta imagen para indicar, que en un análisis clásico como es el que se está utilizando, ya hay muestra evidente de dos grupos diferenciados, a partir de la matriz de distancias entre funciones LISA individuales.

Como se ha comentado en la teoría de este mismo capítulo, queda a decisión del investigador el corte de las ramas del dendrograma y en nuestro caso hemos optado por cortar por los dos grupos que de forma clara se manifiestan. Así, las siguientes dos imágenes son los dos grupos de funciones LISA individuales por separado, que hemos llamado, rama del rasgo y rama del ruido.

Rescatando los datos relacionados con cada una de las funciones LISA individuales correspondientes a cada una de las ramas, obtenemos las siguientes dos imágenes, la del rasgo y la del ruido clasificado.

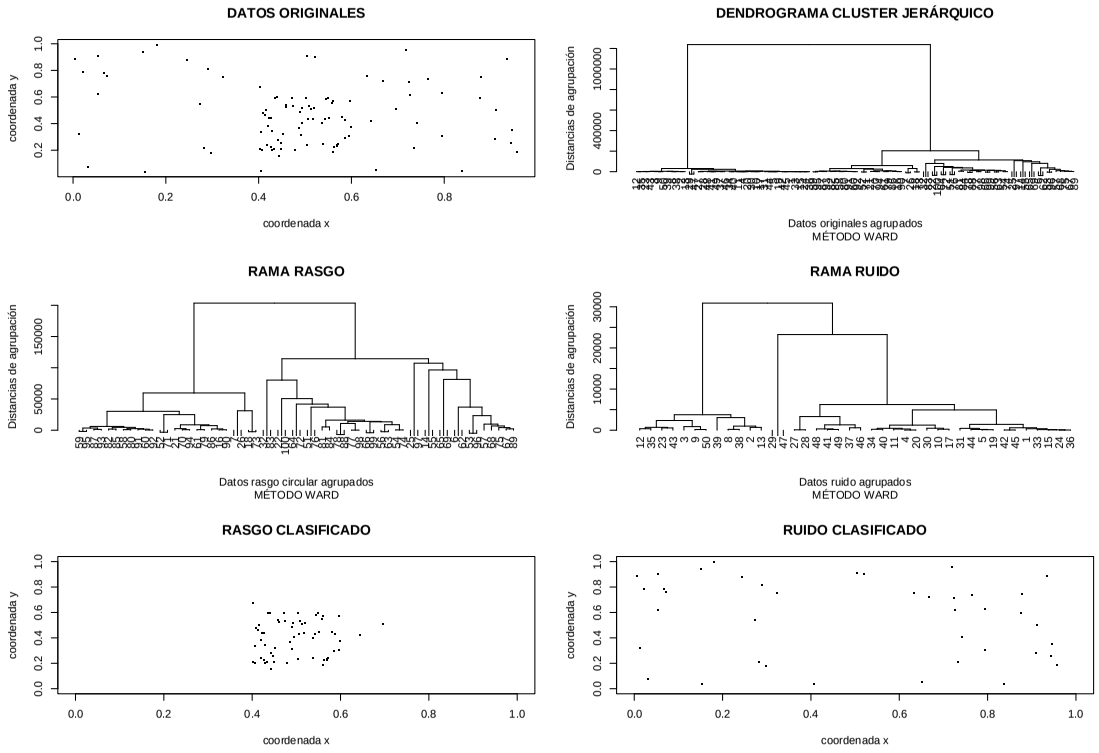

Figura 6.1: Clasificación de un rasgo cuadrado de 50 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

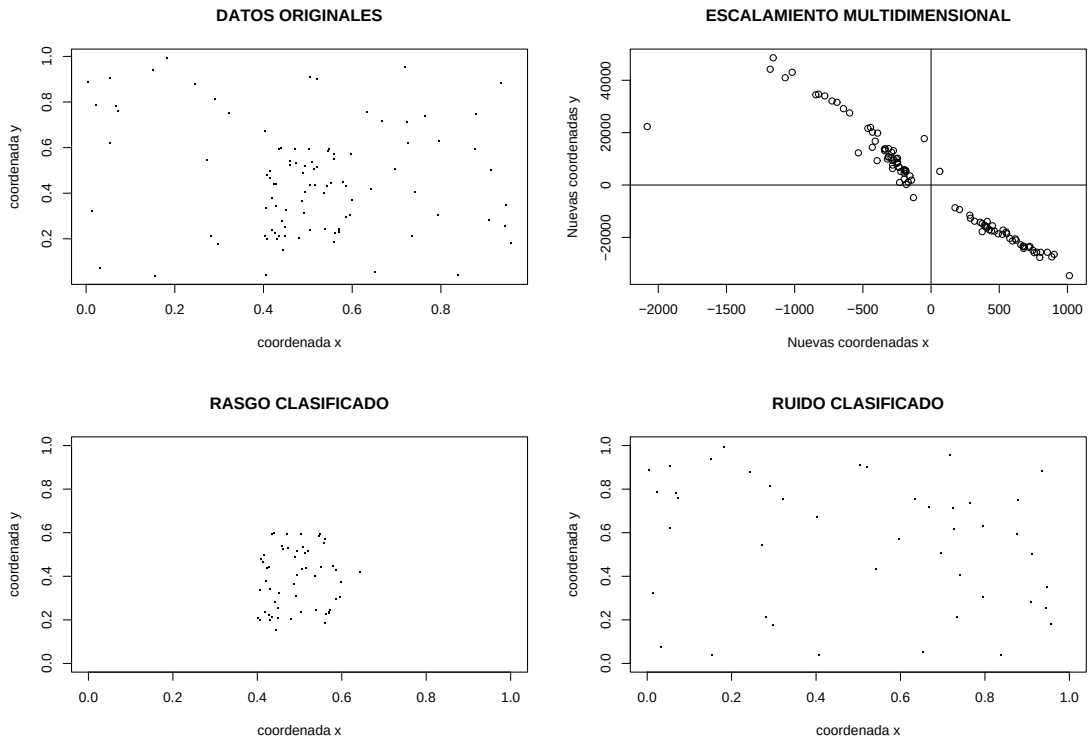

Figura 6.2: Clasificación de un rasgo cuadrado de 50 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales , (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

En este mismo ejemplo, pero utilizando el método del Escalamiento multidimensional, Figura (6.2), observamos resultados igualmente satisfactorios a un 90 %. La primera imagen es la misma que en la Figura (6.1) y corresponde a los datos originales. La segunda de las imágenes muestra las nuevas coordenadas, que el Escalamiento multidimensional proporciona de los datos con los que estamos trabajando. Tal y como hemos comentado en la fase de teoría, las mayores diferencias entre los individuos se sitúan en los ejes coordenados. Si ocurre que hay dos individuos con una gran diferencia (y es muy superior al resto de diferencias entre los individuos), ésta se hace patente de forma tan absoluta, que parece que todas las nuevas localizaciones, a excepción de esas dos, queden agrupadas en una zona extremadamente pequeña, lo que impide su comprensión de manera visual. Este fenómeno se observará en ejemplos posteriores. Pero si podemos eliminar esos dos o más individuos y volvemos a representar, podremos visualizar mucho mejor el grueso del conjunto de puntos, ya que la escala se ajusta a las nuevas dimensiones. Algo así ha ocurrido en este ejemplo, que al eliminar dos individuos cuya diferencia distorsionaba mucho las localizaciones nuevas para el resto de individuos, obtenemos la imagen que se muestra, donde claramente hay dos grupos de analogías, aquellas que se encuentran en el segundo y el cuarto cuadrante de los ejes coordenados.

Aislando las funciones LISA individuales correspondientes a estos dos grupos, identificando con los datos originales y volviendo a representar los dos grupos clasificados, obtenemos el resultado que se observa en las últimas dos imágenes.

Las siguientes dos Figuras (6.3) y (6.4), corresponden a un ejemplo de una simulación con 100 puntos de rasgo en la misma región cuadrada, sobre 100 puntos de ruido en el cuadrado unidad. La primera de las imágenes corresponde a los datos originales.

Con el primer estudio de Cluster jeráquico, Figura (6.3) y observando el dendrograma de la segunda imagen, puede parecer que hay dos grandes grupos a un primer nivel de corte. Un análisis más detallado, nos dirá que no es en el primer nivel donde hay que cortar, sino en el segundo, a una altura de 10^7. De esta forma, en las siguientes dos imágenes y sobre todo en las dos últimas, queda evidente el acierto del corte elegido. El porcentaje de buena clasificación, es del 100 %.

Para el estudio de Escalamiento multidimensional, Figura (6.4), la situación es similar a la que ocurre en el primer caso. Al eliminar aquellos individuos, cuya

diferencia distorsionaba enormemente las localizaciones del resto de individuos, se obtiene el mapa que se presenta en la segunda imagen. La buena clasificación del rasgo original obtenida es, al igual que en el Cluster jeráquico, de un 100 %.

Siguiendo de manera creciente, en la cantidad de puntos de rasgo y de ruido, las siguientes dos Figuras (6.5) y (6.6), corresponden a un ejemplo de una simulación con 100 puntos de rasgo en la misma región cuadrada, sobre 150 puntos de ruido en el cuadrado unidad. La primera de las imágenes corresponde a los datos originales.

El primer estudio, Cluster jeráquico, Figura (6.5), muestra un dendrograma en la segunda imagen, que de la misma manera que en los casos anteriores, puede parecer que haya dos grandes grupos a un primer nivel de corte. Un análisis más detallado, vuelve a evidenciar que no es en el primer nivel donde hay que cortar, sino en el segundo, a una altura de 10^7. Con este corte, el porcentaje de buena clasificación para los puntos del rasgo, es del 98 %.

Estudiando el Escalamiento multidimensional, Figura (6.6), la situación es similar a la que ocurre en los anteriores casos, al menos al considerar todos los individuos. Ocurre, que al eliminar algunos de ellos, cuya diferencia distorsionaba enormemente las localizaciones del resto de individuos, se vuelve a repetir la situación y vuelven a posicionarse dos en los extremos de los ejes, es decir, aparentemente no mejoramos el aspecto del gráfico. Esto no es inconveniente para la obtención numérica del resultado, como se muestra en las dos últimas imágenes. Si vamos eliminando puntos hasta conseguir una escala donde las localizaciones puedan verse, obtenemos la situación que se muestra en la Figura (6.7). La buena clasificación del rasgo original obtenida es de un 99 %.

El siguiente ejemplo, Figuras (6.8) y (6.9), corresponde al caso de una simulación con 150 puntos de rasgo en la misma región cuadrada, sobre 150 puntos de ruido en el cuadrado unidad. La primera de las imágenes, corresponde a los datos originales.

Para el análisis, Cluster jeráquico, Figura (6.8), se muestra un dendrograma a la segunda imagen, en el que hay que bajar hasta el cuarto nivel de corte, a una

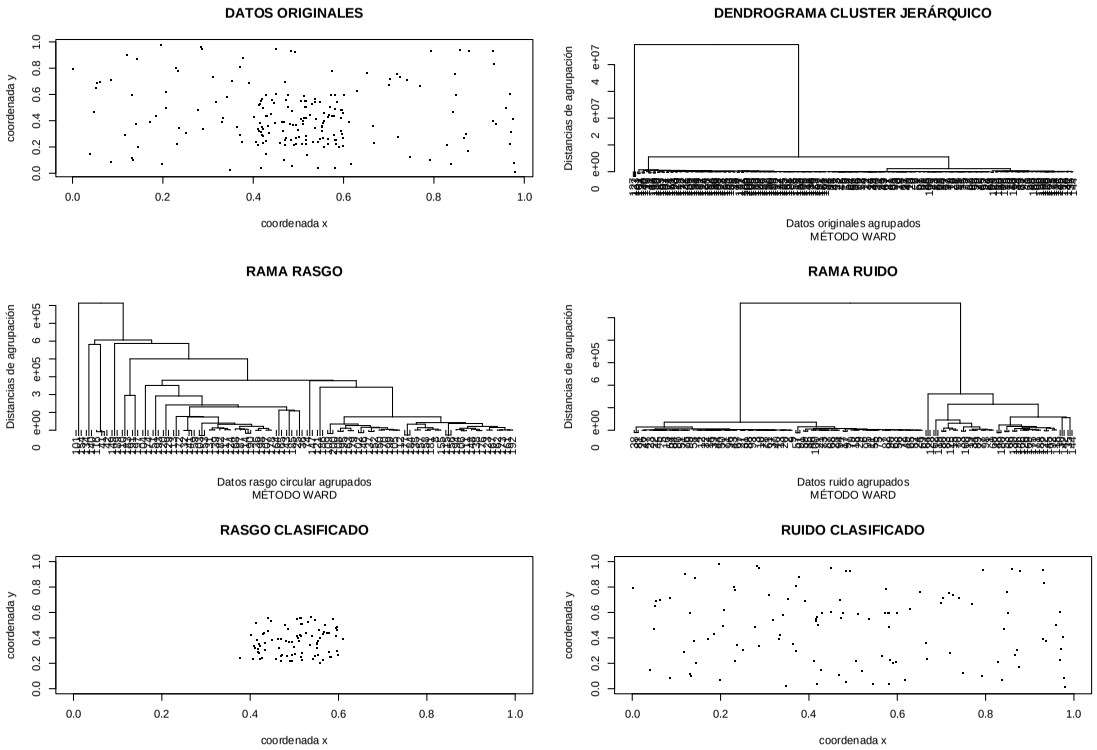

Figura 6.3: Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 100 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

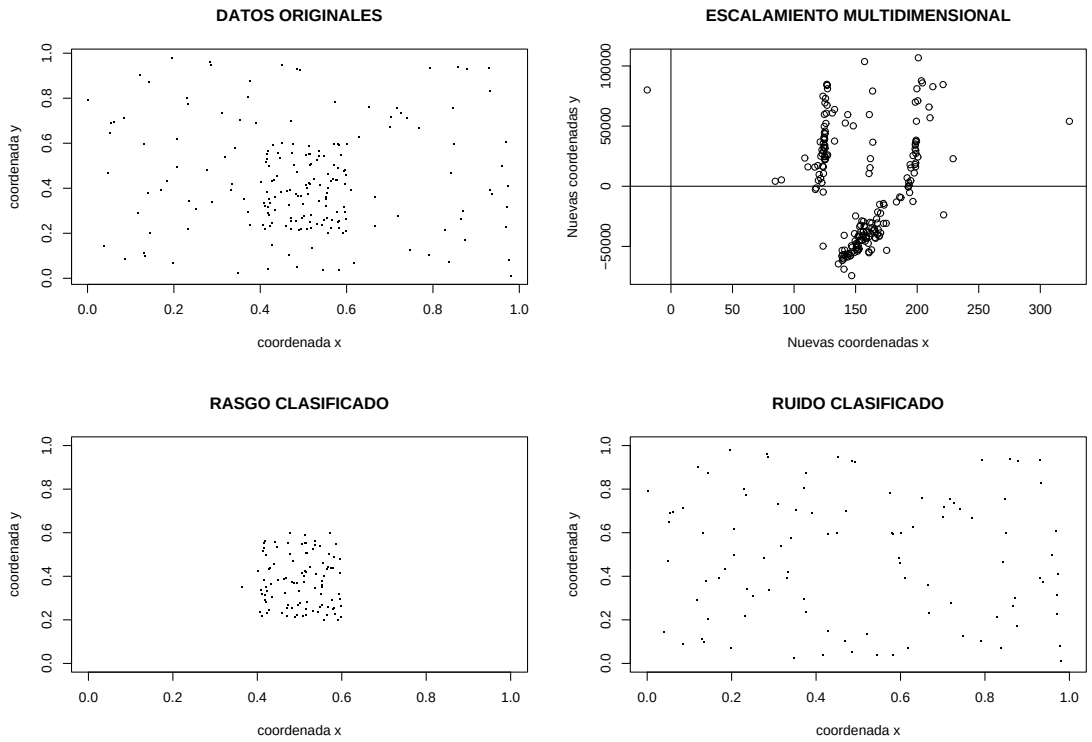

Figura 6.4: Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 100 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

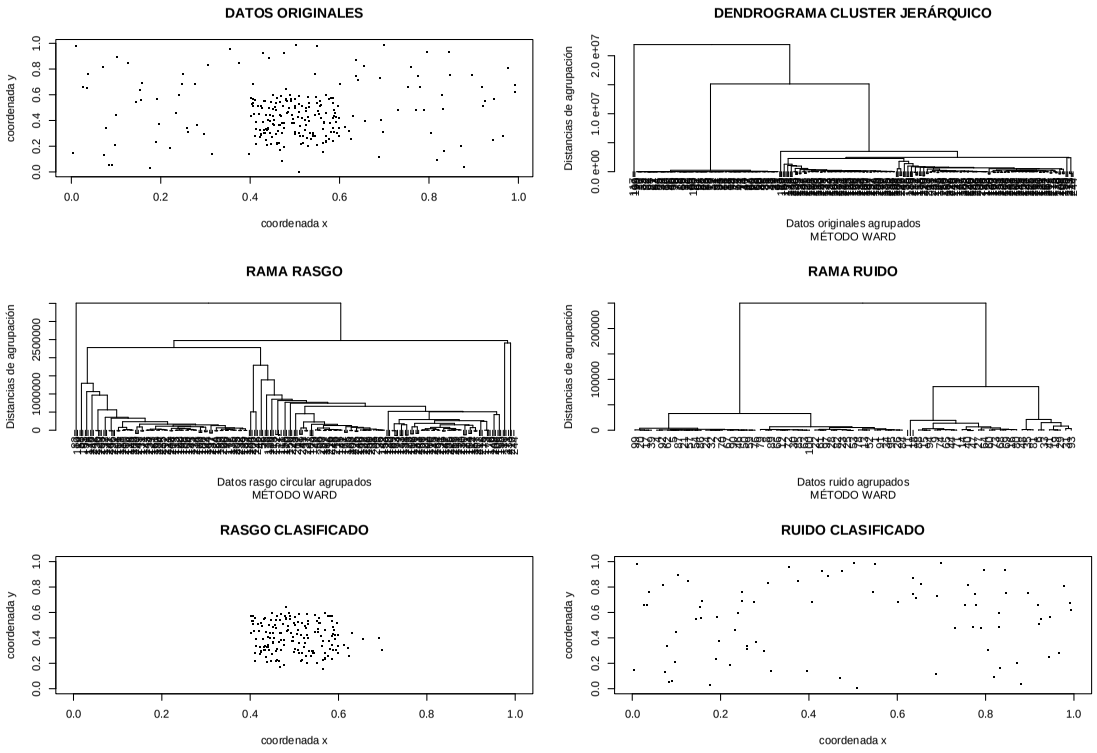

Figura 6.5: Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

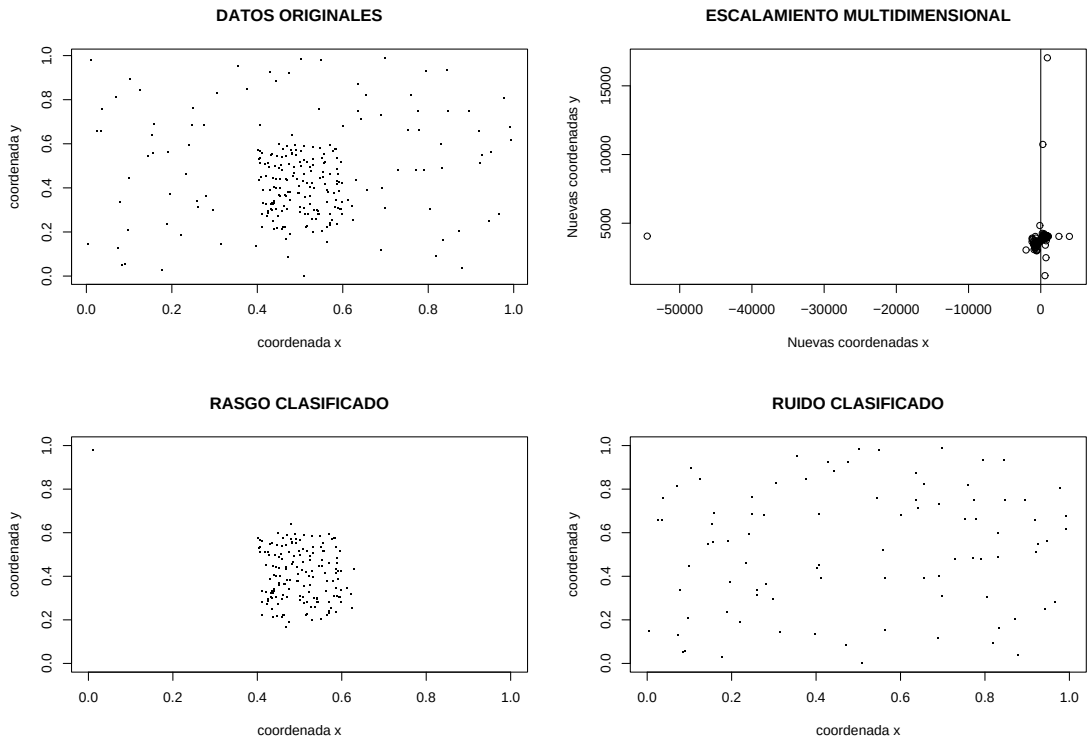

Figura 6.6: Clasificación de un rasgo cuadrado de 100 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

altura de $1.5\text{x}10^7$, para obtener los dos grupos de funciones LISA individuales buscados, los correspondientes a rasgo y ruido. Con este corte, el porcentaje de buena clasificación para los puntos del rasgo, es del 95 %.

La situación al utilizar el Escalamiento multidimensional, Figura (6.9), para la agrupación de distancias, es similar a la que ocurre en el caso anterior. Vuelve a suceder que al eliminar aquellos individuos, cuya diferencia distorsiona enormemente las localizaciones del resto de individuos, no se mejora la representación de la imagen. Esto vuelve a no ser inconveniente para la obtención numérica del resultado, como se muestra en las dos últimas imágenes. De la misma forma, con un seguimiento más exhaustivo de los puntos cuya diferencia es grande, obtenemos una mejor representación, Figura (6.10). La buena clasificación del rasgo original obtenida es de un 95 %.

Pasamos a un ejemplo en el cual el rasgo considerado es de forma circular. Las Figuras (6.11) y (6.12), corresponden a una simulación con 228 puntos de rasgo en una región circular, sobre 50 puntos de ruido en el cuadrado unidad. La primera de las imágenes, corresponde a los datos originales.

Realizamos el primer análisis, el Cluster jeráquico, Figura (6.11), en la que se muestra el dendrograma correspondiente. En esta ocasión hay que bajar hasta el segundo nivel de corte, a una altura de $6\text{x}10^7$, para obtener los dos grupos de funciones LISA individuales buscados, los correspondientes a rasgo y ruido. Con este corte, el porcentaje de buena clasificación para los puntos del rasgo, es del 96 %.

Para el estudio de Escalamiento multidimensional, Figura (6.12), obtenemos un rotundo 99 % de buena clasificación. Hemos eliminado algunos individuos, cuya diferencia distorsiona las localizaciones del resto de individuos y la representación de la imagen mejora poco. Eliminando más puntos, Figura (6.13), la representación mejora considerablemente.

El último caso que vamos a considerar en este estudio de simulación, Figuras (6.14), (6.15), (6.16) y (6.17), es un ejemplo en el cual hay dos rasgos. Uno de

forma circular con 164 puntos y uno cuadrado con 50 puntos, sobre 50 puntos de ruido en el cuadrado unidad. La primera de las imágenes, como es habitual, corresponde a los datos originales.

En el primero de los análisis, Cluster jeráquico, Figura (6.14), se muestra el dendrograma correspondiente en la segunda imagen. En un primer nivel de corte, a una altura de $3x10^7$, se obtienen dos grupos en principio claramente diferenciados. Al representar los dos grupos, como se muestra en las dos últimas imágenes, son el rasgo circular en un grupo y el rasgo cuadrado y ruido, en otro. Esta clasificación es lógica, dado que la intensidad del rasgo que ha quedado separado, es superior a la intensidad del segundo rasgo, el cuadrado y del ruido. Estos dos últimos tienen intensidades más parecidas, por tanto han quedado unidos en un primer nivel de clasificación.

Si realizamos un segundo corte, pero sólo de la segunda rama, la que habíamos considerado ruido, a una altura de $3.5x10^5$, Figura (6.15), el resultado es una nueva separación en dos grupos. En las dos últimas imágenes de esta Figura, podemos observar que uno de los grupos es parte del rasgo cuadrado, mientras que en el segundo grupo queda el resto de este rasgo, junto con el ruido. Continuando con esta técnica llegaríamos a aislar totalmente el rasgo del ruido. El procedimiento manual que vamos realizando acaba cuando se desee, pero ya intuimos, que dada una imagen real más compleja, realizar esta secuencia de actuaciones lleva asociado un gran porcentaje de aleatoriedad en el resultado final, según el investigador que va seccionando el dendrograma.

Para el estudio de Escalamiento multidimensional, Figura (6.16), se repite una situación parecida a casos anteriores. Hemos vuelto a eliminar algunos de los individuos, cuya diferencia distorsiona las localizaciones del resto de individuos y la representación de la imagen no mejora, como se muestra en la segunda imagen. Será en la Figura (6.18), donde al eliminarse la totalidad de puntos cuyas distancias son muy elevadas, la representación quede más clara. En esta ocasión tenemos un problema parecido al caso del cluster jerárquico, en el cual, con un procedimiento similar a los ejemplos anteriores, queda claramente diferenciado el rasgo circular, por tener mayor intensidad de puntos, del otro grupo que sería el rasgo cuadrado y el ruido.

Con la ayuda de la Figura (6.18), dado que la imagen queda muy confusa al estar todas las localizaciones muy concentradas y próximas al origen de coordenadas, se procede a encontrar el valor del eje de abcisas u ordenadas, por el que

separar los dos grupos y de forma que queden identificadas tanto la imagen de rasgos, como la de ruido, Figura (6.17). En esta ocasión, el valor de corte ha sido para x=300 y el resultado es fiable a un 80 %. Para el resto de ejemplos simulados anteriores, el valor de corte ha sido el origen de coordenadas, respecto del eje X o eje Y.

Se ha mostrado la evidencia de la eficacia de la utilización de las funciones LISA individuales, en el mismo contexto que el capítulo anterior, pero siguiendo un método diferente. Haciendo un ejercicio de reflexión a nivel de comparación entre los dos procedimientos propuestos, es interesante pararnos en algunos detalles.

Parece más exacto el procedimiento del capítulo en el que nos encontramos, dado que se actúa directamente sobre las funciones LISA, se calculan distancias entre ellas y se analiza la matriz de estas distancias por un método concreto, bien sea Cluster jerárquico o bien Escalamiento multidimensional. En el caso del capítulo anterior, al introducir el algoritmo SEM, da la impresión de pérdida de exactitud, dado que se introduce la aleatoriedad estocástica. En este sentido es posible esta reflexión y parece lógica esta observación.

Si nos fijamos en el tiempo computacional, la comparación se decanta por el método propuesto en el capítulo anterior, de forma taxativa. El coste de calcular las matrices de distancias de Mahalanobis (sin realizar Cluster jerárquico o Escalamiento multidimensional), es extraordinariamente superior a cualquiera de los cálculos necesarios para la clasificación completa, realizando distancias a la esperanza de una función LISA de un punto de ruido. Si bien es cierto, que para situaciones donde hay pocos puntos y poca diferencia de intensidad entre los puntos de ruido y un sólo rasgo, el método de este capítulo se muestra más rotundo a la hora de clasificar, proporcionando mejores resultados.

Esta situación cambia cuando es el caso de dos rasgos. Además del analizado anteriormente, se han analizado más casos y en todos ellos el problema se encuentra en separar adecuadamente los grupos que generarán el rasgo del ruido, en situaciones donde las intensidades son diferentes y alguno de los rasgos tiene intensidad parecida al ruido. En ambos métodos se presentan situaciones en las

que el investigador ha de decidir por dónde separar los grupos, situación que deja muy a criterio personal el resultado. Los ejemplos mostrados presentan buenas clasificaciones en último término, pero al coste computacional de tiempo dedicado a obtener la matriz D de distancias, que era alto, hay que añadir el tiempo en el cual se decide, dentro de cada método a seguir, las particularidades propias de cada uno.

Éste ha sido el motivo principal de no extender excesivamente el número de casos de la simulación y de elegir un caso real, en el cual, el número de observaciones fuera relativamente pequeño. En ningún caso se pueden plantear problemas reales como los abordados en el Capítulo 5, por la imposibilidad de manejar matrices de más de 2000 entradas por fila o columna, donde cada una de ellas tenía el producto de tres matrices, una de ellas inversa de dimensión 100x100.

6.6. Clasificaciones de rasgos en presencia de ruido. Imágenes reales

En la Comunidad Valenciana, dada la gran vegetación que tiene y el clima mediterráneo que soporta, sufre en muchas ocasiones incendios forestales. La localización espacial de éstos ocurridos durante el periodo 2000-2001 se muestra en la Figura (6.19) y da muestra de la gravedad del problema siendo muchos los esfuerzos desde la Generalitat Valenciana por investigar en prevención, acción y métodos paliativos a zonas devastadas. En la Figura (6.20) se muestra una imagen con efecto tridimensional, en la que se puede observar la función de intensidad de estas localizaciones. Ya se puede apreciar que coincide mayor intensidad con zonas geográficas conocidas, como el interior de Castellón de la Plana y especialmente la zona más oriental, por el límite entre la provincia de Valencia y Alicante, donde hay grandes formaciones montañosas.

En este sentido elegimos este ejemplo, para mostrar un caso real en el cual el método de las funciones LISA individuales nuevamente da respuesta. Se calcula la matriz de distancias D y sobre ella aplicaremos Cluster jerárquico tradicional y Escalamiento multidimensional. Las detecciones de rasgos ponen en evidencia las

localizaciones de grupos de incendios en entornos geográficamente similares (macizos o montañas) que comparten la misma vegetación y las mismas condiciones climatológicas.

En las Figuras (6.21) y (6.22), se muestran las clasificaciones de la matriz de distancias obtenida entre las funciones LISA individuales por los métodos que ya se han ido utilizando en la sección de ejemplos simulados. En la Figura (6.19), como ya hemos comentado, se muestra la totalidad de los focos de los incendios y ya se puede apreciar en ella que hay zonas con mayor concentración de localizaciones.

La Figura (6.21), donde se ha utilizado Cluster jerárquico, muestra la clasificación de rasgo, o zona de mayor concentración de focos de incendios, la costa este, correspondiente a la zona de la Marina Alta (Denia, Gandía etc) y parte del interior de estas comarcas, llegando al centro de la provincia de Alicante. También queda clasificado como zona de mayor incidencia de incendios, el interior de Castellón. Hay que hacer notar, que efectivamente, son zonas muy montañosas y por tanto la acumulación de incendios es mayor.

Del mismo modo, la Figura (6.22), muestra la clasificación utilizando Escalamiento multidimensional, del mismo banco de datos y en ella se puede apreciar, que la clasificación ha sido sensible para las mismas zonas comentadas en el caso de Cluster jerárquico, si bien ambas no son exactamente iguales. Esto es debido a la forma poco delimitada, de lo que estamos considerando en esta imagen como rasgo. En todos los ejemplos de la fase de simulación, los rasgos están muy definidos, con formas muy concretas e intensidades de puntos notablemente diferentes entre los rasgos y el ruido. Además, ambos métodos comparten la característica de necesitar al investigador para dar por definitiva la clasificación y en este sentido, incluir una rama más o menos del dendrograma o un valor de corte diferente en el Escalamiento multidimensional, hace variar el resultado. Finalmente, insistir en el hecho de la detección por parte de los dos métodos, Cluster jerárquico y Escalamiento multidimensional, de la zona de mayor incidencia de incendios, la ya comentada Marina Alta de Alicante, tal y como queda reflejada en los picos más altos de la función de intensidad.

En este sentido, el método propuesto clasifica este tipo de imágenes y se muestra válido y oportuno una vez más, detectando y separando zonas que, efectivamente, tienen comportamientos diferentes aunque a primera vista, pudiera

parecer que no. Las zonas detectadas, son consideradas de especial riesgo de incendios, dada su geografía y alta frecuencia de incidentes relacionados con el fuego. Por todo lo expuesto, consideramos buena, la clasificación mostrada.

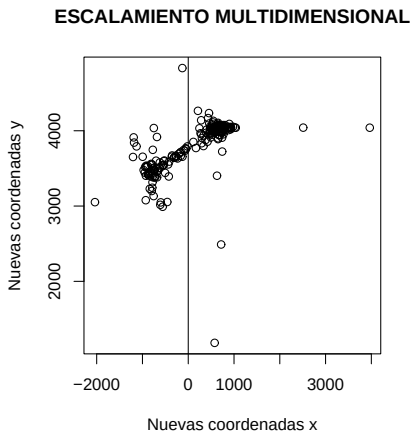

Figura 6.7: Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.6), donde se han eliminado los valores de gran distancia.

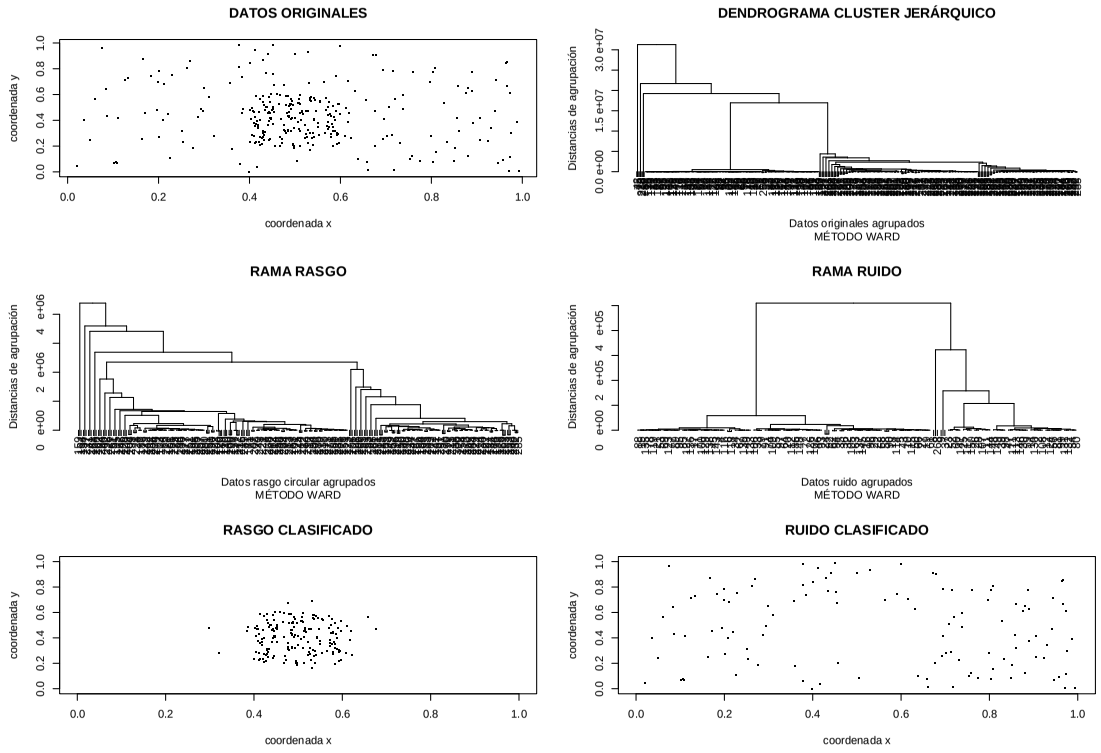

Figura 6.8: Clasificación de un rasgo cuadrado de 150 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

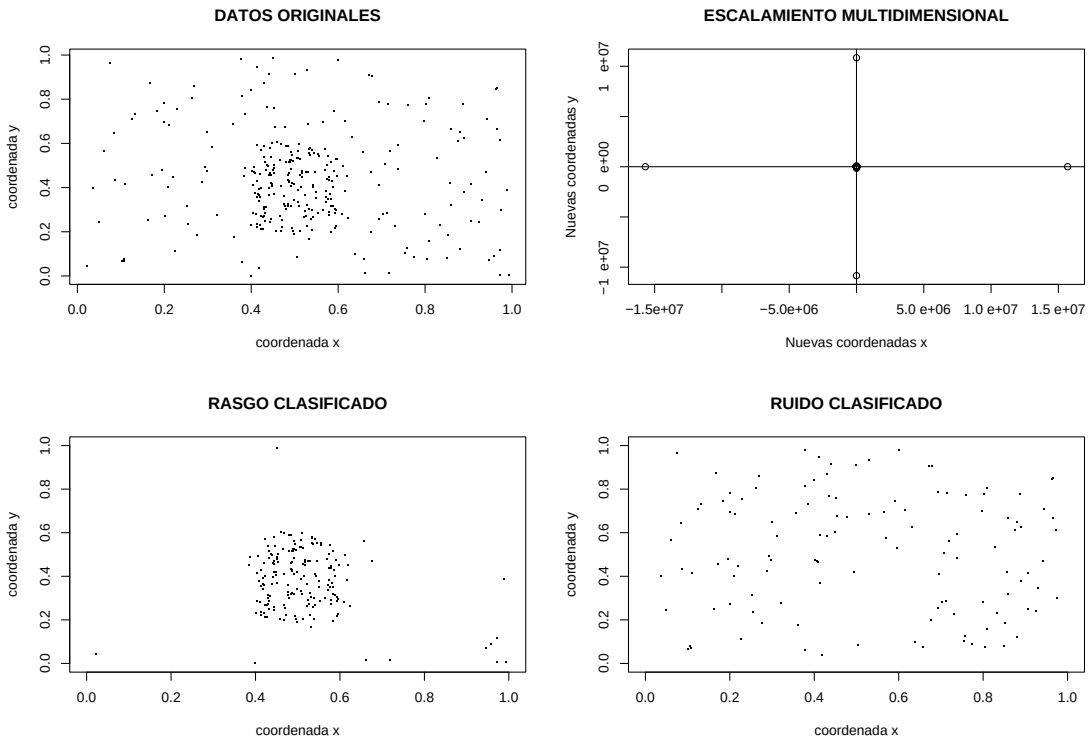

Figura 6.9: Clasificación de un rasgo cuadrado de 150 puntos sobre ruido de 150 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

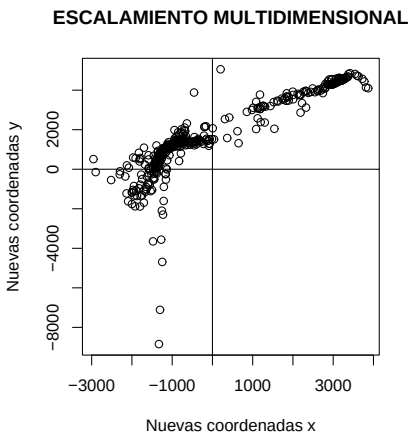

Figura 6.10: Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.9), donde se han eliminado los valores de gran distancia.

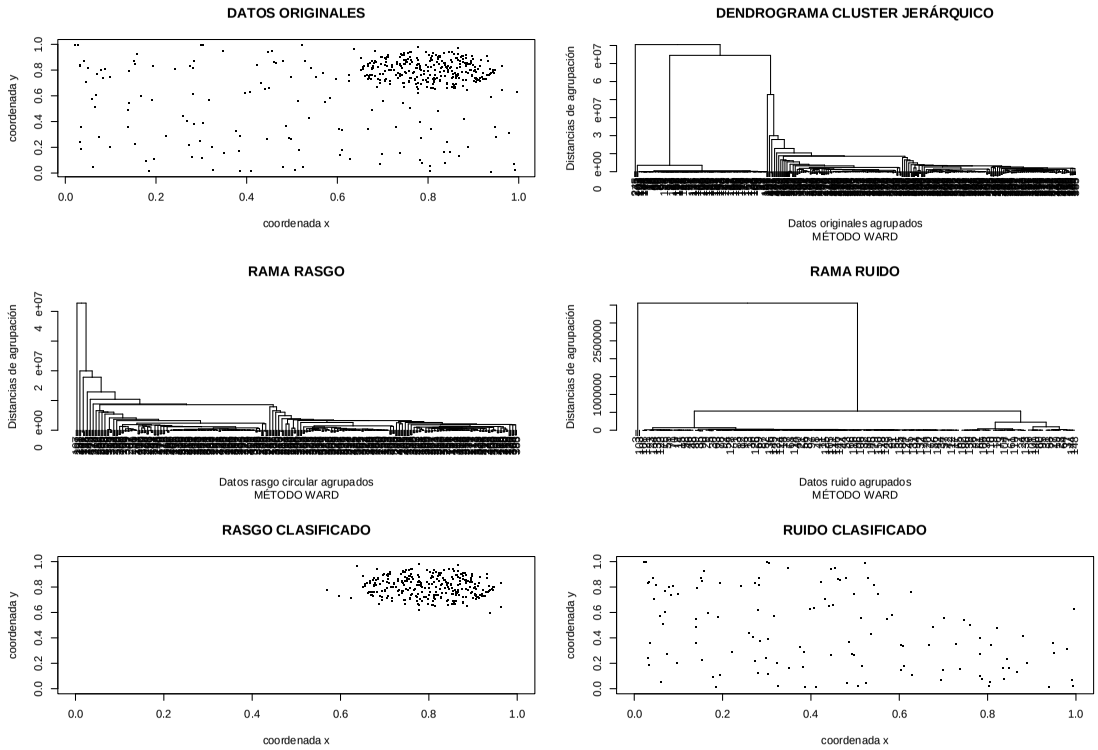

Figura 6.11: Clasificación de un rasgo circular de 228 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

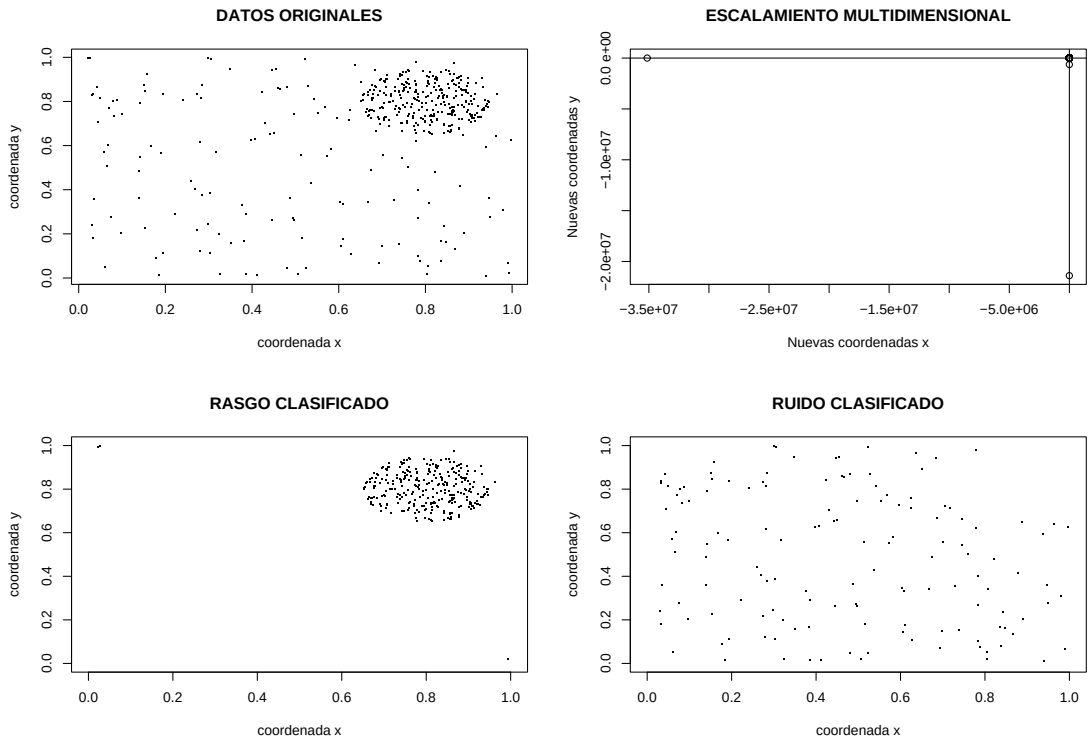

Figura 6.12: Clasificación de un rasgo circular de 228 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

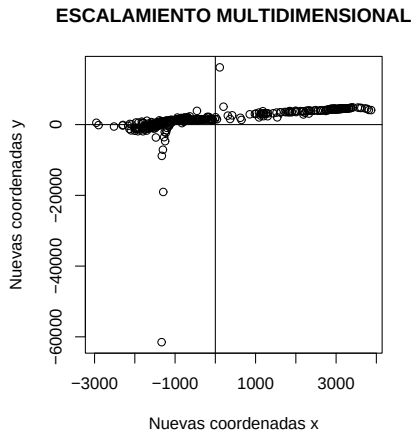

Figura 6.13: Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.12), donde se han eliminado los valores de gran distancia.

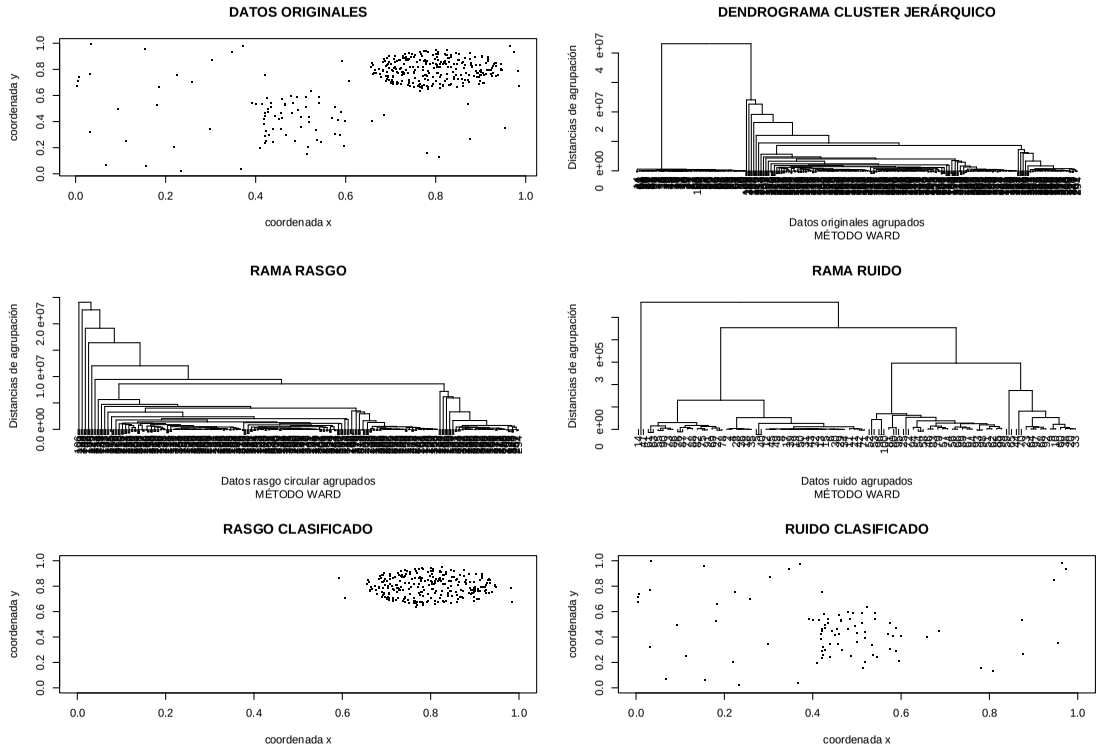

Figura 6.14: Clasificación de dos rasgos, uno circular de 164 puntos, uno cuadrado de 50 puntos sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

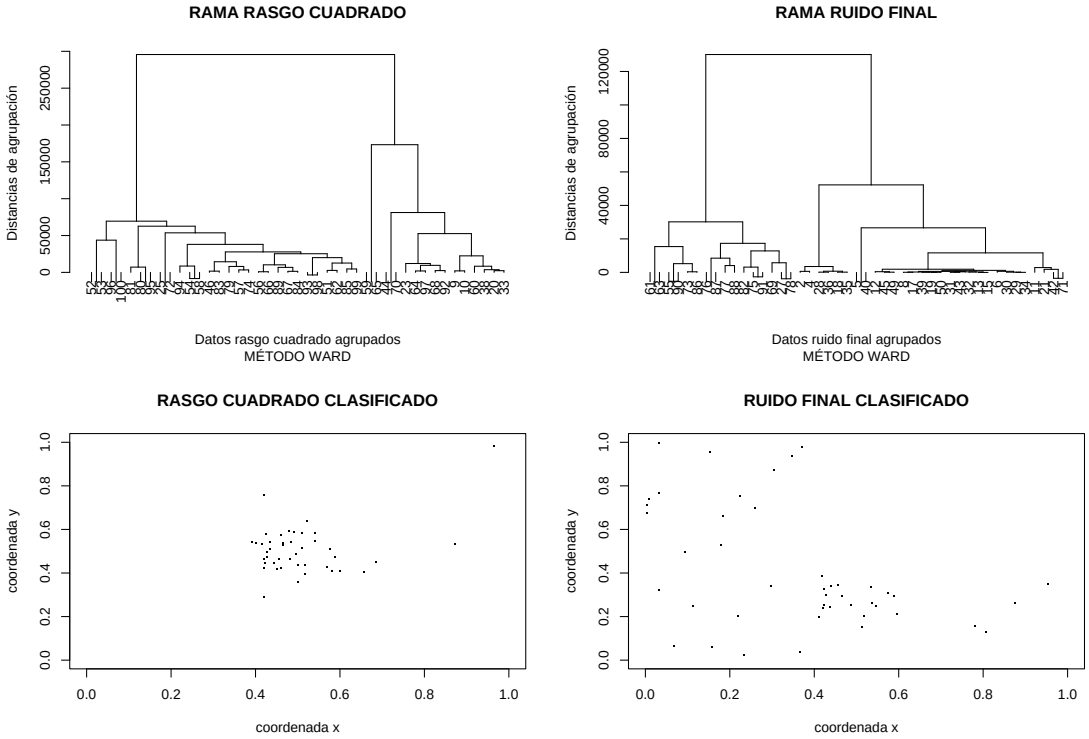

Figura 6.15: Clasificación de un rasgo cuadrado de 50 puntos para la rama de ruido de la Figura (6.14), sobre ruido de 50 puntos en el cuadrado unidad, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

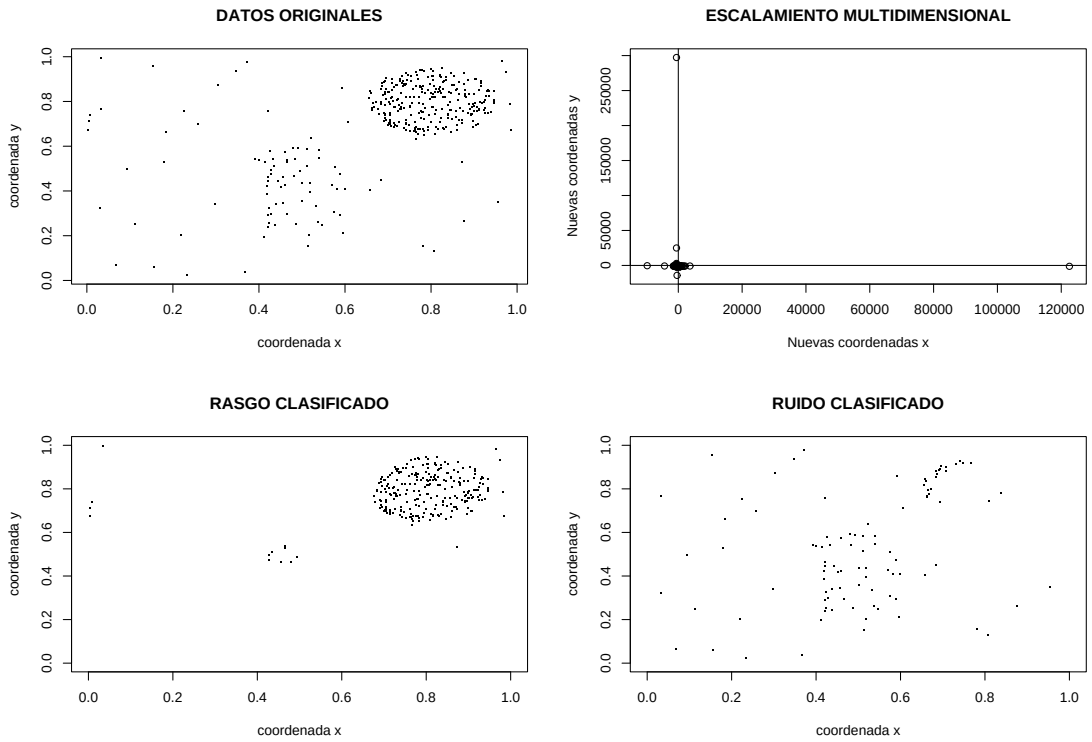

Figura 6.16: Clasificación de dos rasgos, uno circular de 164 puntos y uno cuadrado de 50 puntos, sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del rasgo clasificado, (derecha) representación del ruido clasificado.

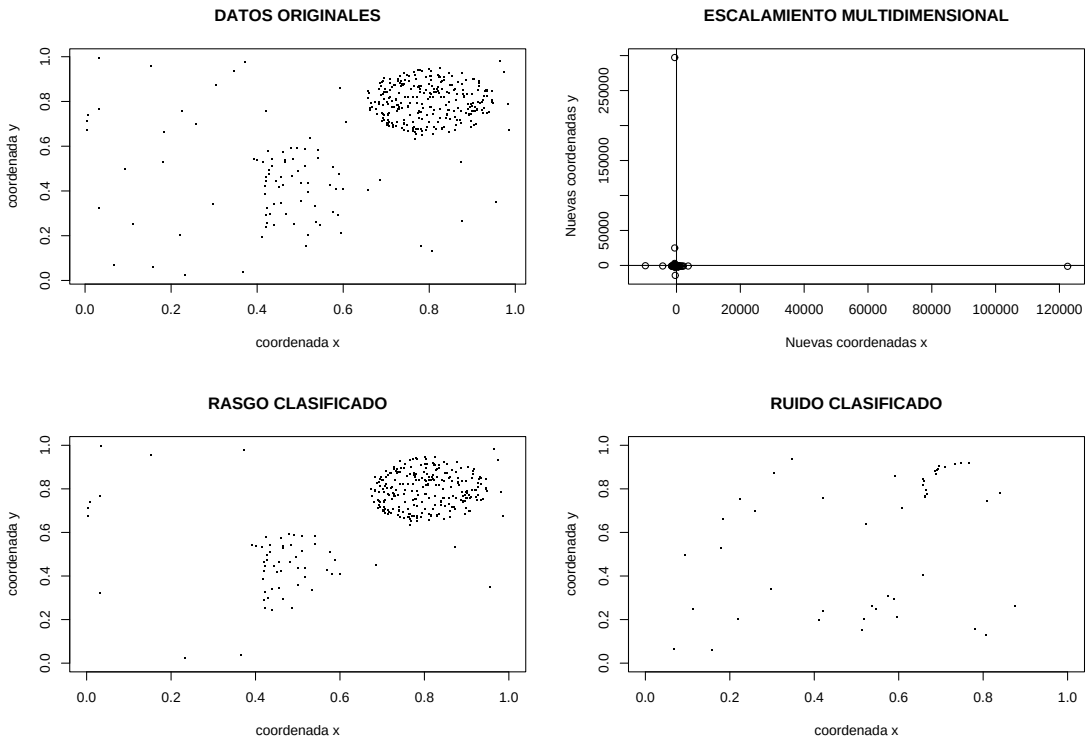

Figura 6.17: Clasificación de dos rasgos, uno circular de 164 puntos y uno cuadrado de 50 puntos, sobre ruido de 50 puntos en el cuadrado unidad, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del rasgo clasificado (II), (derecha) representación del ruido clasificado (II).

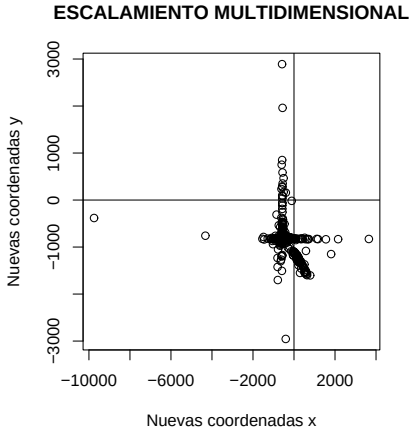

Figura 6.18: Representación de ejes de coordenadas nuevos, proporcionados por CMDS, correspondiente a la Figura (6.16), donde se han eliminado los valores de gran distancia.

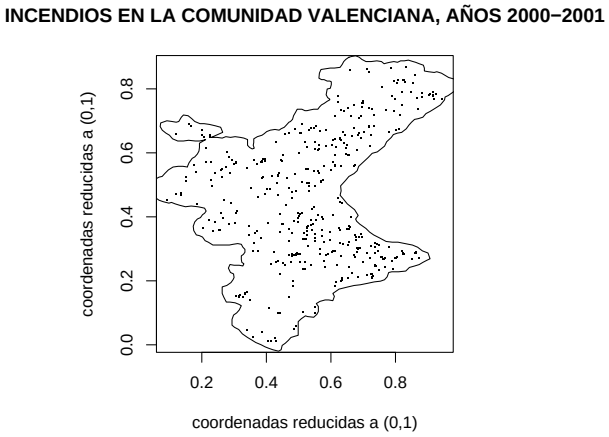

Figura 6.19: Localizaciones espaciales reducidas a (0,1), de focos de incendios durante los años 2000-2001, en la Comunidad Valenciana (España).

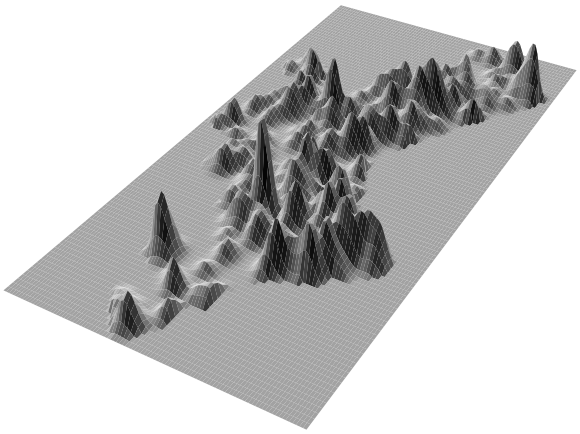

Figura 6.20: Función de intensidad de focos de incendios durante los años 2000-2001, en la Comunidad Valenciana (España).

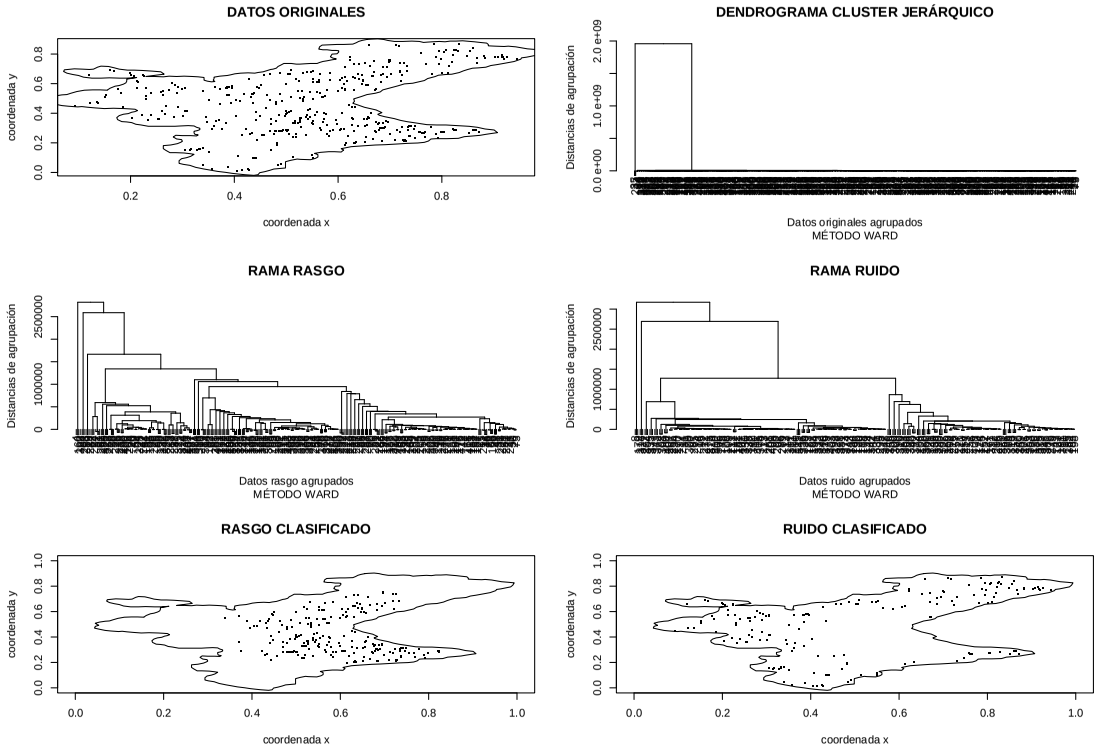

Figura 6.21: Clasificación de los focos de los incendios de la Comunidad Valenciana (España), durante los años 2000-2001, utilizando Cluster jerárquico. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de Dendrograma. *Segunda fila:* (izquierda) representación de rama de Dendrograma correspondiente a rasgo, (derecha) representación de rama de Dendrograma correspondiente a ruido. *Tercera fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

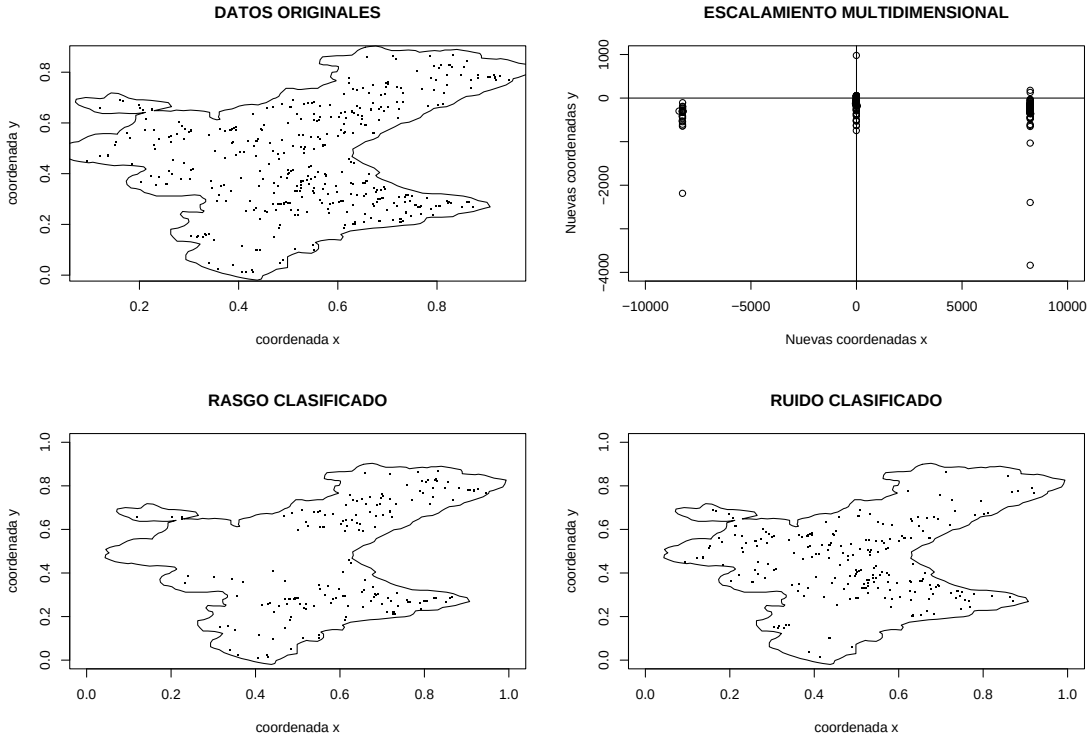

Figura 6.22: Clasificación de los focos de los incendios de la Comunidad Valenciana (España), durante los años 2000-2001, utilizando Escalamiento multidimensional. *Primera fila:* (izquierda) representación de datos originales, (derecha) representación de ejes de coordenadas nuevos, proporcionados por CMDS. *Segunda fila:* (izquierda) representación del ruido clasificado, (derecha) representación del rasgo clasificado.

Capítulo 7

Conclusiones y futuras investigaciones

La utilización de las imágenes digitalizadas, se está convirtiendo en una herramienta indispensable en cualquier ámbito de la ciencia. Continuamente hay avances respecto a la calidad con la que se obtienen y se transmiten y a menudo somos espectadores de aparatos nuevos que incorporan sensores de imágenes que tienen utilidades, hasta hace pocos años, inimaginables.

Cómo tratar estas imágenes de un modo científico de forma que sea una herramienta útil para mejorar la tarea de otras investigaciones, es un campo en el que actualmente se trabaja. Parece lógico pensar que en una imagen encontraremos rasgos candidatos a ser el motivo de estudio y que por su interés particular, se necesiten aislar.

La teoría de procesos puntuales, parece ser el marco de referencia idóneo en este sentido. Podemos considerar la imagen como un conjunto de puntos (píxeles) y por tanto estamos en la situación de un patrón puntual. A partir de ese momento, cualquier concepto estudiado en procesos puntuales, podrá ser aplicado a las imágenes digitalizadas.

El problema que el presente trabajo ha intentado abordar es aislar zonas de estas imágenes, que un motivo concreto hizo que tuvieran mayor concentración de puntos, del resto de la imagen. La motivación puede ser múltiple y se ha expuesto en numerosas ocasiones en el presente texto, pero la realidad es que hay

pocas referencias de trabajos que lo hayan abordado. Cierto es que McLachlan & Basford, 1988 [71]; Bensmail, Celeux, Raftery & Robert, 1994 [10] dan una solución novedosa y original en este sentido, pero de eso hace apenas 15 años. Se propone una técnica, basada en distancias al K-vecino más próximo, que mediante el algoritmo EM, clasifica los puntos de una región, en zonas de rasgo y ruido.

Por otra parte, tomamos las funciones que ofrecen información de manera global del estado de agregación de un proceso puntual. En Cressie & Brant, 1997 [27]; 2001 [28], se utiliza esta metodología para detectar grupos de puntos del patrón que previamente estaban identificados y se establecen relaciones entre ellos. La metodología que hemos propuesto se basa en la construcción de funciones locales que recogen características de segundo orden y en este sentido susceptibles a aportar información sobre la formación de rasgos, las llamadas funciones LISA.

La novedad radica en utilizar estas funciones LISA con caracter individual, para detectar agrupaciones de puntos en una imagen, de forma que lo que se revela realmente son agrupaciones de funciones LISA y al hacer corresponder estas funciones a los datos originales, éstos han quedado clasificados. Por tanto se pone de manifiesto la estrecha y absoluta relación entre las funciones utilizadas y el concepto de agregación de un proceso puntual y todo esto desde un punto de vista totalmente nuevo, un punto de vista individual para cada punto del patrón.

Otra novedad introducida es el algoritmo SEM. Dada la ausencia de distribución de probabilidad para las distancias entre las funciones LISA y dada la necesidad de tener valores de la función de densidad, se ha recurrido a estimarlos, mediante un Stochastic-step, entre el E-step y el M-estep del algoritmo EM.

Los resultados obtenidos son potentes y satisfactorios y en este sentido, la aportación del presente trabajo va dirigida a reducir las condiciones iniciales cuando se aborda un problema de detección de rasgos en un patrón puntual. Utilizamos únicamente las localizaciones espaciales de los puntos.

Como conclusión al trabajo presentado y atendiendo a criterios de rapidez computacional, simplicidad de algoritmos y resultados obtenidos y después de revisar los estudios existentes hasta el momento en este campo, la metodología propuesta en los Capítulos 5 y 6 sería, bajo nuestro criterio, la óptima para abordar un problema de estas características. La claridad con la que las funciones LISA individuales responden y abordan la agrupación en un proceso puntual,

las hace fundamentales en un estudio de clasificación y detección de rasgos en presencia de ruido.

Hay mucha tarea por realizar en este campo. A medida que hemos ido profundizando en el tema, van surgiendo ideas de cambios y de extensiones de conceptos a situaciones de mayor dimension. Considerar otras funciones de segundo orden que sustituyan a las funciones LISA, como la K-función, la L-función u otras, supondría el apasionante reto de comenzar de cero nuevamente y rehacer los pasos sobre los que hemos caminado, pero con una estructura paralela. Comparar estos nuevos métodos con los desarrollados en este trabajo, encontrando ventajas e inconvenientes y buscar aplicaciones concretas donde cada uno se manifieste mejor, es una línea de investigación que promete mucha dedicación y trabajo.

También se intuye larga tarea en el hecho de incorporar la dimensión del tiempo a cada uno de los individuos de un patrón puntual, extendiendo por tanto, la situación al caso espacio-temporal, para resolver el problema de detección de rasgos en presencia de ruido en su evolución en un periodo de tiempo. Ésta es otra de las líneas de investigación que se ha pensado seguir, por su interés y atractivo en campos como la medicina, botánica, epidemiología, geografía etc.

En el mismo sentido, pero en otra línea de procedimiento, tendríamos la posibilidad de utilizar componentes principales. El análisis de componentes principales puede encuadrarse dentro del conjunto de técnicas multivariantes conocidas como métodos factoriales. Se pretende sintetizar un gran conjunto de datos, crear estructuras de interdependencia entre variables cuantitativas para crear unas nuevas variables que sean función lineal de las originales y de las que podamos hacer una representación gráfica. El objetivo del análisis de componentes principales será reducir la dimensión de un conjunto de p variables a un conjunto de m variables, $p > m$, para mejorar la interpretación de los datos. Las nuevas variables, las componentes principales, determinan lo esencial de las variables originales que además tienen unas propiedades interesantes:

1. Son ortogonales (cada componente representa una dirección del espacio de las variables originales).
2. Cada componente está incorrelada con la anterior.
3. La primera componente es la que más varianza contiene y la j-ésima tiene más varianza que la $(j-1)$-ésima...

Las componentes principales son las combinaciones lineales de las variables que hacen máxima su varianza y esto se consigue obteniendo los autovalores y los autovectores de la matriz de correlaciones o distancias entre las observaciones, en nuestro caso. La gran diferencia está ahora en que las distancias entre los individuos no son distancias entre las localizaciones espaciales, sino entre funciones LISA individuales. En este sentido, hay estudios al respecto calculando componentes principales entre datos funcionales, detectando diferentes tipos de patrones puntuales, cuando éstos se encuentran en un proceso múltiple de patrones puntuales (Illian, 2004 [54]). A partir de la matriz D, calculada en el Capítulo 6, aplicaríamos el método de componentes principales, como una alteración de la utilización de Cluster jerárquico o Escalamiento multidimensional.

Por último y para finalizar con las futuras investigaciones, en el Capítulo 6, se expone la distribución teórica de la variable $Y_i - Y_l$, que como se ha comentado ampliamente, es la diferencia de vectores cuyas componentes son las funciones LISA individuales, correspondientes a los individuos del patrón puntual. De esta variable, conocemos y hemos utilizado en los cálculos la esperanza, la varianza y la covarianza.

Como se ha probado sobradamente en el Capítulo 5, el comportamiento de las funciones LISA individuales de rasgo y de ruido son diferentes. Este hecho es independiente del número de rasgos, de la forma o la intensidad de puntos que tengan, siempre que no sea tan débil esta intensidad que se confunda con la de puntos del ruido.

En este sentido, una nueva vía de investigación pasa por la construcción de un test de contraste que modelice el comportamiento de esta variable, bajo aleatoriedad. Para ello necesitamos que estas diferencias se ajusten a un modelo lineal, así podríamos determinar una hipótesis nula, si el individuo pertenece a ruido y la alternativa si no pertenece. Cómo construir el estadístico que se distribuyera χ^2 y permitiera el contraste, parece no encerrar gran dificultad. Intuimos la dificultad en asegurar no tanto que las diferencias de los vectores de funciones LISA individuales de ruido se ajusten a un modelo lineal, que gráficamente no se cumple dado que en una banda de confianza tienen trazados arbitrarios, sino en una transformación de ellas, bien sea el logaritmo, el valor absoluto..., alguna que lo cumpliera. El objetivo sería contrastar un vector cualquiera de un punto de un patrón puntual con la hipótesis construida. El test clasificaría este vector en ruido, si cumple la hipótesis nula, o rasgo en caso contrario. Identificando el

vector con el punto del cual se ha construido, tendríamos la clasificación en rasgo y ruido de nuestro proceso puntual.

Mucho es pues, el trabajo que queda pendiente y esperamos en el futuro poder ir dando respuestas a todas y cada una de las preguntas planteadas.

Capítulo 8

APÉNDICE: Códigos Splus, R y Fortran

En este último capítulo, se presentan los códigos que se han utilizado en la elaboración de este trabajo. El objetivo es mostrar la parte que puede ser más oculta en la redacción de esta memoria, pero sin embargo, la parte fundamental sobre la que se sostienen todas las consideraciones de los resultados que hemos obtenido y analizado.

Los cálculos relativos a estadística espacial han sido realizados mediante R, un software gratuito y muy apropiado para la realización de cálculos estadísticos. Se ha utilizado la librería Splancs, desarrollado por Barry Rowlingson y Peter Diggle, y sostenida por Roger Bivand para el entorno R. Todo este software puede obtenerse en la página http://www.R-project.org/.

Se presentan códigos para la elaboración de:

1. Simulación de patrones puntuales.
2. Cálculo de las distancia al K-ésimo vecino más próximo.
3. Cálculo de la distancia de grid al K-ésimo vecino más próximo.
4. Cálculo de la densidad producto.
5. Cálculo de las funciones LISA

6. Cálculo de NNCLEAN1, para el cálculo de las variables que clasifiquen en rasgo y ruido las localizaciones espaciales.

7. Código para simular el ruido y rasgo y obtener la clasificación.

8. Código para calcular la densidad producto y funciones LISA de datos simulados.

9. Código para cluster jerárquico y cluster basado en el algoritmo EM de datos simulados.

10. Código para representar funciones LISA e histogramas de las distancias.

11. Código para representar funciones LISA de diferentes rasgos por separado.

12. Código para generar rasgos circulares.

13. Código para generar rasgos con forma L.

14. Código para clasificar rasgos y ruido, basado en SEM.

15. Código para obtener matriz de distancias D, de Mahalanobis, entre los vectores de funciones LISA individuales de todos los puntos.

16. Código para obtener dendrogramas y ejes del escalamiento multidimensional y la clasificación en rasgos y ruido.

```
-----------------------------------
# SIMULACION DE PATRONES PUNTUALES
-----------------------------------

# Realizacion de un inhomogeneous Poisson process

intens_function(x,y,n){ n+(20*sqrt(n)*(x+y))

#el max de intens en (0,1)x(0,1) es n+40*sqrt(n) }

intens1_function(x,y){ 60*(1+sin(2*pi*x)+sin(2*pi*y))
```

```
#el max de intens1 en (0,3)x(0,3) es max(z)=120 }

xvals_seq(0,3,len=151) yvals_seq(0,3,len=151) z_matrix(nrow=151,ncol=151)
for(i in 1:151){z[i,]_intens1(xvals[i],yvals)} persp(xvals,yvals,z)
image(xvals,yvals,z) image.legend(z) max(z)

inhomo_function(poly,nr){

# para funcion intens

max.inten_nr+40*sqrt(nr) par.pois_area(poly)*max.inten
pts_csr(poly,rpois(1,par.pois)) n_length(pts[,1]) max.inten_nr+40*sqrt(n)
keep_rep(F,n) keep[runif(n)<intens(pts[,1],pts[,2],n)/max.inten]_T
x_pts[,1] y_pts[,2] mypts_cbind(x[keep],y[keep]) return(mypts) }

inhomo1_function(poly){

#para funcion intens1

max.inten_120 par.pois_area(poly)*max.inten
pts_csr(poly,rpois(1,par.pois)) n_length(pts[,1]) keep_rep(F,n)
keep[runif(n)<intens1(pts[,1],pts[,2])/max.inten]_T x_pts[,1] y_pts[,2]
mypts_cbind(x[keep],y[keep]) return(mypts) }


#realizacion de un Poisson cluster process

cluster_function(par.inten,nr,sigma){

nparents_rpois(1,area(poly)*par.inten) parents_csr(poly,nparents)
noffspring_rpois(nparents,nr/nparents)

#numero de puntos resultantes

xdeviations_rnorm(noffspring),sd=sigma)

# x-desviaciones de los progenitores
```

```
ydeviations_rnorm(noffspring),sd=sigma)

# y-desviaciones de los progenitores

parentsx_parents[,1]

# x-coordenadas de progenitores
parentsy_parents[,2]

# y-coordenadas de progenitores

offspringx_rep(parentsx,noffspring)+xdeviations

# x-coordenadas de descend.

offspringy_rep(parentsy,noffspring)+ydeviations

# y-coordenadas de descend.

offspring_cbind(offspringx,offspringy)

# puntos del proceso cluster de Poisson

}

#realizacion de un hard-core process

f.hardcore_function(x,delta){
#funcion utilizada en tesis de Linda Brandt
p.24 f_NULL if(x>=0&x<=delta){f_x} else {f_delta} res_f return(res) }

hardcore_function(poly,nr,delta){

pts_csr(poly,rpois(1,nr)) x_pts[,1];y_pts[,2] n_length(pts[,1]) m_runif(n)
conteo_rep(0,n) keep_rep(T,n)
```

```
#delete events nn_n-1 for(i in 1:nn){
  for (j in (i+1):n){
#   if(i!=j){
     distij_sqrt(((x[i]-x[j])^2)+((y[i]-y[j])^2))
     ff_f.hardcore(m[j],delta)
      if(m[j]<m[i]){
        if(distij<ff){conteo[i]_conteo[i]+1}
        else{conteo[i]_conteo[i]}}
      else{conteo[i]_conteo[i]}

   }
 if(conteo[i]>=1){keep[i]_F}
 }
mypts_cbind(x[keep],y[keep]) return(mypts) }


------------------------------------------------------------
# CALCULO DE LAS DISTANCIAS AL K-ESIMO VECINO MAS PROXIMO
------------------------------------------------------------

NNclean.kthNND_function(data,k,distances=NULL){ data_as.matrix(data)
d_dim(data)[2] n_dim(data)[1] if (n>800){ options(object.size=5e7) } if
(is.null(distances)==T){ distances_dist(data) } if
(n!=attr(distances,"Size")){ stop(message="Distance object does not match
the data") }

kthNND_rep(0,n) Labels_1:(n-1) kthNND[1]_sort(distances[Labels])[k]
Labels[(2):(n-1)]_Labels[(2):(n-1)]+(n-1-1) for (i in 2:n){
kthNND[i]_sort(distances[Labels])[k] Labels[1:(i-1)]_Labels[1:(i-1)]+1
Labels[(i+1):(n-1)]_Labels[(i+1):(n-1)]+(n-i-1) }

alpha.d_(2*(pi**(d/2)))/(d*gamma(d/2)) dDk_function(x,lamda,k,d,alpha.d){
(exp(-lamda*alpha.d*(x**d))*2*((lamda*alpha.d)**k)*(x**(d*k-1)))/gamma(k)
}

kthNND

}
```

```
------------------------------------------------------------
# CALCULO DE LA DISTANCIA DE GRID AL K-ESIMO PUNTO MAS CERCANO
------------------------------------------------------------

nndistFk_function(data,grid,k){ ngrid_length(grid[,1])
ndata_length(data[,1]) nndFk_rep(0,ngrid) for(i in 1:ngrid){
 d_rep(0,ndata)
 xg_grid[i,1]
 yg_grid[i,2]
  for(j in 1:ndata){
   d[j]_(data[j,1]-xg)**2+(data[j,2]-yg)**2
  }
  nndFk[i]_sqrt(sort(d)[k])
} return(nndFk) }


-----------------------------------------
#CALCULO DE LA DENSIDAD PRODUCTO
-----------------------------------------

productdensity_function(data,delta,a,b,t,poly) {

#funcion para el calculo de la densidad producto global con correccion de
#borde de Osher

x_data[,1];y_data[,2];xy_cbind(x,y);n_length(x) m_length(t) delta_delta
a_a b_b

#lamda_pdense(xy,poly) g_rep(0,m) pair_NULL
pair_.Fortran("proden",t=as.double(t),n=as.integer(n),
        m=as.integer(m),delta=as.double(delta),
        a=as.double(a),b=as.double(b),
        x=as.double(x),y=as.double(y),g=as.double(g))
result_cbind(t,pair[[9]]) return(result) }


       subroutine proden(t,n,m,delta,a,b,x,y,g)
```

```
c      subroutina para el calculo de la densidad producto global con c
correccion de borde dada por c_X(r))

       integer n,m
       real*8 t,x,y,g,d,delta,de,pi
       real*8 den,w,suma,a,b
       dimension x(n),y(n),d(1400,1400),w(1400,1400),t(1400)
       dimension suma(1400),g(1400),den(1400)

       pi=3.141593d0
c      distancias euclideas
       do 40 i=1,n
         do 50 j=1,n
           if (j.ne.i) then
c           x1=dabs(x(i)-x(j)) c           y1=dabs(y(i)-y(j))
           d(i,j)=((x(i)-x(j))**2+(y(i)-y(j))**2)**(0.50d0)
           endif
50        continue 40       continue

        de=delta
        do 55 k=1,m
        suma(k)=0.0d0
         do 60 i=1,n
          do 70 j=1,n
           if (j.ne.i) then
            if (dabs(d(i,j)-t(k)).ge.de) then
              w(i,j)=0.0d0
            else
              w(i,j)=(3.0d0/(4.0d0*de))*(1.0d0-(((d(i,j)-t(k))**2)
     e     /(de**2)))
            endif
           else
             w(i,j)=0.0d0
           endif
       suma(k)=suma(k)+w(i,j)
70     continue 60     continue 55     continue
       do 100 k=1,m
```

```
       den(k)=pi*(2.0d0)*t(k)*
     e     ((a*b)-(t(k)*((2*(a+b))-t(k)))/pi)
       g(k)=suma(k)/den(k)
100    continue
       return
       end
```

```
-------------------------------------------
#CALCULO DE LAS FUNCIONES LISA
-------------------------------------------

lisafunctmodif_function(data,data1,delta,a,b,t,poly) {

# funcion para el calculo de la densidad producto individual (lisa) con
# correccion de borde de Ohser
# data1 sera en cada caso el ruido o el rasgo

x_data[,1];y_data[,2];xy_cbind(x,y);n_length(x)
xr_data1[,1];yr_data1[,2];n1_length(xr) m_length(t) delta_delta a_a b_b
#lamda_pdense.s(xy,poly) result_NULL

for(i in 1:n1){
  x1_xr[i]
  y1_yr[i]
  g_rep(0,m)
  pair_NULL
  pair_.Fortran("lisa1",t=as.double(t),n=as.integer(n),
        i=as.integer(i),
        m=as.integer(m),delta=as.double(delta),
        a=as.double(a),b=as.double(b),x1=as.double(x1),
        y1=as.double(y1),
        x=as.double(x),y=as.double(y),g=as.double(g))
  result_cbind(result,pair[[12]])
}

return(result) }
```

```
     subroutine lisa1(t,n,i,m,delta,a,b,x1,y1,x,y,g)
c      subroutina para el calculo de la densidad producto individual
(lisa) c      con correccion de borde dada por c_X(r))

       integer n,m,i
       real*8 t,x,y,g,d,delta,de,pi
       real*8 borde,w,suma,a,b,x1,y1
       dimension x(n),y(n),d(1500,1500),w(1500),t(1500)
       dimension suma(1500),g(1500),borde(1500)

       pi=3.141593d0
c      distancias euclideas de x1,y1 al resto

         do 50 j=1,n
           if (x1.ne.x(j)) then
           d(i,j)=((x1-x(j))**2+(y1-y(j))**2)**(0.50d0)
           endif
50         continue


        de=delta
        do 55 k=1,m
          suma(k)=0.0d0
           do 70 j=1,n
            if (x1.ne.x(j)) then
             if (dabs(d(i,j)-t(k)).ge.de) then
              w(j)=0.0d0
            else
              w(j)=(3.0d0/(4.0d0*de))*(1.0d0-(((d(i,j)-t(k))**2)
     e     /(de**2)))
            endif
           else
             w(j)=0.0d0
           endif
       suma(k)=suma(k)+w(j)
```

```
70      continue

        borde(k)=(n-1)/(pi*(2.0d0)*t(k)*(a*b)*
     e      ((a*b)-(t(k)*((2*(a+b))-t(k)))/pi))
        g(k)=suma(k)*borde(k)
55      continue


        return
        end
```

```
-------------------------------------------------------
# CODIGO NNCLEAN1, para el calculo de las variables
# que clasifiquen en rasgo y ruido las localizaciones
# espaciales
-------------------------------------------------------

NNclean1<-function(data, k, distances = NULL, edge.correct=F, wrap = 0.1,
convergance = 0.001) {

    data <- as.matrix(data)
    d <- dim(data)[2]
    n <- dim(data)[1]
    if(n > 800) {
        options(object.size = 50000000)
    }
# # Correccion toroidal # if( (d==2)  && (edge.correct==T) ){
     r1 <- diff(range(data[,1]))
     r2 <- diff(range(data[,2]))
     tran2 <- matrix(c(rep(0,n),rep(r2,n)),byrow=F,nrow=n)
     tran1 <- matrix(c(rep(r1,n),rep(0,n)),byrow=F,nrow=n)
    aux.dat <- rbind(data+tran1,data-tran1,data+tran2,data-tran2,
    data+tran1+tran2,data-tran1+tran2,data-tran1-tran2, data+tran1-tran2)
    aux.dat <- aux.dat[ aux.dat[,1] < (max(data[,1]) +wrap*r1) &
aux.dat[,1] >
   (min(data[,1]) -wrap*r1) & aux.dat[,2] < (max(data[,2]) +wrap*r2) &
```

```
   aux.dat[,2] > (min(data[,2]) -wrap*r2), ]
    full.data <-  rbind(data,aux.dat)
    }
    else{
        full.data <- data
    }

# Funcion para la densidad de D_K #
    dDk <- function(x, lambda, k, d, alpha.d)
    {
        (exp( - lambda * alpha.d * x^d) * 2 * (lambda * alpha.d)^k * x^
        ( d * k - 1))/gamma(k)
    }

#
        if( is.null(distances) ){ distances <- dist(full.data)}
# #  Calculo de la distancia al k-esimo vecino mas cercano #
        kthNND <- rep(0, n)
        Labels <- 1:(n - 1)
        kthNND[1] <- sort(distances[Labels])[k]
        Labels[(2):(n - 1)] <- Labels[(2):(n - 1)] + (n - 1 - 1)
        for(i in 2:n) {
            kthNND[i] <- sort(distances[Labels])[k]
            Labels[1:(i - 1)] <- Labels[1:(i - 1)] + 1
            Labels[(i + 1):(n - 1)] <- Labels[(i + 1):(n - 1)] + (n-i-1)
        }
    }
    kthNND <- kthNND[1:n]
    alpha.d <- (2 * pi^(d/2))/(d * gamma(d/2))
# # Algoritmo E-M #
    delta <- rep(0, n)
    delta[kthNND > (min(kthNND) + diff(range(kthNND))/3)] <- 1

#asigno a valores altos de kthNND delta=1

    p <- 0.5
    lambda1 <- k/(alpha.d * mean((kthNND[delta == 0])^d))
```

```
    lambda2 <- k/(alpha.d * mean((kthNND[delta == 1])^d))
    loglik.old <- 0
    loglik.new <- 1

    while(abs((loglik.new - loglik.old)/loglik.new) > convergance) {
# E - step
        delta <- (p * dDk(kthNND, lambda1, k = k, d = d, alpha.d =
            alpha.d))/(p * dDk(kthNND, lambda1, k = k, d = d,
            alpha.d = alpha.d) + (1 - p) * dDk(kthNND, lambda2, k = k,
d = d,
            alpha.d = alpha.d))
# M - step
        p <- sum(delta)/n
        lambda1 <- (k * sum(delta))/(alpha.d * sum((kthNND^d) * delta))
        lambda2 <- (k * sum((1 - delta)))/(alpha.d * sum((kthNND^d) *
        (1 - delta)))loglik.old <- loglik.new
        loglik.new <- sum( - p * lambda1 * alpha.d * ((kthNND^d) * delta)
 - (
        1 - p) * lambda2 * alpha.d * ((kthNND^d) * (1 -delta)) + delta *
 k *
        log(lambda1 * alpha.d)+(1 - delta) * k * log(lambda2 * alpha.d))
        print(loglik.new)
    }
# # Plot del histograma # #
             par(mfrow=c(2,2))
             par(pty="s")
             plot(data,xlab="x",ylab="y")
          hist(kthNND, nclass = 20, axes = T, ylab = "Estimacion de la
Mixtura",
             xlim = c(0, max(kthNND)),
              probability=T, xlab = paste("Distance to", eval(k),
              "th nearest neighbour"))
          box()
          support <- seq(0, max(kthNND), length = 200)
          lines(support,(p * dDk(support, lambda1, k = k, d = d, alpha.d
             = alpha.d) + (1 - p) * dDk(support, lambda2, k = k, d
             = d, alpha.d = alpha.d)))
```

```
# # z recoge las clasificaciones. 1= cluster. 0= ruido. #
    probs<- dDk(kthNND, lambda1, k = k, d = d, alpha.d = alpha.d)/(dDk(
        kthNND, lambda1, k = k, d = d, alpha.d = alpha.d) + dDk(kthNND,
        lambda2, k = k, d = d, alpha.d = alpha.d))
    return(z=round(probs), probs, lambda1, lambda2, p, kthNND)
}
-------------------------------------------------------------------------
# CODIGO PARA SIMULAR EL RUIDO Y RASGOS Y OBTENCION DE LA CLASIFICACION
-------------------------------------------------------------------------

#a recoge el ruido a_csr.s(unit.pol,400)
#b recoge el patron a restaurar
b_csr.s(cbind(c(0.4,0.6,0.6,0.4),c(0.2,0.2,0.6,0.6)),300) ab_rbind(a,b)

par(mfrow=c(2,2)) par(pty="s")

#1 pru.k5_NNclean1(ab,5) sel_pru.k5$z==1
absel.k5_cbind(ab[,1][sel],ab[,2][sel])
plot(absel.k5,ylim=c(0,1),xlim=c(0,1),xlab="seleccion",ylab="rasgo")

intensidad.k5_interp(ab[,1],ab[,2],pru.k5$probs,xo=seq(0,1,len=80),
yo=seq(0,1,len=80))
image(intensidad.k5,xlab="Mapa de intensidades")
image.legend(intensidad.k5)

#2 pru.k10_NNclean1(ab,10) sel_pru.k10$z==1
absel.k10_cbind(ab[,1][sel],ab[,2][sel])
plot(absel.k10,ylim=c(0,1),xlim=c(0,1),xlab="seleccion",ylab="rasgo")

intensidad.k10_interp(ab[,1],ab[,2],pru.k10$probs,xo=seq(0,1,len=80),
yo=seq(0,1,len=80))
image(intensidad.k10,xlab="Mapa de intensidades")
image.legend(intensidad.k10,x=0,y=1.3,size=c(2.0,0.2))

#3 pru.k15_NNclean1(ab,15) sel_pru.k15$z==1
absel.k15_cbind(ab[,1][sel],ab[,2][sel])
plot(absel.k15,ylim=c(0,1),xlim=c(0,1),xlab="seleccion",ylab="rasgo")
```

```
intensidad.k15_interp(ab[,1],ab[,2],pru.k15$probs,xo=seq(0,1,len=80),
yo=seq(0,1,len=80))
image(intensidad.k15,xlab="Mapa de intensidades")
image.legend(intensidad.k15,x=0,y=1.3,size=c(2.0,0.2))

#a recoge el ruido a_csr(unit.pol,600)
#b recoge el patron a restaurar
b_csr(cbind(c(0.4,0.8,0.8,0.4),c(0.2,0.2,0.6,0.6)),300)
b1_csr(cbind(c(0.1,0.6,0.6,0.1),c(0.8,0.8,0.9,0.9)),300) ab1_rbind(a,b,b1)

par(mfrow=c(2,2)) par(pty="s")

#1 pru1.k5_NNclean1(ab1,5) sel_pru1.k5$z==1
absel.k5_cbind(ab1[,1][sel],ab1[,2][sel])
plot(absel.k5,ylim=c(0,1),xlim=c(0,1),xlab="seleccion",ylab="rasgo")

intensidad.k5_interp(ab1[,1],ab1[,2],pru1.k5$probs,xo=seq(0,1,len=80),
yo=seq(0,1,len=80))
image(intensidad.k5,xlab="Mapa de intensidades")
image.legend(intensidad.k5)

#2 pru1.k10_NNclean1(ab1,10) sel_pru1.k10$z==1
absel.k10_cbind(ab1[,1][sel],ab1[,2][sel])
plot(absel.k10,ylim=c(0,1),xlim=c(0,1),xlab="seleccion",ylab="rasgo")

intensidad.k10_interp(ab1[,1],ab1[,2],pru1.k10$probs,xo=seq(0,1,len=80),
yo=seq(0,1,len=80))
image(intensidad.k10,xlab="Mapa de intensidades")
image.legend(intensidad.k10)

#3 pru1.k15_NNclean1(ab,15) sel_pru1.k15$z==1
absel.k15_cbind(ab1[,1][sel],ab1[,2][sel])
plot(absel.k15,ylim=c(0,1),xlim=c(0,1),xlab="seleccion",ylab="rasgo")

intensidad.k15_interp(ab1[,1],ab1[,2],pru1.k15$probs,xo=seq(0,1,len=80),
yo=seq(0,1,len=80))
```

```
image(intensidad.k15,xlab="Mapa de intensidades")
image.legend(intensidad.k15)

-------------------------------------------------------------------------
# CODIGO PARA LA DENSIDAD PRODUCTO Y FUNCIONES LISA DE DATOS SIMULADOS
-------------------------------------------------------------------------

ab.pd_productdensity(ab,0.1,1,1,seq(0.0005979,0.5,len=100),unit.square)
ab.lisa.ruido.1_lisafunctmodif(ab,a,0.1,1,1,seq(0.0005979,0.5,len=100),
unit.square)
ab.lisa.rasgo.1_lisafunctmodif(ab,b,0.1,1,1,seq(0.0005979,0.5,len=100),
unit.square)

-------------------------------------------------------------------------
# CODIGO PARA CLUSTER JERARQUICO Y CLUSTER BASADO EN EL ALGORITMO EM DE
DATOS SIMULADOS
-------------------------------------------------------------------------

cltree.ab_mhtree(ab) cl_mhclass(cltree.ab, 1:2) clpairs (ab, cl[,"2"])
cltree_mhtree(ab, modelid = "EI") cltree.abVI_mhtree(ab, modelid = "VI")
cl.abVI2_mhclass(cltree.abVI, 2) z.abVI2_me( ab, modelid = "VI" ,
ctoz(cl.abVI2)) clpairs(ab, ztoc(z.abVI2)) uncer.ab_1 - apply( z.abVI2, 1,
"max") quantile(uncer.ab) pars.abVI2_mstep( ab, modelid = "VI", z.abVI2)
cltree.ab_mhtree(ab) cl.ab12_mhclass(cltree.ab, 1:2) z.abEI2_me( ab,
modelid = "EI" , ctoz(cl.ab12[,"2"])) bic(ab, modelid = "EI", z.abEI2)
z.abVI2_me( ab, modelid = "VI" , ctoz(cl.ab12[,"2"])) bic(ab, modelid =
"VI", z.abVI2) bicvals.1_emclust1( ab, nclus = 1:2, modelid = c("VI",
"VVV")) bicvals.2_emclust( ab, nclus = 1:2) plot(bicvals.2)
-------------------------------------------------------------------------
# CODIGOS CAPITULO 5
-------------------------------------------------------------------------

# 1. CODIGO PARA REPRESENTAR FUNCIONES LISA, HISTOGRAMAS ETC.
# Valido para todas posibilidades, solo cambiar el titulo del
# postscript donde se ha de guardar las representaciones

REPRESENTACION.LISA<-function(data,delta, a, b, t,poly,edge.correct=T,
```

```
d=2,wrap) {

#    DATA RECOGE TODOS LOS PUNTOS ORIGINALES
#    TANTO RASGO COMO RUIDO)

    lt<-length(t)
    data <- as.matrix(data)
    d<-length(data[1,])

#   n recogera la cantidad total de valores:
    n<-length(data[,1])
    n.ruido<-length(ruidoR[,1])
    n.rasgo<-length(rasgoR[,1])

    cat("n de puntos=",n,"\n")
    ar<-a*b

    lam<-(n-1)/ar

    distancias.a.media.teorica<-rep(0,n)
    funciones.lisa.de.todos.ptos<-matrix(0,nrow=lt,ncol=n)

    if(n > 800) {
        options(object.size = 50000000)
    }

# CORRECCION DE BORDE TOROIDAL:

if( (d==2)  && (edge.correct==T) ){ r1 <- diff(range(data[,1])) r2 <-
diff(range(data[,2])) tran2 <-
matrix(c(rep(0,n),rep(r2,n)),byrow=F,nrow=n) tran1 <-
matrix(c(rep(r1,n),rep(0,n)),byrow=F,nrow=n) aux.dat <-
rbind(data+tran1,data-tran1,data+tran2,data-tran2, data+tran1+tran2,
data-tran1+tran2, data-tran1-tran2, data+tran1-tran2) aux.dat <- aux.dat[
aux.dat[,1] < (max(data[,1]) +wrap*r1) & aux.dat[,1] > (min(data[,1])
-wrap*r1) & aux.dat[,2] < (max(data[,2]) +wrap*r2) & aux.dat[,2] >
(min(data[,2]) -wrap*r2), ] full.data <-  rbind(data,aux.dat) } else{
```

```
full.data <- data }

data <- full.data

# CALCULO DE LAS FUNCIONES LISA DE CADA UNO DE LOS PTOS
# ESTA FUNCION CALCULA LA DENSIDAD PRODUCTO INDIVIDUAL
# (FUNCION LISA)CON CORRECCION DE BORDE OSHER,
# esta implementada en fortran, en el archivo:
# product-density-lisa-funct_s.so, para lo cual, antes
# de lanzar el programa se ha de hacer un dyn.load ("product...")

funciones.lisa.de.todos.ptos<-lisafunct(data,delta,a,b,t,poly)

long.intervalo<-(t[length(t)]-t[1])/length(t) mediateorica<-lam^2+(lam/ar)

# CALCULO DE LAS DISTANCIAS (L1) DE CADA UNA DE LAS FUNCIONES LISA
# INDIVIDUAL A LA MEDIA TEORICA

distancias.a.media.teorica.l.uno<-rep(0,n)

for(i in 1:n){ distancias.a.media.teorica.l.uno[i]<-sum(abs
(funciones.lisa.de.todos.ptos[,i]-mediateorica)*long.intervalo) }

# CALCULO DE LAS DISTANCIAS (L2) DE CADA UNA DE LAS FUNCIONES LISA
# INDIVIDUAL A LA MEDIA TEORICA

distancias.a.media.teorica.l.dos<-rep(0,n)

for(i in 1:n){
distancias.a.media.teorica.l.dos[i]<-sum((
(funciones.lisa.de.todos.ptos[,i]-mediateorica)**2)
*long.intervalo)
}

# CALCULO DE LAS DISTANCIAS (divergencia de KULLBACK) DE CADA UNA DE
# LAS FUNCIONES LISA INDIVIDUAL A LA MEDIA TEORICA
```

```
distancias.a.media.teorica.KULLBACK<-rep(0,n)

for(i in 1:n){
distancias.a.media.teorica.KULLBACK[i]<-sum(
(abs(funciones.lisa.de.todos.ptos[,i]-
mediateorica)*abs(log(funciones.lisa.de.todos.ptos[,i]/mediateorica)
))*long.intervalo)
}

    postscript("3rasgos_cuadrado_formaL_circulo_ruido_.eps")

    par(mfrow=c(3,3))

#   Representacion de puntos originales, rasgo y ruido, por separado:

    plot(rbind(ruidoR,rasgoR),xlab="Datos originales, Coordenada x",
    ylab="Coordenada y",
    xlim=c(0,1),ylim=c(0,1),pch=".")
    plot(ruidoR,xlab="Ruido original, Coordenada x",ylab="Coordenada y",
    xlim=c(0,1),
    ylim=c(0,1),pch=".")
    plot(rasgoR,xlab="Rasgo original, Coordenada x",ylab="Coordenada y",
    xlim=c(0,1),
    ylim=c(0,1),pch=".")


#   Representacion de graficas de lisas individuales, conjunto
#   y por separado:

    plot(t,funciones.lisa.de.todos.ptos[,1],xlab="Representacion f.
    Lisas de todos datos,
    distancias",ylab="Valor Lisa",type="l",col=1,ylim=c(0,1.2e+07))
    for(k in 2:n.ruido){lines(t,funciones.lisa.de.todos.ptos[,k],col=1)}
    for(k in (n.ruido+1):n){lines(t,funciones.lisa.de.todos.ptos[,k],col=3)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)
```

```
    plot(t,funciones.lisa.de.todos.ptos[,1],xlab="Representacion f.
    Lisas de Ruido,
    distancias",ylab="Valor Lisa",type="l",col=1,ylim=c(0,1.2e+07))
    for(k in 2:n.ruido){lines(t,funciones.lisa.de.todos.ptos[,k],col=1)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)

    plot(t,funciones.lisa.de.todos.ptos[,n.ruido+1],xlab="Representacion f.
    Lisas de Rasgo,
    distancias",ylab="Valor Lisa",type="l",col=3,ylim=c(0,1.2e+07))
    for(k in (n.ruido+2):n){lines(t,funciones.lisa.de.todos.ptos[,k],col=3)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)

#   Histograma de las distancias de las funciones LISA a media teorica:

    hist(distancias.a.media.teorica.l.uno[1:n.ruido],col=2,br=20,xlab=
    "Distancia L1 de
    funciones LISA a media teorica",ylab="Numero de individuos",main="")
    hist(distancias.a.media.teorica.l.uno[(n.ruido+1):n],add=T)

    hist(distancias.a.media.teorica.l.dos[1:n.ruido],col=2,br=20,xlab=
    "Distancia L2 de
    funciones LISA a media teorica",ylab="Numero de individuos",main="")
    hist(distancias.a.media.teorica.l.dos[(n.ruido+1):n],add=T)

    hist(distancias.a.media.teorica.KULLBACK[1:n.ruido],col=2,br=20,
    xlab="Distancia
    KULLBACK de funciones LISA a media teorica",ylab="Numero de individuos",
    main="")
    hist(distancias.a.media.teorica.KULLBACK[(n.ruido+1):n],add=T)

    dev.off()
}
-------------------------------------------------------------------------
# 2. CODIGO PARA REPRESENTAR FUNCIONES LISA de diferentes
# rasgos por separado
```

```
REPRESENTACION.LISA<-function(data,delta, a, b,
t,poly,edge.correct=T,wrap) {

#    DATA RECOGE TODOS LOS PUNTOS ORIGINALES
#    TANTO RASGO COMO RUIDO)

    lt<-length(t)
    data <- as.matrix(data)
    d<-length(data[1,])

#   n recogera la cantidad total de valores:
    n<-length(data[,1])
    n.ruido<-length(ruidoR[,1])
    n.rasgo<-length(rasgoR[,1])

    cat("n de puntos=",n,"\n")
    ar<-a*b

    lam<-(n-1)/ar

#   INICIALIZANDO LAS VARIABLES:

    distancias.a.media.teorica<-rep(0,n)
    funciones.lisa.de.todos.ptos<-matrix(0,nrow=lt,ncol=n)

#   Instruccion que permite utilizar numero de puntos alto:

    if(n > 800) {
        options(object.size = 50000000)
    }

# CORRECCION DE BORDE TOROIDAL:

if( (d==2)  && (edge.correct==T) ){ r1 <- diff(range(data[,1])) r2 <-
diff(range(data[,2])) tran2 <-
matrix(c(rep(0,n),rep(r2,n)),byrow=F,nrow=n) tran1 <-
```

```
matrix(c(rep(r1,n),rep(0,n)),byrow=F,nrow=n) aux.dat <-
rbind(data+tran1,data-tran1,data+tran2,data-tran2, data+tran1+tran2,
data-tran1+tran2, data-tran1-tran2, data+tran1-tran2) aux.dat <- aux.dat[
aux.dat[,1] < (max(data[,1]) +wrap*r1) & aux.dat[,1] > (min(data[,1])
-wrap*r1) & aux.dat[,2] < (max(data[,2]) +wrap*r2) & aux.dat[,2] >
(min(data[,2]) -wrap*r2), ] full.data <-  rbind(data,aux.dat) } else{
full.data <- data }

data<-full.data

# CALCULO DE LAS FUNCIONES LISA DE CADA UNO DE LOS PTOS
# ESTA FUNCION CALCULA LA DENSIDAD PRODUCTO INDIVIDUAL
# (FUNCION LISA)CON CORRECCION DE BORDE OSHER, esta funcion
# esta implementada en fortran, en el archivo:
# product-density-lisa-funct_s.so, para lo cual, antes
# de lanzar el programa se ha de hacer un dyn.load ("product...")

funciones.lisa.de.todos.ptos<-lisafunct(data,delta,a,b,t,poly)

long.intervalo<-(t[length(t)]-t[1])/length(t) mediateorica<-lam^2+(lam/ar)

# CALCULO DE LAS DISTANCIAS (L1) DE CADA UNA DE LAS FUNCIONES LISA
#INDIVIDUAL A LA MEDIA TEORICA

distancias.a.media.teorica.l.uno<-rep(0,n)

for(i in 1:n){
distancias.a.media.teorica.l.uno[i]<-sum(abs(funciones.lisa.de.todos.
ptos[,i]-mediateorica)*long.intervalo) }

# CALCULO DE LAS DISTANCIAS (L2) DE CADA UNA DE LAS FUNCIONES LISA
# INDIVIDUAL A LA MEDIA TEORICA

distancias.a.media.teorica.l.dos<-rep(0,n)

for(i in 1:n){
distancias.a.media.teorica.l.dos[i]<-sum(((funciones.lisa.de.todos.
```

```
ptos[,i]-mediateorica)**2)*long.intervalo) }

# CALCULO DE LAS DISTANCIAS (divergencia de KULLBACK) DE CADA UNA DE
# LAS FUNCIONES LISA INDIVIDUAL A LA MEDIA TEORICA
distancias.a.media.teorica.KULLBACK<-rep(0,n)

for(i in 1:n){
distancias.a.media.teorica.KULLBACK[i]<-sum((abs(funciones.lisa.de.todos.
ptos[,i]-mediateorica)*abs(log(funciones.lisa.de.todos.ptos[,i]/
mediateorica)))*long.intervalo) }

    postscript("3_rasgos_cuadrado_formaL_circulo_ruido_.eps")

    par(mfrow=c(3,3))

#   Representacion de puntos originales, rasgo y ruido, por separado:

    plot(rbind(ruidoR,rasgoR),xlab="Datos originales, Coordenada x",ylab=
    "Coordenada y",
    xlim=c(0,1),ylim=c(0,1))
    plot(ruidoR,xlab="Ruido original, Coordenada x",ylab="Coordenada y",
    xlim=c(0,1),
    ylim=c(0,1))
    plot(rasgoR,xlab="Rasgo original, Coordenada x",ylab="Coordenada y",
    xlim=c(0,1),
    ylim=c(0,1))

#   Representacion de graficas de lisas individuales, conjunto
#   y por separado:

    plot(t,funciones.lisa.de.todos.ptos[,1],xlab="Representacion f.
    Lisas de todos datos,
    distancias",ylab="Valor Lisa",type="l",col=1,ylim=c(0,1.2e+07))
    for(k in 2:n.ruido){lines(t,funciones.lisa.de.todos.ptos[,k],
col=1)}
    for(k in (n.ruido+1):n){lines(t,funciones.lisa.de.todos.ptos[,k],
col=3)}
```

```
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=
100),col=2)

    plot(t,funciones.lisa.de.todos.ptos[,1],xlab="Representacion f.
    Lisas de Ruido,
    distancias",ylab="Valor Lisa",type="l",col=1,ylim=c(0,1.2e+07))
    for(k in 2:n.ruido){lines(t,funciones.lisa.de.todos.ptos[,k],
col=1)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,
len=100),col=2)

    plot(t,funciones.lisa.de.todos.ptos[,n.ruido+1],xlab="Representacion f.
    Lisas de Rasgo,
    distancias",ylab="Valor Lisa",type="l",col=3,ylim=c(0,1.2e+07))
    for(k in (n.ruido+2):n){lines(t,funciones.lisa.de.todos.ptos[,k],
col=3)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)

    plot(t,funciones.lisa.de.todos.ptos[,501],xlab="Representacion f.
    Lisas de Rasgo
    Cuadrado, distancias",ylab="Valor Lisa",type="l",col=1,
ylim=c(0,1.2e+07))
    for(k in 502:600){lines(t,funciones.lisa.de.todos.ptos[,k],col=3)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)

    plot(t,funciones.lisa.de.todos.ptos[,601],xlab="Representacion f.
    Lisas de Rasgo forma
    L, distancias",ylab="Valor Lisa",type="l",col=1,ylim=c(0,1.2e+07))
    for(k in 602:1000){lines(t,funciones.lisa.de.todos.ptos[,k],col=3)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)

    plot(t,funciones.lisa.de.todos.ptos[,1001],xlab="Representacion f.
    Lisas de Rasgo
    Circular, distancias",ylab="Valor Lisa",type="l",col=1,
```

```
ylim=c(0,1.2e+07))
    for(k in 1002:n){lines(t,funciones.lisa.de.todos.ptos[,k],col=3)}
    lines(seq(0.0059,0.5,len=100),seq(mediateorica,mediateorica,len=100),
    col=2)

    dev.off()
}
-------------------------------------------------------------------------------
# 3. LANZADERA DE LAS REPRESENTACIONES
# Valida para cualquier lanzamiento de uno, dos o tres rasgos, basta con
# anular las lineas correspondientes a los rasgos que no se desea incluir,
# e ir cambiando las cantidades de los puntos

ruidoR<-csr(polygrande,500) rasgoC<-csr(polypeque,100)
rasgoL<-csr(nuestro.poly,400)
poly.circulo<-funcion.para.circulo(centrox=0.8,centroy=0.8,radio=0.15,
npts=8000,
poly=polygrande)
rasgoR<-rbind(rasgoC,rasgoL,poly.circulo)

representa<-REPRESENTACION.LISA(data=rbind(ruidoR,rasgoR),delta=0.10, a=1,
b=1, t=seq(0.005,0.5,len=100),poly=polygrande, edge.correct=F, wrap = 0.1)
-------------------------------------------------------------------------------
# 4. CODIGO PARA GENERAR RASGOS CIRCULARES

funcion.para.circulo<-function(centrox,centroy,radio,npts,poly) {
    xo<-centrox
    yo<-centroy
    r<-radio

#   El circulo se ha de recortar de una zona de puntos

    pts.iniciales<-csr(poly,npts)
    distancia<-rep(0,npts)

#   Bucle para formar circunferencia:
    for(i in 1:npts){
```

```
    distancia[i]<-sqrt((pts.iniciales[i,1]-xo)^2+(pts.iniciales[i,2]-yo)^2)
    }

#   Seleccionamos solo los de dentro circulo:
    sel<-distancia<=radio
    result<-pts.iniciales[sel,]
    return(result)
}
-----------------------------------------------------------------
# 5. CODIGO PARA GENERAR RASGOS CON FORMA L

# Con las siguientes instrucciones (estan comentadas  porque solo
# se hace la primera vez), se dibuja el poligono, como se quiere:

#library(splancs)
#plot(runif(100),runif(100),type="n",xlab="",ylab="")
#nuestro.poly<-getpoly()

# El detalle es que hay que cerrar el poligono para ello se toman los
#valores iniciales y se  le cierra con el punto inicial:

#nuestro.poly<-rbind(nuestro.poly,c(0.1803557,0.4718537))
#ruido<-csr(unit.square,400)
#rasgo.L<-csr(nuestro.poly,300)

# Para representarlos:

#plot(runif(100),runif(100),type="n",xlab="",ylab="")
#points(nuestro.poly,type="l")
#ruido<-csr(unit.square,400)
#rasgo<-csr(nuestro.poly,300)
#points(ruido)
#points(rasgo)

#plot(rbind(ruido,rasgo))
-----------------------------------------------------------------
6.CODIGO PARA CLASIFICAR RASGOS Y RUIDO, BASADO EN SEM
```

```
# CODIGO PARA CLASIFICAR PUNTOS EN RASGO
# Y RUIDO, UTILIZANDO LADIVERGENCIA DE
# KULLBACK. En caso de utilizar L1 o L2, bastara
# cambiar las lineas de codigo donde se calcule la
# distancia.
# Valido para clasificar cualquier numero
# de rasgos, basta con cambiar el postscript

SEM.LISA<-function(data, delta, a, b, t,
nptosruidosimulacion,nptosrasgosimulacion,poly, poly1,edge.correct=F,
wrap, convergance) {

# Poly recoge el poligono grande que define la imagen
# (en simulaciones es el cuadrado unidad)y poly1 recoge
# el poligono donde se genera el rasgo para la simulacion
# dentro del SEM.

# DATA RECOGE TODOS LOS PUNTOS ORIGINALES
# (TANTO RASGOS COMO RUIDO)

    lt<-length(t)
    data <- as.matrix(data)
    d<-length(data[1,])

#   n recogera la cantidad total de valores:

    n<-length(data[,1])
    n.ruido<-length(ruido[,1])
    n.rasgo<-length(rasgo[,1])

    cat("n de puntos=",n,"\n")
    ar<-a*b

    lam<-(n-1)/ar

    distancias.a.media.teorica<-rep(0,n)
    funciones.lisa.de.todos.ptos<-matrix(0,nrow=lt,ncol=n)
```

```
    if(n > 800) {
        options(object.size = 50000000)
    }

# CORRECCION DE BORDE TOROIDAL:

if( (d==2)  && (edge.correct==T) ){ r1 <- diff(range(data[,1])) r2 <-
diff(range(data[,2])) tran2 <-
matrix(c(rep(0,n),rep(r2,n)),byrow=F,nrow=n) tran1 <-
matrix(c(rep(r1,n),rep(0,n)),byrow=F,nrow=n) aux.dat <-
rbind(data+tran1,data-tran1,data+tran2,data-tran2, data+tran1+tran2,
data-tran1+tran2, data-tran1-tran2, data+tran1-tran2) aux.dat <- aux.dat[
aux.dat[,1] < (max(data[,1]) +wrap*r1) & aux.dat[,1] > (min(data[,1])
-wrap*r1) & aux.dat[,2] < (max(data[,2]) +wrap*r2) & aux.dat[,2] >
(min(data[,2]) -wrap*r2), ] full.data <-  rbind(data,aux.dat) } else{
full.data <- data }

# CALCULO DE LAS FUNCIONES LISA DE CADA UNO DE LOS PTOS

funciones.lisa.de.todos.ptos<-lisafunct(data,delta,a,b,t,poly)

# CALCULO DE LAS DISTANCIAS (KULLBACK) DE CADA UNA DE LAS FUNCIONES LISA
# INDIVIDUAL A LA MEDIA TEORICA

long.intervalo<-(t[length(t)]-t[1])/length(t) mediateorica<-lam^2+(lam/ar)

for(i in 1:n){
distancias.a.media.teorica[i]<-sum((abs(funciones.lisa.de.todos.ptos[,i]-
mediateorica)*abs(log(funciones.lisa.de.todos.ptos[,i]/mediateorica)))
*long.intervalo)
}

#ALGORITMO SEM:

#   ASIGNACION DE DELTAS INICIAL, DANDO VALORES ALTOS DE
# distancias.a.media.teorica delta EL VALOR 1
```

```
    deltak<-rep(0,n)
    deltak[distancias.a.media.teorica>(min(distancias.a.media.teorica)+
    diff(range(distancias.a.media.teorica))/3)]<-1

    p<-0.5
    loglik.old<-0
    loglik.new<-1

#   CRITERIO DE CONVERGENCIA

    cont<-0
    while(abs((loglik.new-loglik.old)/loglik.new)>convergance){

    cont<-cont+1

#   E - STEP,SE CALCULA EL VALOR ESPERADO, EN CADA ITERACION, DE
# LOS VALORES DELTA ASOCIADOS A CADA INDIVIDUO. ESTE VALOR ESTARA
# COMPRENDIDO ENTRE 0 Y 1.

#       S - STEP, para el rasgo y ruido, simulamos puntos

#           de rasgo y de ruido, calculamos su funcion

#           lisa y su distancia a la media teorica de rasgo,

#           calculamos histogramas por separado de rasgo y

#           de ruido de las distancias, situamos

#           en cada uno de ellos nuestro valor de distancia

#           del individuo i-esimo y la frecuencia correspondiente

#           a la clase donde se haya situado, sera valor

#           de fRA y fRU
```

```
# Asigno a nRA y nRU, la cantidad que haya quedado,

# en la clasificacion de la iteracion anterior,de

# puntos de rasgo y ruido:

            nRA<-length(deltak[deltak == 1])
            nRU<-length(deltak[deltak == 0])

#           En cada iteracion, se generara rasgo en el cuadrado

#           unidad, tantos como le digamos en lanzadera y ruido

#           en el poligono cuadrado de 0.1 de lado, tantos como

#           le digamos en lanzadera:

            ruido.simulacion<-csr(poly,nptosruidosimulacion)
            rasgo.simulacion<-csr(poly1,nptosrasgosimulacion)

#           Instruccion que hace salir del programa cuando
#           se de el caso en que todos los datos, en una itera-
#           cion sean rasgo o ruido:

            if(nRU==0){break}
            if(nRU==n){break}

            fRU<-rep(0,nRU)
            fRA<-rep(0,nRA)

#           En data.simulacion recogere los valores simulados:

            data.simulacion<-NULL
            data.simulacion<-rbind(ruido.simulacion,rasgo.simulacion)

#           Necesito saber la cantidad total de puntos simulados:
```

```
            g<-length(data.simulacion[,1])

#           Calculo la funcion LISA individual de cada punto  de
#            la simulacion:

            funciones.lisa.de.todos.ptos.simulacion<-matrix(0,nrow=lt,
            ncol=g)
            distancias.a.media.teorica.simulacion<-rep(0,g)
            funciones.lisa.de.todos.ptos.simulacion<-lisafunct(data.
            simulacion, delta,a,b,t,poly)


#           Calculo de las distancias (KULLBACK)de las funciones LISA

#           individual de la simulacion a la teorica

#           La intensidad para este calculo depende ahora

#           de la cantidad de puntos que tenga en la simulacion,

#           de la misma forma que la media teorica:

            lam.simulacion<-(g-1)/ar

            mediateorica.simulacion<-lam.simulacion^2+(lam.simulacion/ar)

            for(k in 1:g){
            distancias.a.media.teorica.simulacion[k]<-sum
            ((abs(funciones.lisa.de.todos.ptos.simulacion[,k]-
            mediateorica.simulacion)*abs(log(funciones.lisa.de.todos.ptos.
            simulacion[,k]/mediateorica.simulacion)))*long.intervalo)
            }

            nptosruidosimulacion1<-nptosruidosimulacion+1
```

```
#           Llamada al histograma para las distancias de las

#           funciones LISA de los puntos ruido de la simulacion,

#            y los almaceno en aRU:

            aRU<-hist(distancias.a.media.teorica.simulacion[1:
            nptosruidosimulacion],plot=F)


#           Llamada al histograma para las distancias de las

#           funciones LISA de los puntos rasgo de la simulacion,

#           y los almaceno en aRA:

            aRA<-hist(distancias.a.media.teorica.simulacion
            [nptosruidosimulacion1:g],plot=F)


#       Para cada individuo, en cada iteracion del SEM:

        denominador<-rep(0,n)

        for(i in 1:n){

#           El siguiente bucle compara la distancia de la funcion

#           LISA del individuo i-esimo, con los limites de las

#           clases del histograma de los puntos de la

#           simulacion de ruido (o rasgo), para colocarlo en la

#           clase correspondiente y, a continuacion, le asigna el

#           valor de la frecuencia (en terminos relativos)de
```

```
#           esa clase, que le llamara fRU (o fRA)(llamo lenaRU

#           y lenaRA al numero de breaks menos uno, porque siempre

#           hay una marca mas que columnas de histograma):

            lenaRU<-length(aRU$breaks)-1
            lenaRA<-length(aRA$breaks)-1

            conteoRU<-0
            for(j in 1:lenaRU){
            if (distancias.a.media.teorica[i]>=aRU$breaks[j])
conteoRU<-conteoRU+1
            }
            conteoRA<-0
            for(j in 1:lenaRA){
            if (distancias.a.media.teorica[i]>=aRA$breaks[j])
conteoRA<-conteoRA+1
            }

            ifelse(conteoRU>0,fRU[i]<-aRU$count[conteoRU]/nRU,fRU[i]<-0)
            ifelse(conteoRA>0,fRA[i]<-aRA$count[conteoRA]/nRA,fRA[i]<-0)

#           Para evitar el caso en que en los histogramas la clase

#           de rasgo o ruido sea nula, por tanto el denominador de la

#           francion de los deltak, sea 0 y de error:

            denominador[i]<-(p*fRA[i])+((1-p)*fRU[i])
            ifelse(denominador[i]==0,deltak[i]<-0,deltak[i]<-(p*fRA[i])/
            ((p*fRA[i])+((1-p)*fRU[i])))


#       Dentro del bucle de cada individuo, calculamos la delta
#       con los valores de fRU y fRA obtenidos, (objetivo
```

```
#       del E-STEP):

       deltak[i]<-(p*fRA[i])/((p*fRA[i])+((1-p)*fRU[i]))

        }


#   FINALIZACION DE E-STEP: REDONDEO DE LA VARIABLE DELTA a 0 y 1:
#   deltak<-round(deltak)
#   M-STEP: CALCULA LOS PARaMETROS NECESARIOS, i.e.: LA PROPORCION
#   DE RASGO FRENTE AL TOTAL DE PUNTOS:

        p <- sum(deltak)/n


#   CALCULO DE LAS VARIABLES PARA EL CRITERIO DE CONVERGENCIA:

    loglik.old<-loglik.new
    vector.frecuencias.total<-c(fRU,fRA)
    loglik.new<-log(sum(vector.frecuencias.total))


#   FIN DEL ALGORITMO SEM:
    }


# La siguiente instruccion coloca en primera columna coordenada

# x del punto, en segunda, coordenada y y en tercera 0 o 1,

# segun sea ruido o rasgo:

    selRU<-deltak==0
    data.ruido.final<-data[selRU,]

    selRA<-deltak==1
    data.rasgo.final<-data[selRA,]
```

```
    numero.puntos.ruido.clasificado<-length(data.ruido.final[,1])
    numero.puntos.rasgo.clasificado<-length(data.rasgo.final[,1])


    matriz.resultado<-cbind(data,c(rep(0,n.ruido),rep(1,n.rasgo)),selRA)

    #calculo del porcentaje de aciertos y fallos en rasgo y ruido

    conteo.mala.clasif<-0
    conteo.buena.clasif<-0
    n.ruido.1<-n.ruido+1
    for(i in n.ruido.1:n){
    ifelse(matriz.resultado[i,3]==matriz.resultado[i,4],conteo.buena.clasif<-
    conteo.buena.clasif+1,conteo.mala.clasif<-conteo.mala.clasif+1)
    }
    porcentaje.buena.clasif<-conteo.buena.clasif/n.rasgo
    porcentaje.mala.clasif<-conteo.mala.clasif/n.rasgo

    cat("conteo.buena.clasif",conteo.buena.clasif,"\n")
    cat("conteo.mala.clasif",conteo.mala.clasif,"\n")
    cat("porcentaje.buena.clasif",porcentaje.buena.clasif,"\n")
    cat("porcentaje.mala.clasif",porcentaje.mala.clasif,"\n")



#   REPRESENTACIONES DEFINITIVAS DE RASGO Y RUIDO

#   A continuacion, calcularmos las distancias de las funciones

#   LISA individual, de los individuos clasificados finalmente

#   como rasgo y como ruido, por separado:

    distancias.ruido<-distancias.a.media.teorica[selRU]
    distancias.rasgo<-distancias.a.media.teorica[selRA]
```

```
#   Con la siguiente instruccion, se construyen 6 ventanas
#   donde ira colocando los graficos que calculemos:

    postscript("KULLBACK_Ruido_cuadrado_crculo_ forma_L_ruido_simulado_
    rasgo_simulado_.eps")

    par(mfrow=c(3,2))

#   Representacion de los puntos de rasgo originales:

    plot(rbind(rasgo),xlab="Rasgo original,coordenada x",
ylab="coordenada y",xlim=c(0,1),ylim=c(0,1),pch=".")

#   Representacion de la clasificacion de Rasgo:

    plot(data.rasgo.final,xlab="Rasgo clasificado,
coordenada x",ylab="coordenada y",
    xlim=c(0,1),ylim=c(0,1),pch=".")

#   Representacion de los puntos de ruido originales:

    plot(ruido,xlab="Ruido original,coordenada x",
ylab="coordenada y",pch=".")

#   Representacion de la clasificacion de Ruido:

    plot(data.ruido.final,xlab="Ruido clasificado,
coordenada x",ylab="coordenada y",pch=".")

#   Representacion de los puntos de ruido simulados:

    plot(ruido.simulacion,xlab="Ruido simulacion,coordenada x",
ylab="coordenada y",
    pch=".")

#   Representacion de los puntos de rasgo simulados:
```

```
    plot(rasgo.simulacion,xlab="Rasgo simulacion,coordenada x",
ylab="coordenada y",
    xlim=c(0,1),ylim=c(0,1),pch=".")

    dev.off()

result<-list(datos=data,vector.clasificacion=deltak,
matriz.resultado=matriz.resultado,densidad.ruido=fRU,densidad.rasgo=fRA,
num.ptos.ruido=nRU,num.ptos.rasgo=nRA,prob=p,
dists.de.originales.a.teorico=distancias.a.media.teorica) return(result)

}
--------------------------------------------------------------------------
# 7. LANZADERA DE LAS CLASIFICACIONES
# Valida para cualquier lanzamiento de uno, dos o tres rasgos, basta con
# anular las lineas correspondientes a los rasgos que no se desea incluir,
# e ir cambiando las cantidades de los puntos


ruido<-csr(polygrande,400) rasgo.C<-csr(polypeque,400)
#rasgo.L<-csr(nuestro.poly,600)
#poly.circulo<-funcion.para.circulo(centrox=0.8,centroy=0.8,radio=0.15,
npts=8000,poly=polygrande)

# Si es un rasgo: rasgo<-rasgo.c

# Si son dos rasgos:
#rasgo<-rbind(rasgo.C,poly.circulo)

#Si son tres rasgos:
#rasgo<-rbind(rasgo.C,rasgo.L,poly.circulo)

# circulo fijo, que tiene 354 puntos

clasifica<-SEM.LISA(data=rbind(ruido,rasgo), delta=0.10, a=1, b=1,
t=seq(0.005,0.5,len=100),
nptosruidosimulacion=800,nptosrasgosimulacion=50,poly=polygrande,
```

```
poly1=polypeque1, edge.correct=F, wrap = 0.1, convergance = 0.001)
-------------------------------------------------------------------------
# 8. LANZADERA DE CLASIFICACIONES PARA CASOS REALES CUYA IMAGEN
# PROVENGA DEL MATROX

# LANZADERA PARA IMAGEN PROCESADA, DE LA
# QUE TENGAMOS UN FICHERO .txt, DONDE HAYA
# TRES COLUMNAS, PRIMERA, COORDENADA X DEL
# PUNTO, SEGUNDA, COORDENADA Y, TERCERA, 0 o 1.
# LOS 1 CORRESPONDEN A BLANCO, POR TANTO NO IN-
# TERESA, LOS 0 A NEGRO, ES LO QUE INTERESA
# Previamente en los ficheros de datos
# que proporciona el matrox, hay que
# eliminar las comas entre las 3 columnas
# Definimos los poligonos donde se va a lanzar el
# rasgo y el ruido simulado:

polygrande<-cbind(c(0,1,1,0),c(0,0,1,1))
polypeque1<-cbind(c(0.1,0.2,0.2,0.1),c(0.8,0.8,0.9,0.9))

# Tomamos, pues, aquellos valores que son
# negros, como datos:

data.matrox<-matrix(scan("celulas_barcelona1.txt"),ncol=3,byrow=T)

sel.data<-data.matrox[,3]==0 data.ceros<-data.matrox[sel.data,c(1,2)]

clasifica<-SEM.LISA(data=data.ceros, delta=0.10, a=1, b=1,
t=seq(0.005,0.5,len=100),
nptosruidosimulacion=500,nptosrasgosimulacion=30,poly=polygrande,
poly1=polypeque1, edge.correct=F, wrap = 0.1, convergance = 0.001)



-------------------------------------------------------------------------
# 9. MODIFICACION DEL CODIGO FORTRAN PARA EL CALCULO DE FUNCIONES
# LISA, EN EL CASO DE IMAGENES REALES
```

```
      subroutine lisa(t,n,i,m,delta,a,b,x1,y1,x,y,g)
c      subroutina para el calculo de la densidad producto individual
(lisa) c      con correccion de borde dada por c_X(r))

       integer n,m,i
       real*8 t,x,y,g,d,delta,de,pi
       real*8 borde,w,suma,a,b,x1,y1
       dimension x(n),y(n),w(450000),t(450000)
       dimension suma(450000),g(450000),borde(450000)

       pi=3.141593d0
c      distancias euclideas de x1,y1 al resto

c         do 50 j=1,n c           if (j.ne.i) then c
d(i,j)=((x1-x(j))**2+(y1-y(j))**2)**(0.50d0) c           endif c50
continue


        de=delta
        do 55 k=1,m
          suma(k)=0.0d0
           do 70 j=1,n
            if (j.ne.i) then
            d=((x1-x(j))**2+(y1-y(j))**2)**(0.50d0)
             if (dabs(d-t(k)).ge.de) then
              w(j)=0.0d0
            else
              w(j)=(3.0d0/(4.0d0*de))*(1.0d0-(((d-t(k))**2)
     e     /(de**2)))
            endif
           else
             w(j)=0.0d0
           endif
       suma(k)=suma(k)+w(j)
70     continue
```

```
        borde(k)=(n-1)/(pi*(2.0d0)*t(k)*(a*b)*
     e     ((a*b)-(t(k)*((2*(a+b))-t(k)))/pi))
        g(k)=suma(k)*borde(k)
55      continue


        return
        end
```

```
-----------------------------------------------------------------------
# CODIGOS CAPITULO 6
-----------------------------------------------------------------------

# 1. CODIGO PARA OBTENER LA MATRIZ D DE DISTANCIAS DE MAHALANOBIS ENTRE
# LOS VECTORES DE FUNCIONES LISA INDIVIDUALES DE TODOS LOS PUNTOS

MAHALANOBIS.LISA<-function(data,delta,a,b,t,poly,d=2,edge.correct=T,wrap,
lintegral) {

delta.in<-delta

# CORRECCION DE BORDE TOROIDAL, PARA LOS DATOS:

if( (d==2)  && (edge.correct==T) ){ r1 <- diff(range(data[,1])) r2 <-
diff(range(data[,2])) tran2 <-
matrix(c(rep(0,n),rep(r2,n)),byrow=F,nrow=n) tran1 <-
matrix(c(rep(r1,n),rep(0,n)),byrow=F,nrow=n) aux.dat <-
rbind(data+tran1,data-tran1,data+tran2,data-tran2, data+tran1+tran2,
data-tran1+tran2, data-tran1-tran2, data+tran1-tran2) aux.dat <- aux.dat[
aux.dat[,1] < (max(data[,1]) +wrap*r1) & aux.dat[,1] > (min(data[,1])
-wrap*r1) & aux.dat[,2] < (max(data[,2]) +wrap*r2) & aux.dat[,2] >
(min(data[,2]) -wrap*r2), ] full.data<-rbind(data,aux.dat) } else{
full.data<-data }

data<-full.data
```

```
#n recogera la cantidad total de valores:
    n<-length(data[,1])
    n.ruido<-length(ruido[,1])
    n.rasgo<-length(rasgo[,1])

# Al numero de radios diferentes que tomamos,
# se le llamara kdistancias:

kdistancias<-length(t)


# Coordenadas espaciales de los individuos:

#xi<-data[individuo1,1]
#yi<-data[individuo1,2]
#xl<-data[individuo2,1]
#yl<-data[individuo2,2]

# CALCULO DE LAS FUNCIONES LISA DE CADA UNO DE LOS PTOS

funciones.lisa.de.todos.ptos<-matrix(0,nrow=kdistancias,ncol=n)
Yi.menos.Yl<-rep(0,kdistancias)

funciones.lisa.de.todos.ptos<-lisafunct(data,delta=delta.in,a,b,t,poly)

#Inicializando las matrices, con MATRIZ llamamos a la

# matriz de varianzas covarianzas para cada dos individuos

# que le llamo el i y el l, cuyas componentes seran la

# expresion 4.2 para la diagonal y la expresion 4.3 para

# el resto de entradas. Con d, llamamos a la matriz

# definitiva de distancias entre los vectores de
```

```
# funciones LISA correspondientes a los datos.

MATRIZ<-matrix(0,ncol=kdistancias,nrow=kdistancias)
MATRIZINV<-matrix(0,ncol=kdistancias,nrow=kdistancias)
d<-matrix(0,ncol=n,nrow=n)

# i y l, son dos individuos:

for (i in 1:n){ for (l in 1:n){

if (i==l) {d[i,l]<-0} else
        {
#   k y m son dos valores concretos de los radios

    kdistanciasa<-kdistancias-1
    for (k in 1:kdistanciasa){

    vdists42<-t[k]
    vdists43<-t[c(k,(k+1))]

#       Las siguientes expresiones
#       estan implementadas en fortran, en el archivo:
#       varianza-teorica-f4-2_s.so y covarianza-teorica-f4-3_s.so,
#       para lo cual, antes de lanzar el programa se ha de hacer un
#       dyn.load("varianza-teorica-f4-2_s.so") y un
#       dyn.load("covarianza-teorica-f4-3_s.so")
#       si es en cola tambien hay que escribir la instruccion:
#       source("funcion-R-para-f4-2.S")y
#       source("funcion-R-para-f4-3.S")

        expresion42<-varianza.teorica.lisa.f4.2(data=data,individuo1=i,
individuo2=l,vdists=vdists42,epsilon=delta,lintegral=lintegral,poly=poly)
        expresion43<-covarianza.teorica.lisa.f4.3(data=data,individuo1=i,
individuo2=l,vdists=vdists43,epsilon=delta,lintegral=lintegral,poly=poly)

        MATRIZ[k,k]<-expresion42
```

```
        MATRIZ[k,(k+1)]<-expresion43
        MATRIZ[(k+1),k]<-MATRIZ[k,(k+1)]

    } #cerrar el k

expresion42ult<-varianza.teorica.lisa.f4.2(data=data,individuo1=i,
individuo2=l,vdists=t[kdistancias],epsilon=delta,lintegral=lintegral,
poly=poly)
MATRIZ[kdistancias,kdistancias]<-expresion42ult

# Calculo del producto de matrices para la

# obtencion de D:

Yi.menos.Yl<-funciones.lisa.de.todos.ptos[,i]-
funciones.lisa.de.todos.ptos[,l]

MATRIZINV<-solve(MATRIZ)

d[i,l]<-t(Yi.menos.Yl)%*%MATRIZINV%*%Yi.menos.Yl

} #cerrar else de d[]

} #cerrar el l

} #cerrar el i

write(t(d),"matrizd1.txt",ncolumns=n,append=F)

write(t(data),"matrizdatos1.txt",ncolumns=2,append=F)

# REPRESENTACIONES

#   Con la siguiente instruccion, se construyen 4 ventanas
#   donde ira colocando los graficos que calculemos:

   postscript("Mahalanobis.eps")
```

```
   par(mfrow=c(2,2))

#   Representacion de los puntos de rasgo originales:

   plot(rasgo,xlab="Rasgo original,coordenada x",ylab="coordenada y",
xlim=c(0,1),ylim=c(0,1),pch=".")

#   Representacion de los puntos de ruido originales:

   plot(ruido,xlab="Ruido original,coordenada x",ylab="coordenada y",
xlim=c(0,1),ylim=c(0,1),pch=".")

#   Histograma de las distancias de rasgo:

   hist(d[,1][1:n.ruido],col=2,br=20,xlab="Distancia entre funciones
LISA de ruido",ylab="numero de individuos")

#   Histograma de las distancias de ruido:

   hist(d[,1][n.ruido+1:n.rasgo],col=2,br=20,xlab="Distancia entre
funciones LISA de rasgo",ylab="numero de individuos")

   dev.off()

result<-d return(result)

}
-----------------------------------------------------------------
# 2. LANZADERA PARA OBTENCION DE LA MATRIZ DE DISTANCIAS

# Este codigo lanza NNCLEAN-MAHALANOBIS

polygrande<-cbind(c(0,1,1,0),c(0,0,1,1))
polypeque<-cbind(c(0.4,0.6,0.6,0.4),c(0.2,0.2,0.6,0.6))

ruido<-csr(polygrande,100) rasgo.C<-csr(polypeque,100)
```

```
#rasgo.L<-csr(nuestro.poly,450)
#poly.circulo<-funcion.para.circulo(centrox=0.8,centroy=0.8,radio=0.15,npts=6000,
poly=polygrande)

# Si es un rasgo: rasgo<-rasgo.C

clasifica.MAHALANOBIS<-MAHALANOBIS.LISA(data=rbind(ruido,rasgo),
delta=0.10, a=1, b=1, t=seq(0.005,0.5,len=25),poly=polygrande,
edge.correct=F, wrap = 0.1, lintegral=22500)
-----------------------------------------------------------------------
# 3. CODIGOS PARA LA OBTENCION DE CLUSTER JERARQUICO Y  ESCALAMIENTO
MULTIDIMENSIONAL

#1. lectura de los datos

# Originales
datos100150<-matrix(matrizdatos2.txt"),ncol=2,byrow=T)

plot(datos100150,xlab="coordenada x",ylab="coordenada
y",pch=".",main="DATOS ORIGINALES")

# Distancias distancias100150<-matrix(scan(matrizd2.txt"),ncol=250)

#2. CLUSTER JERARQUICO

cluster.100150<-hclust(as.dist(distancias100150),method='ward')

# Convierte el resultado de hclust en formato dendrograma
cluster.100150<-as.dendrogram(cluster.100150)

# Corta el dendrograma en dos grupos
corte.en.dos.grupos<-cut(h=1e+7,x=cluster.100150)

# Selecciona de los datos originales, los dos grupos
ruido<-unlist(corte.en.dos.grupos$lower[[2]])
rasgo<-unlist(corte.en.dos.grupos$lower[[3]])
```

```
datos.rasgo.hclust<-datos100150[rasgo,]
datos.ruido.hclust<-datos100150[ruido,]

# Construye cuatro columnas, donde la tercera muestra un 1
si es rasgo en los datos originales, y un 0 si es un ruido.
 La cuarta columna, muestra los mismos valores, pero de la
clasificacion

var.hclust<-rep(1,250) var.hclust[unlist(a$lower[[3]])]<-0
datos.completos.hclust<-cbind(datos100150,c(seq(0,0,len=100),
seq(1,1,len=150)),var.hclust)

# Porcentajes de buena y mala clasificacion

conteo.mala.clasif<-0 conteo.buena.clasif<-0
n<-length(datos.completos.hclust[,1]) for(i in 1:n){
ifelse(datos.completos.hclust[i,3]==datos.completos.hclust[i,4],
conteo.buena.clasif<-
conteo.buena.clasif+1,conteo.mala.clasif<-conteo.mala.clasif+1) }
porcentaje.buena.clasif<-conteo.buena.clasif/100
porcentaje.mala.clasif<-conteo.mala.clasif/150

cat("conteo.buena.clasif",conteo.buena.clasif,"\n")
cat("conteo.mala.clasif",conteo.mala.clasif,"\n")
cat("porcentaje.buena.clasif",porcentaje.buena.clasif,"\n")
cat("porcentaje.mala.clasif",porcentaje.mala.clasif,"\n")

# .eps FINALES:

# CLUSTER JERARQUICO

postscript("100rasgo-150ruido_hclust.eps")
par(mfrow=c(3,2))

# Representacion de los DATOS originales:
plot(datos100150,xlab="coordenada x",ylab="coordenada
```

```
y",pch=".",main="DATOS ORIGINALES")

#   Representacion del DENDROGRAMA:
plot(cluster.100150,xlab="Datos
originales agrupados",ylab="Distancias de agrupacin",main="DENDROGRAMA
CLUSTER JERRQUICO",sub="MTODO WARD")

# Representacion Rama de Rasgo:
plot(corte.en.dos.grupos$lower[[2]],xlab="Datos rasgo
agrupados",ylab="Distancias de agrupacin",main="RAMA RASGO",sub="METODO
WARD")

#   Representacion Rama de Ruido:
plot(corte.en.dos.grupos$lower[[3]],xlab="Datos ruido
agrupados",ylab="Distancias de agrupacin",main="RAMA RUIDO",sub="METODO
WARD")

#   Representacion de los puntos de Rasgo Clasificados:
plot(datos.rasgo.hclust,xlim=c(0,1),ylim=c(0,1),pch=".",xlab="coordenada
x",ylab="coordenada y",main="RASGO CLASIFICADO")

#   Representacion de los puntos de Ruido Clasificados:
plot(datos.ruido.hclust,xlim=c(0,1),ylim=c(0,1),pch=".",xlab="coordenada
x",ylab="coordenada y",main="RUIDO CLASIFICADO") dev.off()

 #3. MULTIDIMENSIONAL SCALING A LOS DATOS.

cmd.100150<-cmdscale(distancias100150,k=2,eig=F,add=F,x.ret=F)

# Muestra la representacion del escalamiento multidimensional
plot(cmd.100150)

# En caso de outliers, eliminamos estos valores

#plot( cmd.100150[-c(214,199,149,171,117,101,168),])

# Marcas de ejes coordenados
```

```
#abline(h=0)

#abline(v=0)

# Seleccion de los valores representados selpos<-cmd.100150[,1]>=0
selneg<-cmd.100150[,1]<0

# Construye cuatro columnas, donde la tercera muestra  un 1 si es
# rasgo en los datos originales, y un 0 si es un ruido. La cuarta
# columna, muestra los mismos  valores, pero de la clasificacion

var.cmd<-rep(0,250) var.cmd<-ifelse(selpos=='TRUE',1,0)
datos.completos.cmd<-cbind(datos100150,c(seq(0,0,len=150),seq(1,1,
len=100)),var.cmd)

# Porcentajes de buena y mala clasificacion

conteo.mala.clasif<-0 conteo.buena.clasif<-0
n<-length(datos.completos.cmd[,1]) for(i in 1:n){
ifelse(datos.completos.cmd[i,3]==datos.completos.cmd[i,4],conteo.
buena.clasif<-
conteo.buena.clasif+1,conteo.mala.clasif<-conteo.mala.clasif+1) }
porcentaje.buena.clasif<-conteo.buena.clasif/150
porcentaje.mala.clasif<-conteo.mala.clasif/150

cat("conteo.buena.clasif",conteo.buena.clasif,"\n")
cat("conteo.mala.clasif",conteo.mala.clasif,"\n")
cat("porcentaje.buena.clasif",porcentaje.buena.clasif,"\n")
cat("porcentaje.mala.clasif",porcentaje.mala.clasif,"\n")

sel1<-datos.completos.cmd[,4]==1 sel0<-datos.completos.cmd[,4]==0

datos.rasgo.cmd<-datos.completos.cmd[sel1,c(1,2)]
datos.ruido.cmd<-datos.completos.cmd[sel0,c(1,2)]

# .eps FINALES:
```

```
# ESCALAMIENTO MULTIDIMENSIONAL

postscript("100rasgo-150ruido_cmd.eps") par(mfrow=c(2,2))

# Representacion de los DATOS originales:
plot(datos100150,xlab="coordenada x",ylab="coordenada
y",pch=".",main="DATOS ORIGINALES")

#   Representacion del ESCALAMIENTO MULTIDIMENSIONAL:
plot(cmd.100150[-c(214,199,149,171,117,101,169),],xlab="Nuevas coordenadas
x",ylab="Nuevas coordenadas y",main="ESCALAMIENTO MULTIDIMENSIONAL")
abline(h=0) abline(v=0)

#   Representacion de los puntos de Rasgo Clasificados:
plot(datos.rasgo.cmd,xlim=c(0,1),ylim=c(0,1),pch=".",xlab="coordenada
x",ylab="coordenada y",main="RASGO CLASIFICADO")

#   Representacion de los puntos de Ruido Clasificados:
plot(datos.ruido.cmd,xlim=c(0,1),ylim=c(0,1),pch=".",xlab="coordenada
x",ylab="coordenada y",main="RUIDO CLASIFICADO") dev.off()
```

Bibliografía

[1] Adams, J. D., Smith, M. O. & Johnson, P.E. (1986). Spectral mixture modeling: A new analysis of roch and soil types at the Viking Lander 1 site. *Journal of Geophysical Research,* **91**, 8098-8112

[2] Aitkin, M., Anderson, D. & Hinde, J. (1981). Statistical modelling of data on teaching styles (with Discussion). *Journal of the Royal Statistical Society, A* **144**, 419-461.

[3] Allard, D. (1995). Non-parametric maximum likelihood estimation of features in spatial point process using Voronoï tesselation. Technical Report 293, Department of Statistics, University of Washington.

[4] Allen, M.P. & Tildesley, D.J. (1987). *Computers Simulation of Liquids.* Oxford University Press, Oxford.

[5] Anselin, K. (1995). Local indicators of spatial association-LISA. *Geographical Analysis*, **27**, 93- 115.

[6] Baddeley, A.J. & Møller, J. (1989). Nearest-neighbour Markov point processes and random sets. *International Statistical Review*, **57**, 90-121.

[7] Banfield, J.D. & Raftery, A.E. (1992). Ice floe identification in satellite images using mathematical morphology an clustering about principal curves. *Journal of the American Statistical Association,* **87**, 7-16.

[8] Banfield, J.D. & Raftery, A.E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**, 803-849.

[9] Basseville, M (1996). Information: entropies, divergences et moyennes. Publication Interne 1020, Institut de Recherche en Informatique et systémes aléatoires.*Irisa*

[10] Bensmail, H., Celeux, G., Raftery, A.E. & Robert, C.P. (1994). Inference for model based cluster analysis. Technical Report 285, Department of Statistics, University of Washington.

[11] Bensmail, H., Celeux, G., Raftery, A.E. & Robert, C.P. (1997). Inference in model-based cluster analysis. *Statistics and Computing*, **7**, 1-10.

[12] Berger, J. O. & Sellke, T. (1987). Testing a point null hypothesis: The irreconcilability of P-values and evidence. *Journal of the American Statistical Association,* **82**, 112-122.

[13] Besag, J.E. (1978). Some methods of statistical analysis for spatial data. *Bulletin of the Institute of International Statistics*, **47**, 77-92.

[14] Besag, J., Milne, R., & Zachary, S. (1982). Point process limits of lattice processes. *Journal of Applied Probability*, **19**, 210-216.

[15] Besag, J. & Green, P.J. (1993). Spatial statistics and Bayesian computation (with discussion). *Journal Royal of the Statistical Society, B* **55**, 25-37.

[16] Besag, J., Green, P. J., Higdon, D. & Mengersen, K. (1994). Bayesian computation and stochastic systems (with discussion and rejoinder). *Statistical Science* **10**, 3-66.

[17] Brant, L. & Cressie N. (1997). Analysis of Spatial Point Patterns using Bundles of Product Density LISA Functions. Preprint No. 97-3.

[18] Byers, S. & Raftery, A.E. (1998). Nearest neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, **93**, 577.

[19] Byers, S. & Raftery, A.E. (1997). Bayesian Estimation and Segmentation of Spatial Point Processes using Voronoï Tilings, Technical Report, number 32.

[20] Celeux, G. & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, **14**, 315-332.

[21] Celeux,G. & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, **28**, 781-793.

[22] Chalmond, B. (1989). An iterative gibbsian technique for reconstruction of many images. *Pattern Regognition*, **22(6)**,747-761

[23] Chernoff, H. & Rubin, H. (1954). The estimation of the location of a discontinuity in density, *Proceedings of the Third Berkeley symposium on mathematical statistics and probability,* University of California Press, 19-37.

[24] Cox, D.R. (1975). Partial likelihood, *Biometrika*, **62**, 269-276.

[25] Creutz, M. (1979). Confinement and critical dimensionality of space-time. *Physical Review Letters*, **43**, 553-556.

[26] Cressie, N.A.C. (1993). *Statistics for Spatial Data.* Wiley, New York.

[27] Cressie N. & Brant, L.B. (1997). Patterns in Spatial Point Locations. Local Indicators of Spatial Association in a Minefield with clutter. *Naval Research Logistics*, **48**, 333-347.

[28] Cressie N. & Brant, L.B. (2001). Analysis of spatial Point Patterns using bundles of product LISA function. *Journal of Agricultural, Biological, and Environmental Statistics*, **6**, 118-135.

[29] Dasgupta, A. & Raftery, A.E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, **93**, 294-302.

[30] Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood for incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society*, B **39**, 1-38.

[31] Diebolt, J. & Robert, C.P. (1994). Estimation of Finite Mixture Distributions through Bayesian Sampling. *Journal of the Royal Statistical Society, B*, **56** 363-375.

[32] Daley, D.J. & Vere-Jones, D. (1988). *An Introduction to the Theory of Point Processes.* Springer-Verlag, Berlin.

[33] Diggle, P.J. (1986). Parametric and non-parametric estimation for pairwise interaction point processes. *Proceedings of the 1st World Congress of the Bernoulli Society.*

[34] Diggle, P.J., Fiksel, T., Ogata Y., Stoyan, D. & Tanemura, M. (1994). On parameter estimation for pairwise interaction point processes. *International Statistical Review*, **62**, 99-117.

[35] Diggle, P.J. (2003). *Statistical Analysis of Spatial Point Patterns*. London: Academic Press.

[36] Fraley, C. & Raftery, A.E. (1998). How many clusters? Which clustering method?-Answers via model-based cluster analysis. *The Computer Journal*, **41**, 578-588.

[37] Fraley, C. & Raftery, A.E. (1999). MCLUST: software for model-based cluster and discriminant analysis. *Technical report*, 342, Department of Statistics, University of Washington.

[38] Friedman, H.P. & Rubin, J. (1967). On some invariant criteria for grouping data. *Journal of the American Statistical Association*, **62**, 1159-1178.

[39] Gelfand, A.E. & Carlin, B.P. (1993). Maximum likelihood estimation for constrained or missing-data models. *Canadian Journal of Statistics*, **21**, 303-311.

[40] Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the bayesians restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721-741.

[41] Geyer, C.J. (1992). Practical Markov chain Monte Carlo (with discussion). *Statistical Science*, **8**, 473-511.

[42] Geyer, C.J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society,* ***B*** **56**, 261-274.

[43] Geyer, C.J. & Møller, J. (1994). Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, **21**, 359-373.

[44] Geyer, C.J. & Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to pedigree analysis. *Journal of the American Statistical Association*, **90**, 909-920.

[45] Geyer, C.J. (1999). Likelihood inference for spatial point processes. In Barndorff-Nielsen. O.E., Kendall, W.S., & van Lieshout, M.N.M., editors *Stochastic Geometry: Likelihood and Computation* Chapman & Hall/CRC, London, Boca Raton, 79-140.

[46] Glotzel, E. (1981). Time reversible and Gibbsian point processes, I: Markovian spatial birth and death processes on a general phase space. *Mathematische Nachrichten*, **102**, 217-222.

[47] Gower, J.C. (1967). A comparison of some methods of cluster analysis. *Biometrics*, **23**, 623-637.

[48] Gupta, A. K., Taneja, V.S. & Janardan, K.G. (1986). Estimation of an arbitrary mixture of uniform models. *Metron*, **44**, 247-256.

[49] Green, P. J. & R. Sibson (1977). Computing dirichlet tesselations in the plane. *The Computer Journal,* **21**, 168-173.

[50] Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97-109.

[51] Hathaway, R.J. (1986). Another interpretation of the EM algorithm for misture distributions. *Journal of Statistics & Probability Letters,* **4**, 53-56.

[52] Heinrich, L., Liebscher, E. (1997). Strong convergence of kernel estimators for product densities of absolutely regular point processes. *Journal of Nonparametric Statistics*, **8**, 65-96.

[53] Hinde, A.L. & Miles, R. E. (1980). Monte Carlo estimates of the distributions of the random polygons of the Voronoï tessellations vith respect to a Poisson process. *Journal of Statistical Computation and Simulation*, **10**, 205-223.

[54] Illian, J. (2004). Ecological communities - spatial point process modelling of highly complex systems. *Proceedings of the fifth French-Danish Workshop on Spatial Statistics and Image Analysis in Biology.*

[55] Jensen, J.L. & Møller, J. (1991). Pseudo-likelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, **3**, 445-461.

[56] Kass, R. E. & Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, **90**, 773-795.

[57] Kaufman, L. & Rousseeuw, P.J. (1990). Finding groups in data. An introduction to cluster analysis. John Wiley & Sons.

[58] Kelly, F.P. & Ripley, B.D. (1976). A note on Strauss model for clustering. *Biometrika*, **63**, 357-360.

[59] Lance, G.N. & Williams, W.T. (1966a). Computer programs for hierarchical polythetic classification. *Computer Journal*, **9**, 60-64.

[60] Lance, G.N. & Williams, W.T. (1966b). A general theory of classificatory sorting strategies: 1. Hierarchical systems. Computer programs for hierarchical polythetic classification. *Computer Journal*, **9**, 373-380.

[61] Lavine, M. & West, M. (1992). A Bayesian method for classification and discrimination. *The Canadian Journal of Statistics*, **20**, 451-461.

[62] Marinari, E. & Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters*, **19**, 451-458.

[63] Marriott, F.H.C. (1975). Separating mixtures of normal distributions. *Biometrics*, **31**, 767-769.

[64] Mase, S. (1992). Uniform LAN condition of planar Gibbsian point processe and optimality of maximum likelihood estimators of soft-core potential functions. *Probability Theory Related Fields*, **92**, 51-57.

[65] Mateu, J. (1995). *Procesos puntuales: modelización y estimación de los modelos Markov.* Tesis Licenciatura. Universitat de Valencia.

[66] Mateu, J. & Lorenzo, G. (2000). Searching for regions of varying spatial intensities in point processes. *Questiio*, **26**, 61-85.

[67] Mateu, J. & Lorenzo, G. (2001). Local second-order characteristics for spatial point-based features detection *Pattern Recognition and Image Analysis. Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis. (Eds.) J. Salvador & Filiberto Pla*, Castellón, **2**, 25-30.

[68] Mateu, J. & Lorenzo, G. (2002). Detección de rasgos en imágenes binarias mediante procesos puntuales espaciales marcados. *Congreso Nacional de Estadística e I.O.*, Vigo, 473-74.

[69] Matrox Electronic Systems Ltd (1991), 1055, boul. St-Régis, Dorval (Québec), Canada H9P 2T4, www.matrox.com .

[70] Matérn, B. (1986). *Spatial Variation.* Lecture Notes in Statistics No. 36, Springer-Verlag, Berlin.

[71] McLachlan, G.J. & Basford, K.E. (1988). *Misture models: Inference and applications to clustering*. Marcel Dekker.

[72] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087-1092.

[73] Milligan, G.W. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, **50**, 159-179.

[74] Møller, J. (1989). On the rate of convergence of spatial birth-and-death processes. *Annals of the Institute Statistical Mathematics*, **41**, 565-581.

[75] Møller, J. (1994). Contribution to the discussion of N.L. Hjort & H. Omre: Topics in spatial statistics, *Scandinavian Journal of Statistics*, **21**, 346-349.

[76] Møller, J. (1996). Markov chain Monte Carlo and spatial points processes. In O.E. Barndorff-Nielsen, W.S. Kendall and M.N.M. van Lieshout (Eds), *Stochastic Geometry: Likelihood and Computation*, Monographs on Statistics ans Applied Probability **80**, Chapman & Hall/CRC, Boca Raton, 141-172.

[77] Murtagh, F. (1985). Multidimensional clustering algorithms, *CompStat Lectures*, **4**. Physica-Verlag.

[78] Neyman J. & Scott E.L. (1956). Struggle for existence. The Tribolium model: Biological and statistical aspects. *Proceedings of the Third Berkeley symposium on mathematical statistics and probability,* University of California Press, **4**, 41-79.

[79] Osher, J. & Stoyan, D. (1981). On the second-order and orientation analysis of planar stationary point processes. *Biometrical Journal*, **23**, 523-533.

[80] Peskun, P. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, **60**, 607-612.

[81] Preston, C.J. (1977). Spatial birth-and-death processes. *Bulletin of the International Statistical Institute*, **46**, 371-391.

[82] Redner, R.A. & Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM* Review, **26**, 195-239.

[83] Ripley, B.D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, **13**, 255-266.

[84] Ripley, B.D. (1977). Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society, B* **39**, 172-212.

[85] Ripley, B.D. & Kelly, F.P. (1977). Markov point processes. *Journal of the London Mathematical Society*, **15**, 188-192.

[86] Ripley, B.D. (1979). Simulating spatial patterns: Dependent samples from a multivariate density. *Journal of the Royal Statistical Society, C* **28**, 109-112.

[87] Ripley, B.D. (1987). *Stochastic Simulation*. John Wiley & Sons, New York.

[88] Ripley, B.D. (1987). *Statistical Inference for Spatial Processes. Cambridge University Press*.

[89] Rowlinson, B.S. & Diggle, P.J. (1993). Splancs: spatial point pattern analysis code in Splus. *Computers and Geosciences*, **19**, 627-55.

[90] Smith, A. & Spiegelhalter, D. (1980). Bayes factors and choice criteria for linear models. *Journal of the Royal Statistical Society*, A **42**, 213-220.

[91] Smith, A.F.M. & Roberts, G.O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, B* **55**, 3-23.

[92] Sokal, R.R. & Michener, C.D. (1958). A statistical method for evaluating systematic relationships. *Univ. Kansas Science Bulletin*, **38**, 1409-1438.

[93] Sokal, R.R. & Sneath, P.H.A. (1963). *Principles of numerical taxonomy*. Freeman, San Francisco.

[94] Stacy, E.W. (1962). A generalization of the gamma distribution. *Annals of Mathematical Statistics*, **33**, 1187-1192.

[95] Stoica, R. S., Descombes, X., Van Lieshout, M. N. M. & Zerubia, J. *An application of marked point processes to the extraction of linear networks from images*. WIT Press, Southampton, UK.

[96] Stoyan, D. (1987). Statistical analysis of spatial point processes: a soft-core model and cross-correlations of marks. *Biometrical Journal*, **29**, 971-980.

[97] Stoyan, D., Bertram, U. & Wendrock, H. (1993). Estimation variances for estimators of product densities and pair correlation functions of planar point processes. *Annals of the Institute of Statistical Mathematics*, **45**, 211-221.

[98] Stoyan, D. & Stoyan, H. (1994). *Fractals, Random Shapes And Point Fields*. John Wiley & Sons, Chichester.

[99] Stoyan, D., Kendall, W.S. & Mecke, J (1995). *Stocastic Geometry and its Applications*. Wiley, New York

[100] Stoyan, D. & Stoyan, H. (2000). Improving ratio estimators of second order point process characteristics. *Scandinavian Journal of Statistics*, **27**, 641-656.

[101] Strauss, D.J. (1975). A model for clustering. *Biometrika*, **62**, 467-475.

[102] Suomela, P. (1976). Construction of Nearest Neighbour Systems. *Annales Academiae Scentiarum Fennicae Ser. A. I. Mathematica, Disertationes* **10**. Department of Mathematics, University of Helsinki.

[103] Tierney, L. & Kadane, J.B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, **81**, 82-86.

[104] Upton, G.J.G. & Fingleton, B.(1985). *Spatial Data Analysis by Example, Vol. 1*. Chichester: Wiley.

[105] Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association*, **58**, 236-244.

[106] Tierney, L. (1994). Markov chain for exploring posterior distributions. *Annals of Statistics*, **22**, 1701-1728.

[107] Winkler, G. (1991). *Image Analysis, Random Fields and Monte Carlo Methods*. Applications of Mathematics. Springer, Berlin.

[108] Wolfe, J. H. (1970). Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, **5**, 329-350.

[109] Younes, L. (1989). Parametric inference for imperfectly observed gibbsian fields. *Probability Thyeory and Related Fields,* **82**, 625-645.