

UNIVERSIDAD NACIONAL DE CÓRDOBA

TESIS DOCTORAL

Reconocimiento de patrones en campos
aleatorios de Markov mediante modelos
bayesianos para la agricultura de precisión

Autor:

Josef BAUMGARTNER

Director:

Dr. Julián PUCHETA

*Tesis presentada en cumplimiento parcial de los requisitos
para el grado de Doctor en Ciencias de la Ingeniería*

en el

Laboratorio de Investigación Matemática Aplicada a Control
Facultad de Ciencias Exactas, Físicas y Naturales

3 de abril de 2016

Resumen

Reconocimiento de patrones en campos aleatorios de Markov mediante modelos bayesianos para la agricultura de precisión

de Josef BAUMGARTNER

La agricultura de precisión sostiene que cada lote posee una variabilidad espacial y por ende no es óptimo aplicar un tratamiento uniforme. Para identificar y cuantificar esta variabilidad espacial en un lote o una zona se busca encontrar patrones estadísticos usando modelos matemáticos destinados a procesar los datos observados.

Se suponen que la variabilidad espacial de un lote puede ser modelada como campo aleatorio de Markov que – de acuerdo al teorema de Hammersley-Clifford – es equivalente a un campo aleatorio de Gibbs. Por lo tanto, el enfoque principal de esta tesis es el desarrollo de novedosos modelos causales de Markov y modelos de Potts basados en la distribución de Gibbs.

Como algoritmo de referencia de modelos causales se usa el *Path Constrained Viterbi Training* (PCVT) mientras que el método llamado *Iterated Conditional Modes* (ICM) se utiliza como referencia de un modelo de Potts. Cabe destacar que los algoritmos PCVT y ICM son modelos universales que no fueron diseñado específicamente para la agricultura de precisión. Entonces, el objetivo de esta tesis es adaptar las suposiciones de los dos algoritmos a las necesidades particulares de la agricultura de precisión pudiendo así originar nuevos algoritmos con un mejor desempeño.

El modelo causal de Markov desarrollado en esta tesis se llama *Complete Enumeration Propagation* (CEP). La gran ventaja de este modelo es que, a diferencia del PCVT, no necesita descartar la mayoría de los posibles estados ocultos a priori para ser tratable. A parte se deriva un nuevo modelo de Potts denominado *Sequential Band Merging* (SBM) que permite el uso de cualquier función de densidad como función de emisión. Otra ventaja que tiene SBM frente a ICM es que trabaja con vecindades adaptativas que se ajustan “online” en cada iteración.

Además, se extiende el esquema teórico de SBM para obtener un novedoso algoritmo de fusión de datos llamado *Markov Data Fusion* (MDF). Este algoritmo es capaz de procesar datos de distintos sensores – por ejemplo, imágenes de teledetección y mediciones locales – para obtener un mapa de variabilidad unificado.

Los datos usados en este trabajo son mayormente imágenes de teledetección pasiva y activa, pero también mediciones *in situ*. En concreto, se utilizan sensores de humedad

de suelo, un medidor de fotosíntesis y un novedoso medidor de potencial de tallo que fue desarrollado en esta tesis.

Para comparar los resultados de los algoritmos estudiados se utiliza el coeficiente de κ . Se observa que, en la gran mayoría de los casos los algoritmos desarrollados en este trabajo tienen un mejor desempeño que los algoritmos de referencia. Expresado en números se puede constatar que el coeficiente de κ de CEP y SBM es hasta un 12,63% más alto que los algoritmos de referencia. En el caso de fusión de datos, MDF supera a sus competidores con un κ de hasta un 29,70% más alto.

Abstract

Pattern Recognition with Bayesian Models in Markov Random Fields for Precision Farming

by Josef BAUMGARTNER

Precision agriculture states that every agricultural field has a spatial variability and therefore it is suboptimal to apply a unified treatment to the whole field. To identify the spatial variability of a field, mathematical models are used to find statistical patterns in the observed data.

We assume that the spatial variability of a field can be described by a Markov Random Field, which according to the Hammersley-Clifford Theorem, is equal to a Gibbs Random Field. Hence it is the goal of this work to derive new causal Markov models as well as Potts models which follow the Gibbs distribution.

As a benchmark algorithm for causal models we use the so called *Path Constrained Viterbi Training* (PCVT) while *Iterated Conditional Modes* (ICM) is a classical realization of the Potts model. Note, that PCVT and ICM are universal models that were not designed specifically for precision agriculture. Hence the objective of this thesis is to adapt the assumptions of the two algorithms to the particular needs of precision agriculture in order to derive new algorithms with better performance.

In this work we develop a new causal Markov model called *Complete Enumeration Propagation* (CEP). The huge advantage of CEP over PCVT is that CEP does not need to discard most of the possible states a priori in order to be feasible. As a representation of the Potts model the so called *Sequential Band Merging* (SBM) is derived. Compared with ICM, this model comes with two important improvements. First, any admissible probability function is allowed, even when dealing with multidimensional data. Second, the size of the neighborhood is estimated “online” after each iteration.

Furthermore, we extend the theoretical scheme of SBM to derive a multisensor data fusion algorithm called *Markov Data Fusion* (MDF). The idea behind MDF is to process different kinds of agricultural data – like satellite images or local measurements – in order to obtain a unified variability map.

Most of the data in this work comes from active and passive remote sensing. Besides that, the data of three ground sensors is used as well. The three sensors measure soil humidity, photosynthesis and the electrical potential of the stem. The last sensor was developed during this thesis.

To evaluate and compare the studied algorithms we use Cohen's κ coefficient. We observe that in most cases CEP and SBM outperform their competitors by achieving up to 12,63 % higher values of κ . In the case of MDF the difference is even bigger with up to 29,70 % higher κ coefficients.

Zusammenfassung

Mustererkennung in Markov Random Fields mittels Bayes'scher Modelle für die Präzisionslandwirtschaft

von Josef BAUMGARTNER

Die Präzisionslandwirtschaft beruht auf der Annahme, dass jedes landwirtschaftliche Feld heterogen ist und dass deshalb eine einheitliche Bewirtschaftung nicht optimal ist. Um die Heterogenität eines Feldes zu erfassen, werden in dieser Arbeit mathematische Modelle entwickelt die mit Hilfe von Messdaten statistische Muster finden.

Es wird grundsätzlich davon ausgegangen, dass ein landwirtschaftliches Feld als Markov Random Field (MRF) modelliert werden kann. Laut dem Hammersley-Clifford Theorem ist ein MRF immer auch ein Gibbs Random Field (GRF). Folglich liegt der Schwerpunkt dieser Arbeit in der Herleitung von kausalen Markov Modellen sowie von Potts Modellen die sich durch eine Gibbs Verteilung beschreiben lassen.

Der Standard Algorithmus für kausale Markov Modelle ist das sogenannte *Path Constrained Viterbi Training* (PCVT) während *Iterated Conditional Modes* (ICM) ein klassisches Potts Modell ist. Dabei sind sowohl PCVT als auch ICM universelle Modelle die nicht speziell für die Präzisionslandwirtschaft entwickelt wurden. Deshalb ist es das Ziel dieser Arbeit die theoretischen Annahmen beider Modelle an die Anforderungen der Präzisionslandwirtschaft anzupassen um neue, geeignetere Modelle herzuleiten.

Im Falle der kausalen Markov Modelle wird ein neuer Algorithmus namens *Complete Enumeration Propagation* (CEP) entwickelt. Der große Vorteil von CEP gegenüber PCVT ist, dass keine Vorauswahl an möglichen Zuständen nötig ist und der Rechenaufwand trotzdem überschaubar bleibt. Außerdem wird ein neues Potts Modell hergeleitet, dass selbst für multidimensionale Messdaten den Einsatz von beliebigen Wahrscheinlichkeitsfunktionen zulässt. Ein weiterer Vorteil des sogenannten *Sequential Band Merging* (SBM) gegenüber ICM ist, dass nach jeder Iteration die Anzahl an benachbarten Knoten neu angepasst wird.

Als Verallgemeinerung des SBM wird ein neuer Algorithmus namens *Markov Data Fusion* (MDF) beschrieben, der die Messdaten von verschiedenen Sensoren verarbeiten kann. Das Ziel dabei ist es mit Hilfe von Fernerkundungsdaten und lokalen Messungen zu einer einheitlichen Beschreibung eines landwirtschaftlichen Feldes zu gelangen.

Die meisten Messdaten dieser Arbeit sind passive und aktive Satellitenbilder, aber auch *in situ* Messungen. Beispiele hierfür sind die Feuchtigkeit des Bodes, die Fotosynthese

einer Pflanze oder der elektrische Widerstand des Stängels einer Pflanze, der mittels einem in dieser Arbeit entwickelten Geräts gemessen werden kann.

Um die Ergebnisse der verschiedenen Algorithmen vergleichen zu können wird Cohen's κ Koeffizient verwendet. In den allermeisten Experimenten erzielen die in dieser Arbeit entwickelten Algorithmen deutlich höhere κ Koeffizienten als die Standard-Algorithmen. So liegen die κ Werte von CEP und SBM bis zu über 12,63 % über denen ihrer Konkurrenten. MDF erreicht sogar bis zu 29,70 % höhere κ Werte als die Vergleichsalgorithmen.

A Vero y a mi familia

Agradecimientos

¡Fue un largo camino desde los inicios de mi tesis doctoral hasta su finalización y no hubiese logrado este objetivo sin la gran ayuda que recibí de todos ustedes!

Por el lado profesional, quiero agradecer profundamente a mi director Julián Pucheta, quien me hizo un lugar en Laboratorio de Investigación Matemática Aplicada a Control (LIMAC). Siempre estuvo presente para analizar, discutir y resolver todos los obstáculos que aparecieron en los últimos años, ya sean de origen teórico, práctico o burocrático. También quiero agradecer a mi co-director Marcelo Scavuzzo, quien aportó muchísima experiencia, conocimiento, contactos y quien me facilitó muchos datos reales que sin él hubiese sido muy difíciles de conseguir. Otro profesor muy importante para mí fue Ladislao Mathe. Como jefe de cátedra me impresionó con su compromiso educativo y su conocimiento como profesional. De la misma manera quiero agradecer a mi comisión asesora, formado por los doctores Oscar Bustos y Víctor Sauchelli, por sus comentarios y sugerencias. También debo un gran abrazo a mucha gente en la Facultad de Ciencias Exactas, Físicas y Naturales (FCEFYN). En primer lugar, quiero agradecer a mis compañeros Cristian Rodríguez Rivero, Santiago Bravo, Valeria Rulloni (todavía te debo un chocolate :-)) y Guillermo Riva. Además, mando un fuerte abrazo de gol a San Juan para mi compañero Javier Gimenez, disfruté muchísimo los trabajos que hicimos juntos y jamás me olvidaré de su conocimiento teórico y su forma precisa de trabajar. También quiero agradecer a Darío Aguilera por su gran ayuda en el desarrollo del medidor del potencial de tallo, a Georgina Flesia (FAMAF) por su apoyo en varios artículos y a Sergio Laboret (FCEFYN) por su conocimiento electrónico. ¡Por último, digo gracias a la Universidad Nacional de Córdoba, a SECyT, a la FCEFYN, a FAMAF, a CONICET y a CONAE, por las oportunidades que me dieron y las puertas que abrieron! Espero no haberlos defraudado.

¡Por el lado personal debo todo a vos Vero! ¡Gracias por estar siempre a mi lado y sobre todo gracias por nuestros hijos! También quiero agradecer muy profundamente a mi familia, que, a pesar de estar a miles de kilómetros, me apoyó incondicionalmente de todas las formas posibles. Además, agradezco a la familia de mi esposa por su atención, presencia y disposición durante todos estos años y a mi compatriota Stephan Kruber por ser un fiel compañero tan lejos de casa. ¡Hablando de compañeros, muchas gracias a los monos y a la vecindad por hacerme sentir como si hubiese vivido toda mi vida en Córdoba! ¡Espero recibirlos a todos por mis pagos algún día!

Índice general

Resumen	II
Abstract	IV
Zusammenfassung	VI
Agradecimientos	IX
Índice general	X
Abreviaturas	XIV
1. Introducción	1
1.1. Agricultura de precisión	1
1.2. Minería de datos	2
1.3. Motivación, desafíos y objetivos	3
1.4. Metodología	4
1.5. Contribuciones de la tesis	5
1.6. Organización de la tesis	6
2. Campos aleatorios de Markov	8
2.1. Fundamentos teóricos	9
2.2. Nomenclatura	10
2.3. Modelos teóricos	10
2.3.1. Modelo causal	11
2.3.2. Modelo de Potts	12
2.4. Aplicaciones, limitaciones y desafíos	13
3. Modelos ocultos de Markov en dos dimensiones	15
3.1. Esquema general	15
3.2. Teoría exacta	20
3.3. Problemas computacionales	23
3.4. Estimación de parámetros	24
3.4.1. Mapa inicial de estados	24
3.4.2. Probabilidades de transición	24
3.4.3. Funciones de emisión	25
3.5. Decodificadores	26
3.5.1. Path Constrained Viterbi Training	26
3.5.2. Complete Enumeration Propagation	32

3.6. Funciones de emisión	37
3.6.1. Teorema de Bayes	39
3.6.2. Test de Kolmogorov-Smirnov	39
3.6.3. Divergencia de Kullback-Leibler	39
3.7. Comparación de algoritmos	40
4. Realizaciones del modelo Potts	42
4.1. Definición general	42
4.2. Iterated Conditional Modes	43
4.3. Sequential Band Merging	47
4.3.1. Motivación	48
4.3.2. Esquema general	49
4.3.3. Probabilidades marginales	50
4.3.4. Aplicación del Bilateral Filter	51
4.3.5. Clasificación	56
4.4. Comparación de algoritmos	61
5. Fusión de datos	64
5.1. Introducción	64
5.2. Estado del arte y desafíos	66
5.3. Sensores, datos y objetivos	68
5.4. Algoritmos estándar	70
5.4.1. K-medias	71
5.4.2. Mezcla de Gaussianas	72
5.4.3. Agrupación Jerárquica	73
5.5. Makorv Data Fusion	75
5.6. Comparación de algoritmos	78
6. Sensores para agricultura de precisión	81
6.1. Teledetección	81
6.1.1. Teledetección pasiva	82
6.1.2. Teledetección activa	83
6.1.3. Imágenes satelitales vs. imágenes aéreas	83
6.2. Sensores locales	84
6.2.1. Humedad de suelo	84
6.2.2. Fotosíntesis	85
6.2.3. Potencial de tallo	86
7. Resultados experimentales	89
7.1. Parámetros y coeficientes	90
7.1.1. Parámetros de los algoritmos	90
7.1.2. El coeficiente κ de Cohen	91
7.2. Imágenes de teledetección	92
7.2.1. Imágenes ópticas (teledetección pasiva)	92
7.2.2. Imágenes de radar (teledetección activa)	101
7.3. Imágenes de referencia	104
7.3.1. Imagen rayo X de una rata	105
7.3.2. Logo de la UTN	108

7.3.3. Imágenes simuladas	109
7.3.4. Imagen estándar	111
7.4. Fusión de datos	114
7.4.1. Datos simulados	115
7.4.2. Datos reales	115
8. Evaluación final	124
8.1. Resumen de la tesis	124
8.2. Discusión de los resultados experimentales	125
8.3. Conclusiones	128
8.4. Futuros trabajos	130
A. Teorema de Hammersley-Clifford	133
B. Distribuciones de probabilidad	137
Bibliografía	139

Abreviaturas

Capítulo 1

CONAE	Comisión Nacional de Actividades Espaciales
EM	Expectation-Maximization
IEEE	Institute of Electrical and Electronics Engineers
HMM	Hidden Markov Models

Capítulo 3

CEP	Complete Enumeration Propagation
KL	Divergencia de Kullback-Leibler
KS	Test de Kolmogorov-Smirnov
PCVT	Path Constrained Viterbi Training

Capítulo 4

BF	Bilateral Filter
ICM	Iterated Conditional Modes
SA	Simulated Annealing
SBM	Sequential Band Merging

Capítulo 5

AJ	Agrupación Jerárquica
DAI-DAO	Data In - Data Out
DAI-FEO	Data In - Feature Out
DEI-DEO	Decision In - Decision Out
FEI-DEO	Feature In - Decision Out
FEI-FEO	Feature In - Feature Out

KM	K -medias
MDF	Markov D ata F usion
MG	Mezcla de G aussianas

Capítulo 6

AVIRIS	A irborne V isible/ I nfrared I maging S pectrometer
OLI	O perational L and I mager
SAR	S ynthetic- A perture R adar
TIRS	T hermal I nfra R ed S ensor

Capítulo 7

ETM+	E nhanced T hematic M apper P lus
HH	Polarización H orizontal- H orizontal
MV	M áxima V erosimilitud
UTN	U niversidad T ecnológica N acional
VV	Polarización V ertical- V ertical

Capítulo 1

Introducción

Vivimos en un mundo donde 25.000 personas mueren cada día por razones relacionadas al hambre [1]. Sin embargo, la población mundial está creciendo rápidamente. Hay estudios que predicen que en el año 2050 van a vivir 9.6 mil millones personas en la tierra (un aumento del 33% con respecto al año 2015) [2]. La pregunta es: ¿Cómo alimentar al mundo si los recursos claves como el agua dulce y las tierras fértiles son limitados [3,4]? Frente a este escenario es inevitable mejorar el manejo de los recursos con tecnologías modernas [5].

En esta tesis se desarrollan novedosos modelos matemáticos capaces de analizar e interpretar datos relacionados a la agricultura. Estos datos pueden ser macroscópicos, como imágenes satelitales o microscópicos, como sensores de humedad de suelo. El objetivo es brindarle al agricultor una herramienta científica capaz de mejorar la rentabilidad y la sustentabilidad de un lote.

1.1. Agricultura de precisión

Durante miles de años, civilizaciones como el imperio romano dependían fuertemente de su agricultura [6]. Recién a fines del siglo XVIII la agricultura europea hizo un gran salto estimulado por la revolución agrícola británica [7]. Desde entonces, la agricultura supo aprovechar avances tecnológicos como nuevas cosechadoras, fertilizantes, etc. [8] pero no hubo un cambio de paradigma hasta finales de los años 80. Recién a partir la liberación del *Global Positioning System* (GPS) para usos civiles y de la utilización

maquinas avanzadas que permitían aplicar químicos con tasas variables, se empezó a desarrollar lo que hoy conocemos como agricultura de precisión [9].

Si bien en los últimos 25 años la agricultura de precisión ha pasado por ciertas adaptaciones – causadas por ejemplo por avances computacionales [10] – la hipótesis principal sigue siendo la misma: aplicar un tratamiento uniforme a un lote no es lo óptimo [11]. Por lo tanto, la base fundamental de la agricultura de precisión es conocer la variabilidad espacial y temporal de un lote [9]. Para eso, es necesario combinar datos hiperespectrales provenientes de teledetección y de mediciones locales *in situ* en un modelo estadístico [12, 13].

A pesar de los grandes avances de los modelos estadísticos relativos a la agricultura de precisión – tantos generativos como discriminativos [14] – todavía hay mucho espacio para mejoras. Los desafíos matemáticos más urgentes son: mejorar la detección de variabilidad espacial y temporal [15], la incorporación de datos masivos [16] y la implementación de sistemas de soporte de decisiones [17].

1.2. Minería de datos

El termino *minería de datos* fue introducido por Gregory Piatetsky-Shapiro en 1989 [18]. En general, *minería de datos* se refiere a métodos y tecnologías realizados en software para extraer información oculta de datos [19].

Hay muchas áreas donde algoritmos de *minería de datos* fueron implementados exitosamente. Algunos ejemplos son el ámbito financiero donde se analizan datos para predecir tendencias de mercados o las historias clínicas de pacientes donde se busca encontrar relaciones ocultas entre tratamientos y síntomas [20].

Las técnicas asociadas a *minería de datos* se pueden separar en dos grupos: métodos predictivos y métodos descriptivos [21]. Los métodos predictivos – como agrupación y clasificación de datos – realizan estimaciones acerca de las propiedades de los datos. Un ejemplo para ese grupo de algoritmos son aplicaciones financieras donde es importante obtener una estimación de lo que va a pasar en el futuro.

En cambio, los métodos descriptivos – como reconocimiento de padrones – buscan caracterizar las propiedades de los datos. Por ejemplo, cuando se estudian relaciones entre

distintos tratamientos médicos y los síntomas de los pacientes, el objetivo es encontrar características (o, en otras palabras, padrones) que aparezcan frecuentemente en la base de datos, pero la idea no es predecir qué consecuencias van a tener nuevos medicamentos.

El caso particular de agricultura de precisión abarca ambos campos de *minería de datos*. Por un lado, es necesario predecir ciertas variables, como la fecha óptima para la cosecha [22] o las características del suelo [23]. Por el otro, es importante conocer el estado del cultivo a través de métodos descriptivos. Algunos ejemplos son la detección del estado hídrico [24] o enfermedades del cultivo [25].

1.3. Motivación, desafíos y objetivos

La agricultura de precisión y la *minería de datos* coexisten desde aproximadamente 25 años. Sin embargo, ambas disciplinas no se han acercado demasiado. Por un lado, hay resentimientos de los ingenieros agrónomos respecto a modelos muy teóricos, por el otro, los informáticos muchas veces prefieren trabajar en ámbitos estériles en vez de “ensuciarse las manos”. Ni hablar de los productores agropecuarios que, en su mayoría, son escépticos con respecto al uso de tecnologías avanzadas, sobre todo porque es difícil cuantificar precisamente los beneficios. Por estas razones, las tasas de implementación de agricultura de precisión son muy bajas, incluso en países avanzadas, como, por ejemplo Inglaterra donde la tasa de agricultura de precisión es del 10 % [26].

Un problema teórico al aplicar algoritmos de *minería de datos* en la agricultura de precisión es que muchas veces no se tienen en cuenta correlaciones espaciales para simplificar los modelos matemáticos [27]. Otro desafío es juntar datos de teledetección (datos macroscópicos) con datos microscópicos, como la humedad de suelo. Si bien hay varios algoritmos de fusión de datos, están casi siempre limitados – a datos de teledetección [28] o a datos locales [29] – pero generalmente no pueden procesar distintos tipos de datos. Por último, cabe destacar que prácticamente no existen modelos estadísticos para la agricultura de precisión que permitan el uso de funciones de probabilidad que no sea la distribución normal, aunque se sabe que dicha distribución no es adecuada para muchos conjuntos de datos [30].

Considerando todos los problemas y desafíos mencionados, el objetivo general de esta tesis es acercar los modelos teóricos basados en campos aleatorios de Markov a la agricultura

de precisión. Para cumplir con este objetivo principal, es necesario resolver los siguientes tres aspectos cruciales:

Primero, es inevitable adaptar las suposiciones de modelos universales a los requerimientos particulares de la agricultura de precisión. Esto implica, entre otras cosas:

- mejorar la flexibilidad de los algoritmos con respecto a las funciones de densidad, ya que muchos datos relacionados con la agricultura de precisión no siguen la distribución normal.
- mejorar la flexibilidad de los algoritmos con respecto a las vecindades usadas, ya que las relaciones espaciales de una variable (como la fotosíntesis) pueden ser muy limitadas, mientras otras variables (como la humedad de suelo) muestran generalmente una fuerte correlación espacial (si el suelo está húmedo en un sitio es probable que también esté húmedo en el siguiente sitio).

Segundo, es importante demostrar que los modelos de este trabajo realmente funcionan mejor para aplicaciones de agricultura de precisión que los algoritmos de referencia. Por consiguiente, hay que diseñar experimentos que representen lo mejor posible las circunstancias de la agricultura real. Cabe destacar que este objetivo se puede lograr con datos reales y con datos simulados, siempre cuando los datos simulados se orientan en las aplicaciones reales.

Tercero, es deseable incorporar varios sensores y medidores de la agricultura de precisión al proceso de reconocimiento de patrones. De esta forma, se aprovechan las tecnologías existentes para aliviar los efectos de mediciones ruidosas y atípicas, aumentando así la confianza del productor agrícola. En este contexto, un objetivo particular de esta tesis es el desarrollo de un nuevo aparato de bajo costo que mida el potencial eléctrico de un tallo.

1.4. Metodología

Esta tesis se basa en la teoría de campos aleatorios de Markov [31]. Para ciertas suposiciones se pueden derivar algoritmos como *Hidden Markov Models* (HMM) [32] o *Iterated Conditional Modes* (ICM) [33] de la teoría universal.

La metodología en este trabajo consiste en modificar y reducir las suposiciones, necesarias para existentes algoritmos, con el objetivo de derivar nuevos modelos y métodos estadísticos. Para verificar que las suposiciones de esta tesis se adapten mejor a las necesidades particulares de la agricultura de precisión que los algoritmos de referencia, se realizan experimentos con datos reales y simulados junto con su verdad de campo. Como índice de concordancia entre los resultados obtenidos y la verdad de campo se utiliza el coeficiente κ de Cohen [34].

1.5. Contribuciones de la tesis

Esta tesis contiene tres contribuciones principales.

- Un nuevo modelo de Markov en dos dimensiones que es una alternativa al algoritmo de Viterbi (CEP)
- Un nuevo modelo basado en el modelo de Potts que permite el uso de funciones de probabilidad arbitrarias (SBM)
- Un nuevo método para la fusión de datos no equidistantes (MDF)

En total se aportaron diez trabajos en congresos locales [35–44], cuatro artículos en congresos internacionales [45–48] y tres publicaciones en revistas internacionales [49–51]. Además, hay dos trabajos que están siendo revisadas para su publicación en revistas internacionales [52, 53].

Durante la elaboración de esta tesis se dieron dos charlas importantes:

- Seminario en el Instituto Gulich - Comisión Nacional de Actividades Espaciales (CONAE), “Segmentación de imágenes con modelos markovianos de dos dimensiones” el 6 de agosto de 2013.
- Charla plenario “Procesos Gaussianos” en el Workshop de Inteligencia Artificial en la Universidad Nacional de Córdoba 2013 organizado por la IEEE Computational Intelligence Society el 6 de noviembre de 2013.

Otros logros destacables son:

- Presentación de algoritmos y resultados en un stand en Mercolactea del 13 al 15 de mayo de 2010.
- Premio “Best Professional Paper Award - First Place” en el congreso *1st BRICS Countries Congress on Computational Intelligence BRICS-CCI*.
- Revisor de la revista IEEE Geoscience and Remote Sensing Letters.
- Revisor de la revista IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.
- Revisor de varios congresos nacionales (Congreso Argentino de AgroInformática - CAI, Reunión de Trabajo en Procesamiento de la Información y Control - RPIC, Congreso Bienal de IEEE Argentina - ARGENCON).
- Implementación y distribución de la *Markov Models Toolbox* en Matlab[®] de acuerdo al *Reproducible Research Paradigm*.

1.6. Organización de la tesis

Esta tesis consiste de tres partes. La primera parte, explica el estado de arte y avances teóricos realizados. Comienza con el capítulo 2, donde se presenta la teoría general de campos aleatorios de Markov. A continuación, se introducen modelos causales en dos dimensiones en el capítulo 3. Además, se describe un modelo de referencia y se presenta un nuevo algoritmo. Luego, se detalla la teoría del modelo de Potts en el capítulo 4. Igual que para modelos causales, se introduce un algoritmo de referencia y un método nuevo. Finalmente, se desarrolla un algoritmo novedoso para la fusión de datos multisensores en el capítulo 5.

La segunda parte de esta tesis evalúa los algoritmos teóricos para datos reales y simulados. Primero, se introducen los sensores más usados en la agricultura de precisión y se destacan algunas de sus características en el capítulo 6. En el capítulo 7 se muestran los resultados experimentales.

En la tercera parte, que corresponde al capítulo 8, se presenta una discusión de los resultados. Además, se sacan conclusiones acerca de las suposiciones y el desempeño de

los modelos teóricos. Por último, se marcan puntos abiertos que pueden ser resueltos en futuros trabajos.

Capítulo 2

Campos aleatorios de Markov

Campos aleatorios de Markov es una teoría de probabilidad para modelar y analizar relaciones espaciales de variables aleatorias. Esta teoría fue introducida por Dobrushin como una extensión de modelos ocultos de Markov en una dimensión [54]. La teoría general se aplica a relaciones en n dimensiones, pero en este trabajo se utilizan solo campos aleatorios en dos dimensiones.

El esquema principal de campos aleatorios de Markov se muestra en la figura 2.1. Detrás de cada observación hay un estado oculto. Asimismo, los estados ocultos están conectados con sus vecinos.

Después de describir los fundamentos teóricos en la sección 2.1 se presentan distintas implementaciones basados en un modelo causal y en el modelo de Potts en la sección 2.3. Finalmente se marca el rol que juegan los campos aleatorios de Markov dentro de la

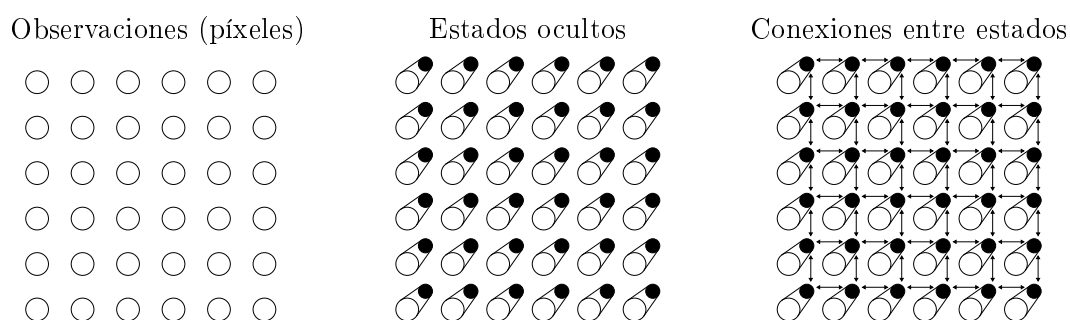


FIGURA 2.1: Esquema de campos aleatorios de Markov. Cada sitio de observación tiene un estado oculto. A la vez los estados ocultos están conectados entre ellos.

minería de datos en la sección 2.4. En la misma sección se muestran algunas aplicaciones de campos aleatorios de Markov en tecnologías emergentes.

2.1. Fundamentos teóricos

Dado un conjunto de nodos $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$ (por ejemplo los píxeles de una imagen). Al conjunto \mathcal{S} se le asignan etiquetas discretas $\boldsymbol{\omega} = \{\omega_s, s \in \mathcal{S}\}$ donde cada ω_s es una variable aleatoria que toma un valor de $\Lambda = \{1, 2, \dots, L\}$. Entonces $\boldsymbol{\omega}$ es un campo aleatorio llamado proceso de etiquetado.

Además consideramos que las mediciones son una realización $\mathcal{X} = \{x_s, s \in \mathcal{S}\}$ de un campo aleatorio que a su vez es una función del proceso de etiquetado. Al proceso \mathcal{X} se denomina proceso de observación.

Entonces hay un proceso oculto de etiquetado $\boldsymbol{\omega}$ y luego un proceso de observación \mathcal{X} , que incluye ruido. Si modelamos las L clases con distintas funciones de probabilidad, se puede pensar en las mediciones como una muestra de $\boldsymbol{\omega}$.

Para que $(\boldsymbol{\omega}, \mathcal{X})$ se convierte en un campo aleatorio de Markov es necesario definir vecindades para cada nodo de \mathcal{S} . Por consiguiente introducimos un grafo no dirigido $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ donde las aristas \mathcal{E} indican que nodos de \mathcal{S} son vecinos y por lo tanto se encuentran en una relación de dependencia.

Se denomina $\mathcal{N}(s_i)$ la vecindad del nodo s_i y se define que $s_j \in \mathcal{N}(s_i)$ si $\{s_i, s_j\} \in \mathcal{E}$. Cabe destacar que para el grafo no dirigido vale $s_j \in \mathcal{N}(s_i) \Leftrightarrow s_i \in \mathcal{N}(s_j)$.

Definición 2.1. Un grafo no dirigido \mathcal{G} se llama campo aleatorio de Markov si para cualquier nodo s_i del grafo se cumple

$$P(\omega_i | \omega_1, \dots, \omega_{i-1}, \omega_{i+1}, \dots, \omega_N) = P(\omega_i | \omega_{\mathcal{N}(s_i)}), \quad (2.1)$$

donde $\omega_{\mathcal{N}(s_i)}$ representa todas las etiquetas de los nodos de la vecindad de s_i .

La definición 2.1 se puede leer como una generalización de la definición de cadenas de Markov a varias dimensiones. Lo que básicamente dice es: la variable ω_i depende de un cierto entorno (o vecindad) pero no de observaciones fuera de su vecindad.

Entonces, el primer paso para aplicar la teoría universal consiste en definir una vecindad. Una vez establecido esta vecindad, el objetivo principal es encontrar las etiquetas ocultas ω . En las siguientes secciones se presentan los dos enfoques principales para hallar ω pero antes se presenta la nomenclatura usada en este trabajo.

2.2. Nomenclatura

Este trabajo investiga exclusivamente datos en dos dimensiones, aunque los conceptos teóricos se pueden extender a más dimensiones sin inconvenientes. Los sitios físicos de los datos son representados por el conjunto \mathcal{S} . Cuando se trata de observaciones en una rejilla rectangular, como los píxeles de una imagen, se refiere a la posición (i, j) o al píxel (i, j) con $s_{i,j}$. En este caso las dimensiones de la rejilla son $u \times v$ y $s_{i,j}$ representa el punto de la fila i en la columna j . Si los datos tienen coordenadas arbitrarias se numeran los sitios de medición y se denomina s_i a la i -ésima posición.

El conjunto de las etiquetas ocultas se llama ω . En caso de tener observaciones en sitios arbitrarios, ω_i hace referencia a la etiqueta de s_i . Para el caso contrario, $\omega_{i,j}$ es la etiqueta de $s_{i,j}$. En ambos casos las etiquetas toman un valor del espacio finito de etiquetas $\Lambda = \{\ell_1, \ell_2, \dots, \ell_L\}$. Además, se llama \mathbf{X} a las observaciones o mediciones. Siguiendo la lógica introducida $x_{i,j}$ representa la medición en $s_{i,j}$ y x_i es el valor observado en s_i . Finalmente se denomina $\mathcal{N}_{i,j}$ a la vecindad del punto $s_{i,j}$ (\mathcal{N}_i en el caso de observaciones no equidistantes) y $\omega_{\mathcal{N}_{i,j}}$ son las etiquetas de los vecinos de $s_{i,j}$.

Cuando se trabaja con modelos estadísticos θ representa los parámetros del modelo. Generalmente se trata de los parámetros de las funciones de probabilidad de las distintas clases y algunas propiedades específicas como probabilidades de transición o parámetros de suavidad.

2.3. Modelos teóricos

Los modelos y algoritmos basados en campos aleatorios de Markov se pueden agrupar en dos clases. Primero, los modelos causales basados en Markov Mesh y segundo los modelos de Potts. En la siguiente sección se introducen los modelos causales y luego se presentan los modelos de Potts.

Para simplificar la notación en esta sección, se supone que \mathcal{S} es una rejilla regular como por ejemplo los píxeles de una imagen. Sin embargo, todas las formulas pueden ser generalizadas a sitios arbitrarios.

2.3.1. Modelo causal

El primer modelo basado en Markov Mesh fue introducido por Abend, Harley and Kanal en 1965 [55] y formalizado por Besag en 1974 [56]. La particularidad de estos modelos es que la dependencia markoviana es causal. A grandes rasgos esto quiere decir que se trabaja con una vecindad que se puede interpretar como el “pasado” de un píxel. Formalizando esta notación se obtiene la siguiente definición.

Definición 2.2. Un modelo de un campo aleatorio de Markov es causal si para todos los sitios $s_{i,j}$ con vecindad $\mathcal{N}_{i,j}$ vale:

$$P(\omega_{i,j} | \omega_{Z_{i,j}}, \mathbf{X}_{Z_{i,j}}, \theta) = P(\omega_{i,j} | \omega_{\mathcal{N}_{i,j}}, \mathbf{X}_{\mathcal{N}_{i,j}}, \theta), \quad (2.2)$$

donde $Z_{i,j} = \{(i', j') : (i', j') \prec (i, j)\}$ para un operador \prec que representa una relación causal.

Algunas posibles implementaciones de \prec pueden ser:

$$(i', j') \prec (i, j) \quad \text{si } i' < i \text{ o } j' < j \quad (2.3)$$

$$(i', j') \prec (i, j) \quad \text{si } i' < i \text{ e } j' < j \quad (2.4)$$

$$(i', j') \prec (i, j) \quad \text{si } i' < i \text{ e } j' - j = i' - i \quad (2.5)$$

En los ejemplos (2.3), (2.4) y (2.5) se estableció una relación markoviana donde $s_{1,1}$ es el inicio y $s_{m,n}$ es el último sitio en la cadena causal. Sin profundizar problemas de inicio y de borde en este capítulo, cabe destacar que se podrían haber establecido las mismas relaciones de cualquier esquina hacia la esquina diagonal. En la figura 2.2 se muestran las vecindades según (2.3), (2.4) y (2.5) y en el capítulo 3 se derivan dos algoritmos de la teoría general.

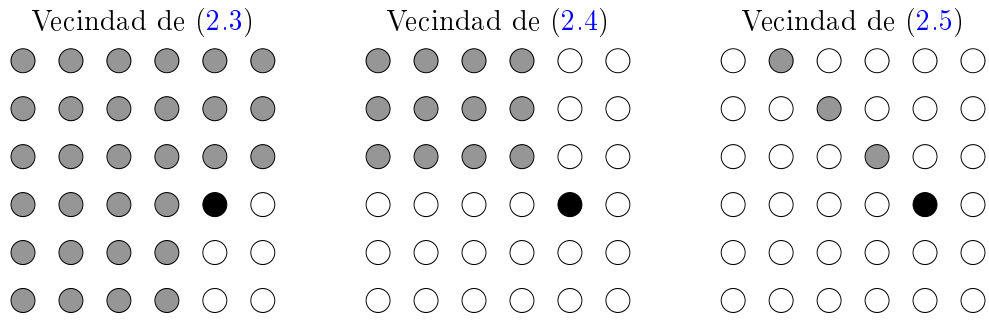


FIGURA 2.2: Ejemplos de vecindades para un modelo causal de Markov. Los sitios grises son los vecinos del sitio negro.

2.3.2. Modelo de Potts

Los modelos de Potts se basan en campos aleatorios de Gibbs. Para que un conjunto de variables aleatorias se considere un campo aleatorio de Gibbs es necesario que siga una distribución de Gibbs.

Primero consideramos el grafo no dirigido $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ de la sección 2.1. Ahora llamamos *clique* a cada subconjunto de nodos que estén completamente conectados entre ellos y denominamos \mathcal{C} el conjunto de todos los cliques c_i en \mathcal{G} .

Definición 2.3. Una función de probabilidad $P(X)$ definida sobre un grafo no dirigido \mathcal{G} se llama distribución de Gibbs si vale:

$$P(X) = \frac{1}{Z} \prod_{c_i \in \mathcal{C}} \phi_i(c_i) \quad \text{con} \quad \phi_i(c_i) > 0 \quad \forall c_i \in \mathcal{C}, \quad (2.6)$$

donde Z es una constante de normalización y $\phi_i(c_i)$ es el potencial del clique c_i .

Se observa que los campos aleatorios de Gibbs se refieren según su distribución a una propiedad global mientras los campos aleatorios de Markov hacen referencia a una propiedad local basada en vecindades. Para conectar ambas teorías usamos el teorema de *Hammersley-Clifford* [57]. Este teorema dice que las definiciones 2.1 y 2.3 son equivalentes. O sea, un campo aleatorio de Markov es al mismo tiempo un campo aleatorio de Gibbs y viceversa.

Entonces es válido derivar métodos y algoritmos para campos aleatorios de Markov usando la teoría de campos aleatorios de Gibbs. En este trabajo estudiamos modelos de Potts

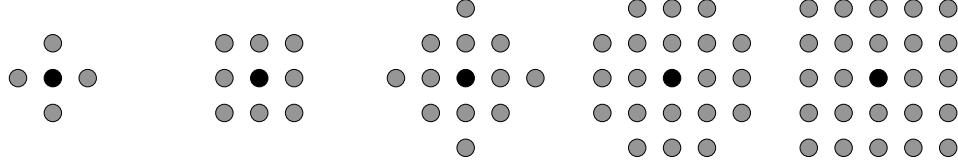


FIGURA 2.3: Ejemplos de vecindades para el modelo de Potts.

que son una clase particular de los modelos gibbsianos. En su forma más general, los potenciales del modelo de Potts son:

$$\Phi_{\mathcal{S}}(\boldsymbol{\omega}) = \begin{cases} -\alpha_{\ell} & \text{si } \omega_s = \ell; \\ -\beta_{\ell, \tilde{\ell}} & \text{si } \omega_s = \ell, \omega_t = \tilde{\ell} \text{ con } s \in \mathcal{N}_t; \\ 0 & \text{en otro caso.} \end{cases} \quad (2.7)$$

La ecuación (2.7) define el modelo de Potts anisotrópico con campo externo. Si $\alpha_{\ell} = 0 \forall \ell \in \Lambda$ el modelo no tiene campo externo y si $\beta_{\ell, \tilde{\ell}} \equiv \beta \forall \ell, \tilde{\ell} \in \Lambda$ se trata de un modelo isotrópico.

Al igual que en la mayoría de las publicaciones relacionadas a modelos de Potts [58, 59], se usan solamente modelos isotrópicos sin campo externo en este trabajo. El potencial de este modelo viene dado por

$$\Phi_{\mathcal{S}}^{\beta}(\boldsymbol{\omega}) = \begin{cases} -\beta & \text{si } \omega_s = \omega_t \text{ con } s \in \mathcal{N}_t; \\ 0 & \text{en otro caso.} \end{cases} \quad (2.8)$$

donde $\beta \in \mathbb{R}$ es un parámetro llamado temperatura inversa [60]. El parámetro β indica el grado de vínculo que hay entre dos observaciones vecinas. Note que mientras más grande es β , más fuerte el vínculo entre dos vecinos. Al contrario, no existe ningún vínculo entre vecinos para el caso extremo $\beta = 0$.

2.4. Aplicaciones, limitaciones y desafíos

Como se mencionó en la sección 2.1, la teoría de campos aleatorios de Markov supone que cada sitio de un conjunto de observaciones está relacionado con sitios vecinos, pero no con sitios fuera de su vecindad. Por un lado, esta característica limita o impide muchas veces la aplicación de la teoría, por el otro es una base teórica muy adecuada para tareas específicas.

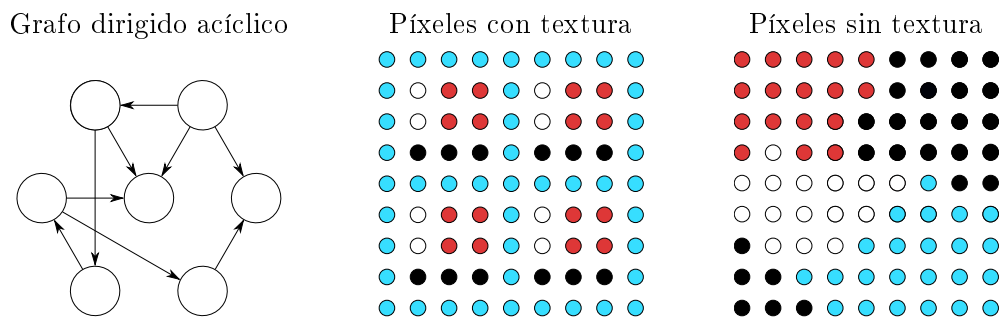


FIGURA 2.4: Es difícil modelar un grafo dirigido acíclico y una imagen con textura con campos aleatorios de Markov mientras la teoría se adapta muy bien a imágenes sin textura con zonas homogéneas.

Uno de los problemas de los campos aleatorios de Markov consiste en que la teoría se basa en un grafo no dirigido. En otras palabras, la relación entre dos sitios es mutua. Por lo tanto, campos aleatorios de Markov no son la herramienta adecuada para modelar dependencias unidireccionales como aparecen en grafos dirigidos acíclicos. Para este tipo de relaciones es preferible utilizar otros modelos como redes bayesianas.

Otro desafío para campos aleatorios de Markov es el procesamiento de datos (sobre todo de imágenes) con textura. Si bien es posible adaptar modelos markovianos para detectar y analizar texturas, hay que destacar que siempre son ajustes artificiales que no surgen naturalmente de la teoría general [61]. Una posible solución es la aplicación de varios campos aleatorios con distintas resoluciones espaciales. Otra solución consiste en la transformación de las observaciones al espacio de la frecuencia. Sin embargo, hay que reconocer que existen modelos más adecuados para este tipo de datos.

En contrario a los escenarios mencionados, la teoría de campos aleatorios de Markov es una de las teorías más aplicadas en la segmentación de imágenes sin textura. Esto se debe a que las suposiciones de la teoría (píxeles vecinos son probablemente de la misma clase) coinciden muy bien con la realidad. Por ejemplo en el área de agricultura de precisión se utiliza extensamente la teoría de campos aleatorios de Markov para detectar zonas homogéneas dentro de un lote [59,62]. En la figura 2.4 se muestran ejemplos de las tareas descritas en esta sección.

En el próximo capítulo se estudian los modelos causales y sus posibles implementaciones a la segmentación de imágenes. Después, en el capítulo 4, se presentan dos modelos de Potts isotrópicos sin campo externo.

Capítulo 3

Modelos ocultos de Markov en dos dimensiones

Los modelos ocultos de Markov en una dimensión tienen una larga y exitosa historia empezando con los trabajos de Baum en 1966 [32]. Estos modelos sirven para modelar y analizar datos secuenciales de una dimensión como reconocimiento de voz [63], secuencias de proteína [64] o cadenas de suministro [65]. En este capítulo se presenta una extensión de los modelos ocultos de Markov a dos dimensiones. Después de marcar el esquema general, se introducen distintas suposiciones para simplificar las cuentas y así llegar a modelos tratables. Cabe destacar que la teoría presentada en este capítulo se puede extender fácilmente a más de dos dimensiones.

3.1. Esquema general

Un modelo oculto de Markov en dos dimensiones es un modelo probabilístico con relación causal, que sirve para describir y analizar observaciones ruidosas con relaciones espaciales. Entonces, el primer paso para definir el modelo es fijar una zona de influencia, o, en otras palabras, una vecindad. En el capítulo 2 se demostró que un modelo causal necesita una vecindad causal definida por un operador \prec .

En esta tesis se trabaja con una vecindad causal estándar que ha sido utilizada por muchos autores [66–68]. La definición formal de la relación \prec viene dada por:

$$(i', j') \prec (i, j) \quad \text{si } i' < i \quad \text{o} \quad i' = i \quad \text{e} \quad j' < j. \quad (3.1)$$

Además definimos la relación “rotada” $\tilde{\prec}$:

$$(i', j') \tilde{\prec} (i, j) \quad \text{si } j' < j \quad \text{o} \quad j' = j \quad \text{e} \quad i' < i. \quad (3.2)$$

Con estas definiciones fijamos los conjuntos:

$$\begin{aligned} \Psi &= \{(i', j') : (i', j') \prec (i, j)\} \\ \tilde{\Psi} &= \{(i', j') : (i', j') \tilde{\prec} (i, j)\} \\ \tilde{\Psi} \cup \Psi &= \{(i', j') : (i', j') \prec (i, j) \text{ o } (i', j') \tilde{\prec} (i, j)\} \\ \tilde{\Psi} \cap \Psi &= \{(i', j') : (i', j') \prec (i, j) \text{ e } (i', j') \tilde{\prec} (i, j)\} \\ \tilde{\Psi} - \Psi &= \{(i', j') : (i', j') \tilde{\prec} (i, j) \text{ e } (i', j') \succ (i, j)\} \end{aligned} \quad (3.3)$$

En la figura 3.1 se muestran los conjuntos Ψ , $\tilde{\Psi}$, $\tilde{\Psi} \cup \Psi$, $\tilde{\Psi} \cap \Psi$, $\tilde{\Psi} - \Psi$ y una vecindad de dos píxeles. La idea ahora es, demostrar que bajo la suposición de Markov el conjunto más grande $\tilde{\Psi} \cup \Psi$ (que incluye los otros conjuntos) puede ser reducido a los dos píxeles de la vecindad \prec según la figura 3.1. O sea, queremos demostrar que:

$$P(\omega_{i,j} | \omega_{i',j'}, x_{i',j'} : (i', j') \in \tilde{\Psi} \cup \Psi) = P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}). \quad (3.4)$$

Primero aplicamos la regla de Bayes para reescribir la probabilidad condicional como una fracción de probabilidades conjuntas. En el mismo paso dividimos $\tilde{\Psi} \cup \Psi$ en las partes Ψ y $\tilde{\Psi} - \Psi$.

$$\begin{aligned} P(\omega_{i,j} | \omega_{i',j'}, x_{i',j'} : (i', j') \in \tilde{\Psi} \cup \Psi) &= \\ \frac{1}{\gamma_0} P(\omega_{i,j}, \omega_{i',j'}, x_{i',j'}, \omega_{i'',j''}, x_{i'',j''} : (i', j') \in \Psi, (i'', j'') \in \tilde{\Psi} - \Psi), & \end{aligned} \quad (3.5)$$

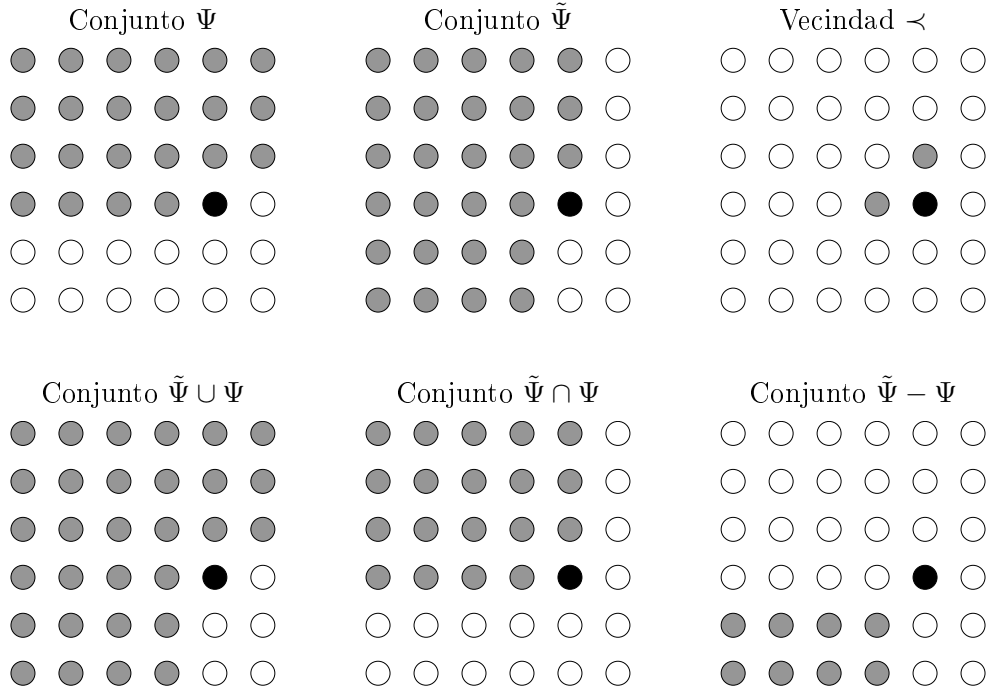


FIGURA 3.1: Usando los conjuntos auxiliares Ψ , $\tilde{\Psi}$, $\tilde{\Psi} \cap \Psi$ y $\tilde{\Psi} - \Psi$ se puede mostrar que los dos píxeles de la vecindad \prec son suficientes para representar el conjunto $\tilde{\Psi} \cup \Psi$ bajo la suposición de Markov.

donde $\gamma_0 = P(\omega_{i',j'}, x_{i',j'} : (i', j') \in \tilde{\Psi} \cup \Psi)$. Ahora, podemos expandir las probabilidades condicionales. Para $\gamma_1 = P(\omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi)$ obtenemos:

$$\begin{aligned} & \frac{1}{\gamma_0} P\left(\omega_{i,j}, \omega_{i',j'}, x_{i',j'}, \omega_{i'',j''}, x_{i'',j''} : (i', j') \in \Psi, (i'', j'') \in \tilde{\Psi} - \Psi\right) = \\ & \frac{\gamma_1}{\gamma_0} P\left(\omega_{i,j} | \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) \\ & P\left(\omega_{i'',j''}, x_{i'',j''} : (i'', j'') \in \tilde{\Psi} - \Psi \mid \omega_{i,j}, \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) \end{aligned} \quad (3.6)$$

En este punto se puede aplicar la suposición de Markov que permite remplazar la dependencia de la vecindad Ψ por solo dos píxeles del “pasado” $\omega_{i-1,j}$ y $\omega_{i,j-1}$. Entonces se cumple

$$\begin{aligned} & \frac{\gamma_1}{\gamma_0} P\left(\omega_{i,j} | \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) \times \\ & P\left(\omega_{i'',j''}, x_{i'',j''} : (i'', j'') \in \tilde{\Psi} - \Psi \mid \omega_{i,j}, \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) = \\ & \frac{\gamma_1}{\gamma_0} P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}) \times \\ & P\left(\omega_{i'',j''}, x_{i'',j''} : (i'', j'') \in \tilde{\Psi} - \Psi \mid \omega_{i,j}, \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) \end{aligned} \quad (3.7)$$

Si ahora asumimos que las observaciones $x_{i,j}$ dependen solo de $\omega_{i,j}$ podemos simplificar la correspondiente probabilidad condicional.

$$\begin{aligned} & \frac{\gamma_1}{\gamma_0} P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}) \times \\ & P\left(\omega_{i'',j''}, x_{i'',j''} : (i'', j'') \in \tilde{\Psi} - \Psi \mid \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) = \\ & P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}) \times \\ & \frac{\gamma_1}{\gamma_0} P\left(\omega_{i'',j''}, x_{i'',j''} : (i'', j'') \in \tilde{\Psi} - \Psi \mid \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) \end{aligned} \quad (3.8)$$

En la figura 3.1 se puede observar que $\tilde{\Psi} \cup \Psi = \Psi \cup (\tilde{\Psi} - \Psi)$. Recuerde que γ_1 actuó sobre el conjunto Ψ entonces se cancela el último término y se obtiene finalmente:

$$\begin{aligned} & P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}) \times \\ & \frac{\gamma_1}{\gamma_0} P\left(\omega_{i'',j''}, x_{i'',j''} : (i'', j'') \in \tilde{\Psi} - \Psi \mid \omega_{i',j'}, x_{i',j'} : (i', j') \in \Psi\right) = \\ & P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}) \end{aligned} \quad (3.9)$$

Entonces, se ha demostrado que

$$P\left(\omega_{i,j} | \omega_{i',j'}, x_{i',j'} : (i', j') \in \tilde{\Psi} \cup \Psi\right) = P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1}). \quad (3.10)$$

La ecuación (3.10) expresa que, para aplicar la vecindad $\tilde{\Psi} \cup \Psi$, es suficiente usar los píxeles $(i-1, j)$ y $(i, j-1)$ bajo la suposición de Markov y asumiendo que cada observación depende solo de su estado oculto. Si además, asumimos que la probabilidad de transición $P(\omega_{i,j} | \omega_{i-1,j}, \omega_{i,j-1})$ no depende de los píxeles (i, j) , $(i-1, j)$ y $(i, j-1)$ se puede aplicar la siguiente simplificación:

$$a_{m,n,l} = P(\omega_{i,j} = l | \omega_{i-1,j} = m, \omega_{i,j-1} = n). \quad (3.11)$$

Note que la suposición de la ecuación (3.11) es muy razonable y no muy restrictiva. Solo expresa que el escenario de pasar de los estados m y n al estado l tiene la misma probabilidad en toda la imagen independientemente de su ubicación. En la figura 3.2 se muestra el efecto de la ecuación (3.11). En consecuencia, hay L^3 distintas probabilidades de transición si la imagen tiene L estados. Sin la simplificación de la ecuación (3.11) habrían $L^3 uv$ distintas probabilidades de transición para una imagen con dimensiones $u \times v$. Si bien $L^3 uv$ es generalmente una cantidad de transiciones manejable, hay que

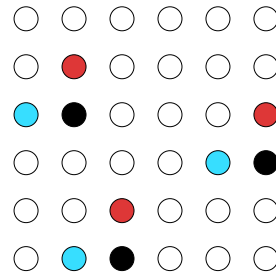


FIGURA 3.2: Ejemplo de la implementación de la ecuación (3.11): la probabilidad de transición de los estados ● y ● al estado ● es la misma en toda la imagen independiente de la ubicación de los píxeles.

tener en cuenta que sería muy difícil estimar todas estas probabilidades con una sola observación.

Por último, se discute un problema que ocurre para los píxeles de la primera columna ($j = 1$) y de la primera fila ($i = 1$). A los píxeles de la primera columna les falta el píxel $(i, j - 1)$ del “pasado” mientras que los píxeles de la primera fila no tienen el píxel $(i - 1, j)$ del “pasado”. Existen dos posibles soluciones para este problema: Primero, agregar copias de la primera columna y fila, como columna y fila “cero”. En este caso se pueden usar las probabilidades de máxima verosimilitud como probabilidades de estado. Segundo, suponer probabilidades de estado uniformes para los píxeles en la columna y la fila “cero”. O sea, cada estado tiene la misma probabilidad en los píxeles auxiliares.

En este trabajo se opta por la segunda opción ya que la primera opción de copiar píxeles tiene una desventaja destacable. Imaginemos que un píxel de la primera columna tiene un valor de gris muy distinto al de sus vecinos, como ocurre frecuentemente en observaciones ruidosas. Entonces, al copiar este valor de gris se fortalece el efecto del ruido porque se agrega un píxel del “pasado” que tiene las mismas características. En cambio, si usamos probabilidades de estado uniformes es más probable que el estado del píxel ruidoso se adapte a su vecindad.

En resumen, hay que subrayar dos ventajas de los modelos causales con respecto a los modelos de Potts (que se basan en campos aleatorios de Gibbs). Primero, los modelos causales no mezclan píxeles del “pasado” y del “futuro”, justamente porque trabajan con una relación causal. Segundo, los modelos causales en general necesitan menos parámetros que los modelos de Potts donde se usan parámetros de vecindad, transiciones de fase, etc. [69].

3.2. Teoría exacta

En la sección anterior se plantearon los fundamentos de modelos ocultos de Markov en dos dimensiones. En esta sección se detalla cómo ajustar y evaluar un modelo oculto de Markov en dos dimensiones de forma exacta. Luego, se muestra porque es generalmente imposible resolver las ecuaciones matemáticas de esta sección sin aplicar algunas suposiciones sobre los parámetros involucrados. Para no complicar las derivaciones de esta sección se utilizan solo datos de una banda (niveles de gris) y no datos multiespectrales. Sin embargo, los conceptos teóricos se pueden generalizar a datos multiespectrales sin inconvenientes.

El objetivo principal de los modelos markovianos consiste en hallar el mapa de estados ocultos óptimo ω^* . Si juntamos los parámetros del modelo en θ y llamamos \mathbf{X} a las observaciones obtenemos:

$$\omega^* = \arg \max_{\omega} P(\omega | \mathbf{X}, \theta). \quad (3.12)$$

Note en la ecuación (3.12), que θ representa las probabilidades de transición $a_{m,n,l}$, las probabilidades de emisión y las probabilidades de inicio. En esta sección se muestran los fundamentos teóricos de cómo obtener θ de forma exacta.

En los modelos ocultos de Markov en una dimensión el método estándar para hallar ω^* es el algoritmo de *Baum-Welch* [70]. Lamentablemente este algoritmo no es tratable para observaciones en dos dimensiones. Por lo tanto, es necesario derivar nuevos métodos para datos que tengan una relación espacial en dos dimensiones. Como base teórica se usa el algoritmo de *Expectation-Maximization* (EM) [71]. Partiendo del algoritmo EM seguimos la derivación de Jia Li et. al. [72].

En lo siguiente, $I(\cdot)$ es la función indicadora que vale 1 si el argumento es cierto y 0 en el caso contrario. Con ω denominamos todas las posibles configuraciones de estados ocultos de \mathcal{S} .

El primer paso para describir un modelo oculto de Markov en dos dimensiones consiste en calcular la probabilidad de transición de los estados m y n al estado l en el píxel (i, j) . Denominamos esta probabilidad $H_{m,n,l}(i, j)$. Si por ahora omitimos el termino de

normalización (y usamos \propto en vez de $=$) obtenemos:

$$H_{m,n,l}(i, j) \propto \sum_{\omega} I(m=\omega_{i-1,j}, n=\omega_{i,j-1}, l=\omega_{i,j}) \prod_{(i',j') \in \mathcal{S}} P(x_{i',j'} | \theta) a_{\omega_{i'-1,j'}, \omega_{i',j'-1}, \omega_{i',j'}} \quad (3.13)$$

En la ecuación (3.13), θ son los parámetros del modelo y $x_{i,j}$ es la observación o el valor de gris del píxel (i, j) . Note que $H_{m,n,l}$ es una suma sobre todas las posibles configuraciones de ω más un producto que involucra todos los píxeles de la imagen. En la próxima sección utilizamos estas características de $H_{m,n,l}$ para explicar el motivo por el cual la ecuación (3.13) generalmente no es tratable pero por ahora seguimos con la derivación exacta.

Con los valores $H_{m,n,l}$ es fácil obtener las probabilidades de transición $a_{m,n,l}$. Lo único que hay que hacer es promediar los valores de $H_{m,n,l}$ sobre todos los píxeles (i, j) :

$$a_{m,n,l} = \frac{\sum_{(i,j) \in \mathcal{S}} H_{m,n,l}(i, j)}{\sum_{l'=1}^L \sum_{(i,j) \in \mathcal{S}} H_{m,n,l'}(i, j)}. \quad (3.14)$$

El problema que enfrentamos aquí es que la ecuación (3.13) determina $H_{m,n,l}$ usando $a_{m,n,l}$ mientras en la ecuación (3.14) se computa $a_{m,n,l}$ usando $H_{m,n,l}$. Para resolver este dilema se utiliza el algoritmo EM que contiene un esquema iterativo para estos tipos de incógnitas. La ventaja del EM es que se puede demostrar que el algoritmo siempre converge a un óptimo local. Obviamente el óptimo local no es necesariamente el óptimo global ω^* pero por lo menos tenemos la certeza de que el algoritmo va a converger. Puesto que las cuentas de esta sección no son tratables – como demostraremos en la sección 3.3 – se detalla el algoritmo EM en la sección 3.5 donde se presentan dos realizaciones de modelos ocultos de Markov en dos dimensiones.

Además de las probabilidades de transición es necesario estimar las probabilidades de emisión. Primero, tenemos que definir una variable auxiliar que nos sirva para ajustar los parámetros de las funciones de probabilidad. Llamamos $Q_l(i, j)$ a la probabilidad de que el píxel (i, j) tenga el estado oculto l . $Q_l(i, j)$ viene dado por:

$$Q_l(i, j) \propto \sum_{\omega} I(l = \omega_{i,j}) \prod_{(i',j') \in \mathcal{S}} P(x_{i',j'} | \theta_{\omega_{i',j'}}) a_{\omega_{i'-1,j'}, \omega_{i',j'-1}, \omega_{i',j'}}, \quad (3.15)$$

donde $\theta_{\omega_{i',j'}}$ son los parámetros de la función de probabilidad que corresponde al estado $\omega_{i',j'}$. En el caso de usar una distribución normal con media μ_l y desviación estándar σ_l

vale $\theta_{\omega_{i,j'}} \equiv (\mu_{\omega_{i,j'}}, \sigma_{\omega_{i,j'}})$.

En este contexto, cabe destacar que el esquema teórico de modelos ocultos de Markov en dos dimensiones no requiere una específica probabilidad de emisión. Cualquier distribución de probabilidad $f(x)$ puede ser usada para modelar la distribución de la observación dado el estado oculto. Las únicas condiciones que tiene que cumplir $f(x)$ son:

$$\begin{aligned} f(x) &> 0 \quad \forall x \in \mathbb{R} \\ \lim_{x \rightarrow -\infty} \int_{-\infty}^x f(\tilde{x}) d\tilde{x} &= 0. \\ \lim_{x \rightarrow \infty} \int_{-\infty}^x f(\tilde{x}) d\tilde{x} &= 1. \end{aligned} \quad (3.16)$$

En lo siguiente, se usa el ejemplo de una distribución normal para mostrar como estimar los parámetros de la función de emisión. Hacemos hincapié que es fácil adaptar el procedimiento a otras funciones de probabilidad como la distribución gamma o Weibull. Para mayor información sobre estas distribuciones se recomienda el libro de Forbes et. al. [73].

Primero supongamos que se observa el estado $l \in \Lambda$ a través de una distribución normal con media μ_l y desviación estándar σ_l . Entonces, la probabilidad de emisión $b_l(x)$ del estado l viene dada por la función de probabilidad.

$$b_l(x_{i,j}) = P(x_{i,j} | \omega_{i,j} = l) = \frac{1}{\sigma_l \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{x_{i,j} - \mu_l}{\sigma_l} \right)^2 \right\} \quad (3.17)$$

Los parámetros μ_l y σ_l de la ecuación (3.17) pueden ser estimados de acuerdo a:

$$\mu_l = \frac{\sum_{i,j} Q_l(i,j) x_{i,j}}{\sum_{i,j} Q_l(i,j)} \quad (3.18)$$

$$\sigma_l = \frac{\sum_{i,j} Q_l(i,j) (x_{i,j} - \mu_l)^2}{\sum_{i,j} Q_l(i,j)} \quad (3.19)$$

El último detalle que falta para completar la descripción teórica de modelos ocultos en dos dimensiones son las probabilidades iniciales. Para el operador “ \prec ”, como definido en la sección 3.1, el píxel (1, 1) marca el inicio del modelo en dos dimensiones. Por lo tanto, tenemos:

$$\pi_l = P(\omega_{1,1} = l) \quad \forall l \in \Lambda. \quad (3.20)$$

Resumiendo esta sección, se ha presentado un esquema teórico que pretende encontrar ω^* según la ecuación (3.12). Como todas las derivaciones están dentro del marco del algoritmo EM tenemos una garantía teórica de que el procedimiento iterativo va a converger. En la próxima sección se muestra el motivo por el cual las cuentas exactas generalmente no son tratables y luego se presentan dos algoritmos en la sección 3.5 que aproximan ω^* haciendo distintas suposiciones.

3.3. Problemas computacionales

Existen varios obstáculos que impiden el cálculo de las variables $H_{m,n,l}$, $a_{m,n,l}$ y Q_l de la sección anterior. Uno de los problemas más grandes aparece en las ecuaciones (3.13) y (3.15) donde hay que ejecutar una suma que abarca todos los posibles mapas de estados ω . Para $u \times v$ píxeles y L estados ocultos, ω contiene $L^{(uv)}$ distintos mapas de estados. Con dos ejemplos se demuestra si una suma con $L^{(uv)}$ elementos es tratable o no. Para eso supongamos que el producto $\prod_{(i',j') \in \mathcal{S}^P} P(x_{i',j'} | \theta_{\omega_{i',j'}}) a_{\omega_{i'-1,j'}, \omega_{i',j'-1}, \omega_{i',j'}}$ se puede calcular en un mili segundo.

Ejemplo 3.1. *Tenemos una imagen muy chica de dimensión 5×5 con dos estados. Entonces ω contiene $2^{25} \approx 3,4 * 10^7$ elementos. Sumar todos estos elementos tardaría un poco más que nueve horas en nuestro escenario.*

Ejemplo 3.2. *Tenemos una imagen chica de dimensión 10×10 con tres estados. Entonces ω contiene $3^{100} \approx 5,2 * 10^{47}$ elementos. Sumar todos estos elementos tardaría más que $1,6 * 10^{37}$ años en nuestro escenario.*

Como se ve en los ejemplos 3.1 y 3.2, las ecuaciones (3.13) y (3.15) no son tratables – salvo para imágenes con muy pocos píxeles y solo dos estados ocultos.

Note que, el hecho de no poder calcular $H_{m,n,l}$ y Q_l inhabilita el cálculo las probabilidades de transición $a_{m,n,l}$ y los parámetros de las distribuciones de emisión de la ecuación (3.17). Por lo tanto, es necesario recurrir a algoritmos subóptimos.

En la próxima sección se presenta un método eficiente para aproximar $a_{m,n,l}$ y los parámetros de las funciones de probabilidad. Luego se introduce el algoritmo estándar para

hallar los estados ocultos en la sección 3.5. En la misma sección se desarrolla un nuevo algoritmo competidor del algoritmo estándar.

3.4. Estimación de parámetros

Para poder estimar los parámetros del modelo oculto en dos dimensiones hay que simplificar las cuentas de la teoría exacta. El primer paso radica en remplazar el esquema exacto – que determina los parámetros en un solo paso – por un esquema iterativo donde cada iteración se acerca un poco más a la solución final.

Como inicio a las aproximaciones iterativas se requiere un mapa inicial de estados ocultos. En lo siguiente se detallan distintos métodos para obtener un mapa inicial.

3.4.1. Mapa inicial de estados

En principio, hay dos opciones para obtener un mapa inicial. Primero, se puede aplicar un algoritmo no supervisado como *k-medias* [74] o *mezcla de gaussianas* [71]. Estos algoritmos sirven para agrupar observaciones multispectrales en una determinada cantidad de clases. Para mayor información acerca de los algoritmos mencionados se recomienda el libro de Bandyopadhyay et. al. [75].

La otra opción se basa en datos de entrenamiento. Si hay información acerca del estado de algunos de los píxeles se pueden ajustar las correspondientes funciones de probabilidad. Luego, se alcanza un mapa inicial comparando la máxima verosimilitud de las distintas clases en cada píxel.

Cabe destacar que los dos métodos no consideran información contextual. En otras palabras, no se tiene en cuenta la relación espacial entre dos píxeles. Sin embargo, es válido iniciar un modelo contextual con un mapa no contextual y luego ajustar las relaciones entre píxeles vecinos en el transcurso de varias iteraciones.

3.4.2. Probabilidades de transición

La idea principal para la estimación de las probabilidades de transición es usar el mapa de estados actual. Si es la primera iteración se toma el mapa inicial. Para las otras

iteraciones se utiliza el mapa de la iteración anterior. Para aproximar $a_{m,n,l}$ en la iteración p se propone

$$a_{n,m,l}^{(p)} = \frac{\sum_{i=2}^u \sum_{j=2}^v I\left(\omega_{i-1,j}^{(p-1)} = n, \omega_{i,j-1}^{(p-1)} = m, \omega_{i,j}^{(p-1)} = l\right)}{\sum_{i=2}^u \sum_{j=2}^v I\left(\omega_{i-1,j}^{(p-1)} = n, \omega_{i,j-1}^{(p-1)} = m\right)}, \quad (3.21)$$

donde $\omega^{(p-1)}$ es el mapa de estados de la iteración anterior. Note que la simplificación propuesta en este trabajo se puede entender como “contar en vez de evaluar” porque solo se cuentan (y normalizan) las transiciones ocurridas en el mapa de estados actual en vez de evaluar todos los posibles mapas de estados. Por lo tanto, este procedimiento tiene un coste computacional muy bajo. Incluso, para imágenes grandes, la ecuación (3.21) puede ser resuelta por una computadora común en pocos segundos.

3.4.3. Funciones de emisión

Para estimar los parámetros de las funciones de emisión se aplica un procedimiento parecido a la ecuación (3.21). Igual que en el caso de aproximar $a_{m,n,l}$ se tiene en cuenta solo el mapa de estados actual. En esta sección se usa el ejemplo de densidades normales para demostrar como estimar los parámetros. Para la media del estado l en la iteración (p) y la desviación estándar se usa

$$\mu_l^{(p)} = \frac{\sum_{i=1}^u \sum_{j=1}^v I\left(\omega_{i,j}^{(p-1)} = l\right) X_{i,j}}{\sum_{i=1}^u \sum_{j=1}^v I\left(\omega_{i,j}^{(p-1)} = l\right)}, \quad (3.22)$$

y

$$\sigma_l^{(p)} = \frac{\sum_{i=1}^u \sum_{j=1}^v I\left(\omega_{i,j}^{(p-1)} = l\right) (X_{i,j} - \mu_l)^2}{\sum_{i=1}^u \sum_{j=1}^v I\left(\omega_{i,j}^{(p-1)} = l\right)}. \quad (3.23)$$

Las ecuaciones (3.22) y (3.23) son los estimadores estándar para μ y σ con la única diferencia que solo se usan las observaciones que actualmente tiene asignado el estado l . En este contexto, es fácil usar densidades que no sean la normal. Por ejemplo, la distribución Gamma o la distribución Weibull son importantes para ciertas aplicaciones como la segmentación de imágenes de radar [76]. Para los estimadores de máxima verosimilitud

de muchas densidades conocidas refiérase a [73]. En la próxima sección se usan todos los parámetros de esta sección para desarrollar decodificadores para modelos ocultos de Markov en dos dimensiones.

3.5. Decodificadores

Los parámetros $a_{m,n,l}$, μ_l y σ_l de la sección anterior son la base para hallar los estados ocultos de las observaciones ruidosas. Sin embargo, el corazón de cualquier algoritmo basado en modelos ocultos de Markov es el decodificador. Aquí es donde se computa toda la información disponible para encontrar un mapa de estados óptimo (o por lo menos subóptimo) según la ecuación (3.12).

En esta sección, se presentan dos posibles decodificadores con distintas características. Por un lado, está el algoritmo estándar llamado *Path Constrained Viterbi Training* (PCVT) que fue estudiado muchas veces por distintos autores [67,68]. Por el otro lado, se desarrolla un nuevo algoritmo denominado *Complete Enumeration Propagation* (CEP). Para ambos decodificadores hay que asumir que la observación de un píxel depende solo del estado oculto de este píxel, pero no de estados ocultos vecinos. Esta suposición es justificable y no es muy restrictivo, sin embargo, hay que mencionarla antes de empezar con la descripción teórica de los dos algoritmos.

3.5.1. Path Constrained Viterbi Training

El algoritmo PCVT aprovecha las características del operador \prec de acuerdo a lo definido en la sección 3.1. Recuerde que \prec determina que la relación causal empieza con el píxel $(1, 1)$ y termina con el píxel (u, v) . En otras palabras, se establece un movimiento de la izquierda arriba hacia la derecha abajo de la imagen. Para describir este movimiento se pueden usar diagonales, donde la diagonal T_1 se refiere al primer píxel, la diagonal T_2 representa los dos píxeles siguientes y así sucesivamente. Formalizando esta notación se obtiene

$$T_1 = (\omega_{1,1}); \quad T_2 = (\omega_{2,1}, \omega_{1,2}); \quad T_3 = (\omega_{3,1}, \omega_{2,2}, \omega_{1,3}); \quad \dots$$

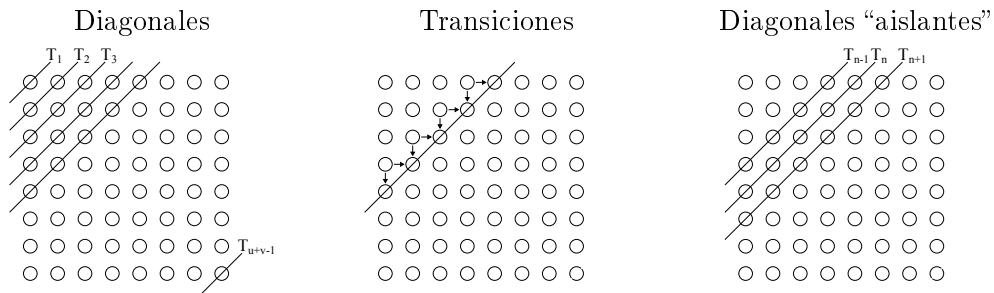


FIGURA 3.3: Diagonales y transiciones del algoritmo PCVT. En este esquema la diagonal T_{n+1} es independiente de la diagonal T_{n-1} dado la diagonal T_n .

Teniendo en cuenta que hay $u+v-1$ diagonales en total se puede expresar la probabilidad conjunta de ω como

$$P(\omega) = P(T_1) P(T_2|T_1) P(T_3|T_2, T_1) \dots P(T_{u+v-1}|T_{u+v-2}, \dots, T_1). \quad (3.24)$$

Ahora podemos aplicar la suposición de Markov para simplificar la ecuación (3.24):

$$\begin{aligned} P(\omega) &= P(T_1) P(T_2|T_1) P(T_3|T_2, T_1) \dots P(T_{u+v-1}|T_{u+v-2}, \dots, T_1) \\ &= P(T_1) P(T_2|T_1) P(T_3|T_2) \dots P(T_{u+v-1}|T_{u+v-2}). \end{aligned} \quad (3.25)$$

Note en la ecuación (3.25) que cada diagonal se puede considerar como un “elemento aislante” que separa la diagonal anterior con la diagonal siguiente. En la figura 3.3 se muestra el efecto de la suposición de Markov.

Lo interesante es que, al aplicar diagonales “aislantes” logramos transformar un modelo de Markov en dos dimensiones a la ecuación (3.25) que se parece mucho a la ecuación estándar de Markov. Recuerde que, en el caso de observaciones de una dimensión se tiene:

$$P(\omega) = P(\omega_1) P(\omega_2|\omega_1) P(\omega_3|\omega_2) \dots P(\omega_{fin}|\omega_{fin-1}) \quad (3.26)$$

Siendo así, se podría usar el algoritmo de Baum-Welch para hallar los estados ocultos. Lamentablemente la complejidad del problema con dos dimensiones es mucho mayor a la del problema unidimensional. Para datos de una dimensión cada elemento de la cadena de Markov está en uno de L estados. En cambio, la diagonal más grande de la ecuación (3.25) contiene $\min(u, v)$ píxeles de los cuales cada uno puede estar en uno de L estados. Entonces, la diagonal entera está en una de $L^{\min(u, v)}$ secuencias formadas por estados

comunes. En el caso de una imagen 50×50 con tres estados, habría aproximadamente $7,2 * 10^{23}$ distintas secuencias de estados sobre la diagonal principal. Este número gigante claramente hace que el algoritmo de Baum-Welch se vuelva intratable incluso para imágenes chicas.

La solución propuesta por el PCVT consiste en cortar las posibles secuencias de cada diagonal a un número manejable. En otras palabras, se descarta la mayoría de las secuencias y se guarda solo unas N secuencias como se muestra en la figura 3.4. Este paso se puede entender como limitar los “caminos” por la imagen que se expresa en la parte “*path constrained*” del nombre del algoritmo.

Lo ideal sería que la selección de las secuencias se efectuará teniendo en cuenta información contextual, pero esto implicaría un alto coste computacional. Por lo tanto, se calculan las máximas verosimilitudes de cada píxel sobre la diagonal para cada estado l por separado según

$$P(\omega_{i,j} = l | X_{i,j}, \theta) \propto P(X_{i,j} | \omega_{i,j} = l, \theta) P(\omega_{i,j} = l | \theta). \quad (3.27)$$

Luego se seleccionan los N secuencias con el criterio de maximizar la verosimilitud de la correspondiente diagonal. Si bien es necesario omitir la relación entre píxeles vecinos en este paso hay que destacar que es solo una simplificación temporal que no se aplica a la parte principal del algoritmo. Sin embargo, es posible descartar algunas secuencias, que en realidad formarían parte del mapa de estados óptimo.

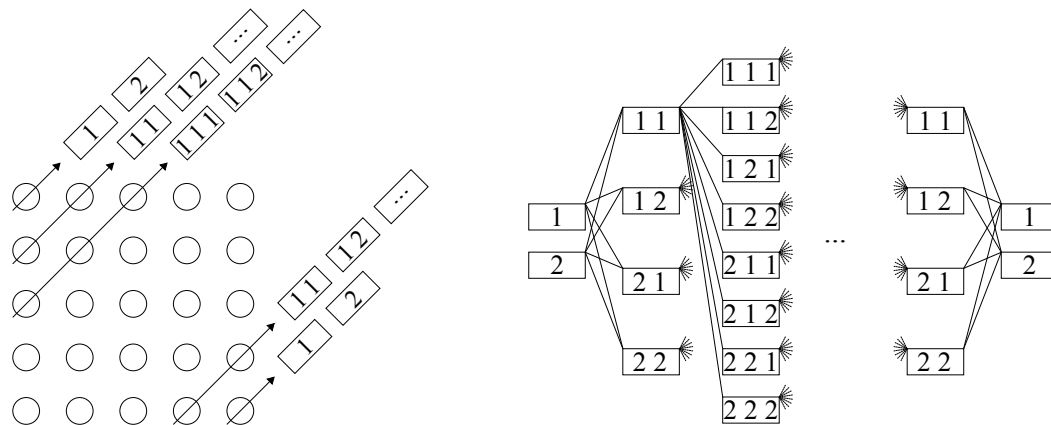
Antes de seguir con el algoritmo PCVT hacemos hincapié en que algunas probabilidades de transición pueden ser igual a cero. En este caso, se borran las correspondientes conexiones entre secuencias como se muestran en la figura 3.5.

Después de haber cortado la cantidad de secuencias por diagonal a N estamos en condiciones de aplicar el algoritmo de Viterbi [77]. Primero, llamamos $\omega_{d,s}$ a la secuencia de estados número s sobre la diagonal d . Además, introducimos la variable $\delta_d(s)$ que es la máxima probabilidad conjunta de las observaciones $\mathbf{X}_1, \dots, \mathbf{X}_d$ y las diagonales T_1 hasta T_d .

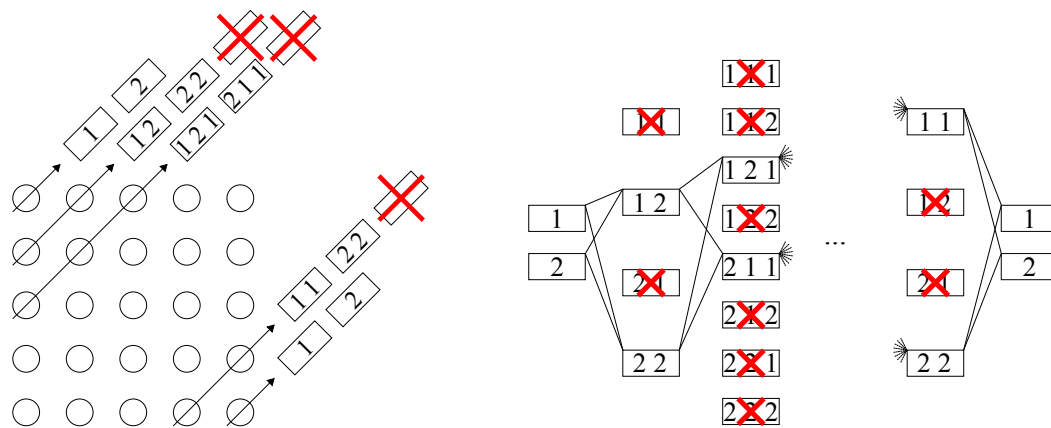
$$\delta_d(s) = \max_{k_1, \dots, k_{d-1}} P(\omega_{1,k_1}, \dots, \omega_{d-1,k_{d-1}}, \omega_{d,s}, \mathbf{X}_1, \dots, \mathbf{X}_d | \theta), \quad (3.28)$$

$$\forall d = 1, \dots, u + v - 1; \quad s = 1, \dots, N.$$

Imagen con dos estados y todas las posibles secuencias



Selección de las dos secuencias más probables por diagonal (o sea $N = 2$)



Solo las secuencias más probables

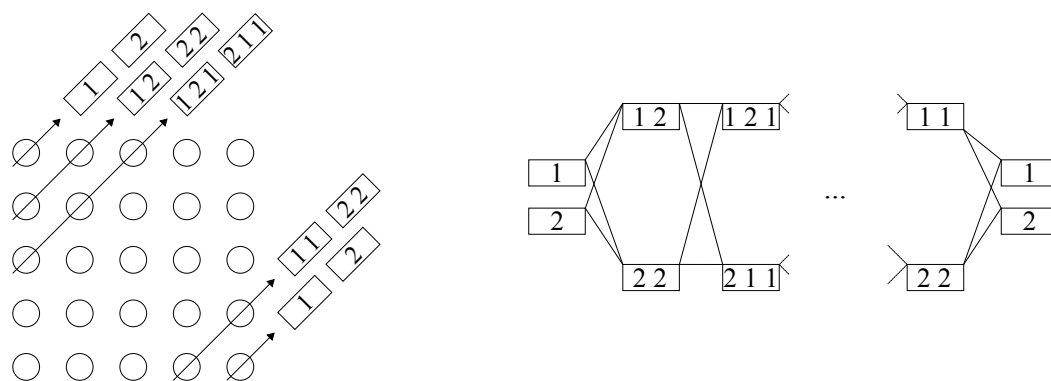


FIGURA 3.4: Achicando la cantidad de secuencias sobre cada diagonal.

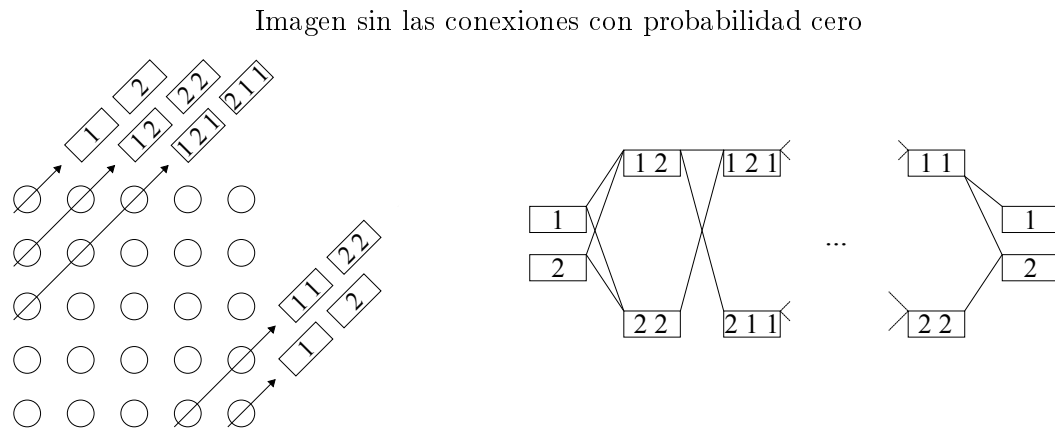
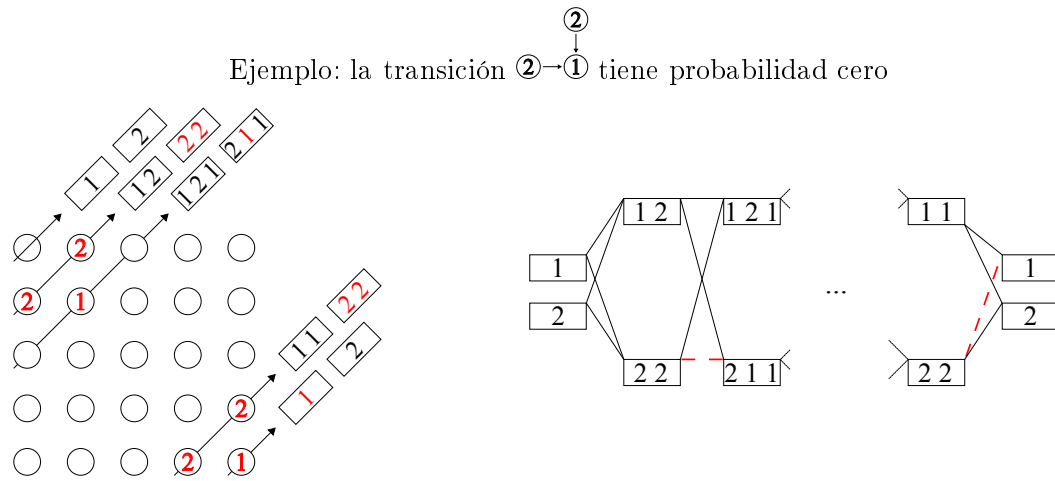


FIGURA 3.5: Conexiones entre secuencias si algunas probabilidades de transición son igual a cero.

En otras palabras, la ecuación (3.28) nos indica cual es el camino más probable para llegar a la secuencia s en la diagonal d . Si por ejemplo, $\delta_4(3)$ es máximo para $k_1 = 2$, $k_2 = 4$ y $k_3 = 1$ sabemos que para llegar a la secuencia 3 sobre la diagonal 4 tenemos que pasar por las secuencias 2 en la primera diagonal, luego secuencia 4 en la segunda diagonal y finalmente secuencia 1 en la tercera diagonal.

Note que, para el caso particular $\delta_1(s)$ no hay diagonales predecesoras. Entonces, se usan las probabilidades iniciales de los estados ocultos que vienen dados por

$$\delta_1(s) = P(T_1 = \omega_{1,s}) \quad \forall s = 1, 2, \dots, N. \tag{3.29}$$

Antes de empezar con el algoritmo de Viterbi hay que definir otras dos variables auxiliares. Por un lado, necesitamos la probabilidad de emitir la observación sobre la diagonal d dada la secuencia s . Esta probabilidad está determinada por

$$b_{\omega_{d,s}}(\mathbf{X}_d) = \prod_{(i,j) \in \Delta(d)} b_{\omega_{d,s}(i,j)}(X_{i,j}), \quad (3.30)$$

donde $\Delta(d)$ representa a todos los píxeles de la diagonal d . Y por último calculamos $\tilde{a}_{d,s_d,s_{d+1}}$ que es la probabilidad de transición de la secuencia s_d sobre la diagonal d a la secuencia s_{d+1} sobre la diagonal $d+1$. Lo único que tenemos que hacer es multiplicar todas las transiciones de los píxeles de la diagonal d a la diagonal $d+1$.

$$\begin{aligned} \tilde{a}_{d,s_d,s_{d+1}} &= P\left(T_{d+1} = \omega_{d+1,s_{d+1}} | T_d = \omega_{d,s_d}, \theta\right) \\ &= \prod_{(i,j) \in \Delta(d+1)} a_{\omega_{d,s_d}(i-1,j), \omega_{d,s_d}(i,j-1), \omega_{d+1,s_{d+1}}(i,j)} \\ &\forall d = 1, \dots, u+v-2; \quad s_d, s_{d+1} = 1, \dots, N. \end{aligned} \quad (3.31)$$

Ahora podemos usar las ecuaciones (3.28), (3.30) y (3.31) para definir el esquema recursivo del algoritmo de Viterbi. La idea es ir avanzando de una diagonal a la siguiente usando los resultados guardados en $\delta_d(s)$. Lo que más nos interesa en ese contexto es determinar la secuencia predecesora (“ k ”) más probable dado la secuencia actual (“ s ”) y teniendo en cuenta las variables $\delta_d(k)$ y $\tilde{a}_{d,k,s}$. Para la secuencia s sobre la diagonal $d+1$ obtenemos

$$\begin{aligned} \delta_{d+1}(s) &= \left[\max_{1 \leq k \leq N} \delta_d(k) \tilde{a}_{d,k,s} \right] b_{\omega_{d+1,s}}(\mathbf{X}_{d+1}) \\ &\forall d = 1, \dots, u+v-2 \quad \forall s = 1, 2, \dots, N. \end{aligned} \quad (3.32)$$

Note en la ecuación (3.32), que el termino en corchetes usa las probabilidades máximas de la diagonal anterior ($\delta_d(k)$) en conjunto con las probabilidades de transición ($\tilde{a}_{d,s_d,s_{d+1}}$) para incorporar información contextual mientras el termino $b_{\omega_{d+1,s_{d+1}}}$ se ocupa de evaluar las probabilidades de emisión. Lo importante es que $\delta_d(s)$ contiene toda la información acerca del camino más probable desde la diagonal T_1 hasta la diagonal T_d . Entonces, se reduce significativamente la complejidad computacional. Por ejemplo, una imagen de dimensión 50×50 tiene 99 diagonales. Si fijamos $N = 10$ hay que calcular $99 * 10^2 = 9900$ veces $\delta_d(s)$ (la transición de una diagonal con 10 secuencias a la siguiente diagonal con

10 secuencias implica 10^2 operaciones). O, de forma general, el algoritmo tiene una complejidad del orden $\mathcal{O}((uv - 1)N^2)$.

Después de calcular $\delta_{d+1}(s)$ según ecuación (3.32) se busca el índice k (de la diagonal d) que lleva a la secuencia s (de la diagonal $d + 1$) con la probabilidad más alta y se guarda este índice en una variable φ :

$$\begin{aligned} \varphi_{d+1}(s) &= \arg \max_{1 \leq k \leq N} \{\delta_d(k) \tilde{a}_{d,k,s}\} \\ \forall \quad d &= 1, \dots, u + v - 2 \quad s = 1, 2, \dots, N \end{aligned} \quad (3.33)$$

En la primera parte de la figura 3.6 se muestra ejemplarmente que significa la ecuación (3.33). Cuando el algoritmo recursivo llega a la última diagonal $u + v - 1$ (que es un solo píxel) hay que buscar el estado oculto $\omega_{u,v}$ más probable de este píxel:

$$\omega_{u,v} = \arg \max_{1 \leq k \leq L} \{\delta_{u+v-1}(k)\}. \quad (3.34)$$

A partir de este estado $\omega_{u,v}$ se puede usar la variable φ que nos indica cual es la secuencia predecesora más probable del estado $\omega_{u,v}$.

$$\omega_d^* = \varphi_{d+1}(\omega_{d+1}^*) \quad \forall d = (u + v - 2), (u + v - 3), \dots, 1. \quad (3.35)$$

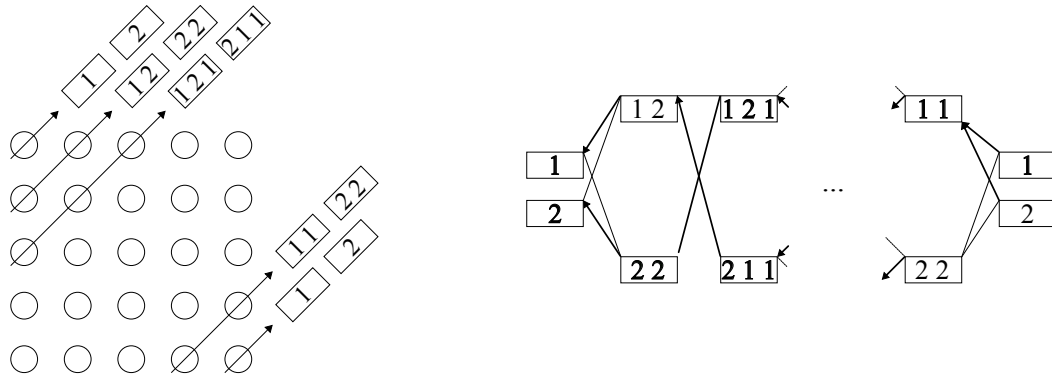
Si aplicamos la ecuación (3.35) de forma recursiva volvemos por las diagonales hasta llegar a la diagonal T_1 que representa al píxel (1,1). Por consiguiente, hemos logrado hallar el mapa de estados ocultos ω^* con el algoritmo de Viterbi. En la segunda parte de la figura 3.6, se muestra un ejemplo para esta parte del algoritmo.

Un esquema completo del PCVT como detallado en esta sección (siguiendo en partes el desarrollo de Joshi et. al. [67]) se presenta en el algoritmo 1. Cabe destacar que solo el paso 5) del algoritmo 1 tiene un notable coste computacional del orden $\mathcal{O}((uv - 1)N^2)$. Todos los otros pasos tienen una complejidad insignificante.

3.5.2. Complete Enumeration Propagation

En esta sección se presenta un nuevo algoritmo llamado *Complete Enumeration Propagation* (CEP). En contrario al PCVT, el CEP no usa el Viterbi Training para hallar

Algoritmo de Viterbi - Parte 1



Algoritmo de Viterbi - Parte 2

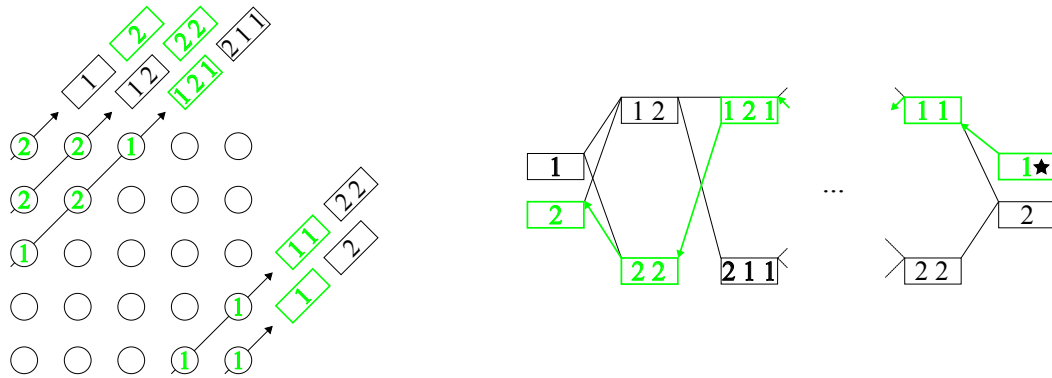


FIGURA 3.6: Los dos pasos de algoritmo de Viterbi. Primero, determinar los predecesores más probables de cada secuencia. Segundo, seleccionar el camino más probable.

Algoritmo 1: Path-Constrained Viterbi Training (PCVT)

1. Inicializar mapa de estados
 2. Estimar parámetros de las funciones de probabilidad (θ)
 3. Calcular a_{nml} según ecuación (3.21)
 4. Elegir las N mejores secuencias para cada diagonal usando máxima verosimilitud
 5. Correr el algoritmo de Viterbi según las ecuaciones (3.32), (3.33), (3.34) y (3.35)
 6. Iterar los pasos 2), 3), 4) y 5) hasta convergencia
-

el mapa de estados ocultos y tampoco descarta secuencias poco probables. Sin embargo, CEP hace más suposiciones acerca de píxeles vecinos que PCVT, como se detalla a continuación.

Mientras PCVT usa secuencias sobre diagonales para desarrollar su teoría, CEP pone un enfoque en las probabilidades de estado de cada píxel por separado. La idea principal de CEP consiste en propagar estas probabilidades desde el píxel $(1, 1)$ hasta el píxel (u, v) según la notación \prec .

La probabilidad de estado del píxel (i, j) viene dado por $P(\omega_{i,j}|\omega_{i,j-1}, \omega_{i-1,j}, X_{i,j})$. Aplicando la regla de Bayes y dejando de lado el termino de normalización obtenemos

$$P(\omega_{i,j}|\omega_{i,j-1}, \omega_{i-1,j}, X_{i,j}) \propto P(\omega_{i,j}, \omega_{i,j-1}, \omega_{i-1,j}, X_{i,j}). \quad (3.36)$$

Ahora podemos expandir la probabilidad conjunta en sus respectivas partes

$$P(\omega_{i,j}, \omega_{i,j-1}, \omega_{i-1,j}, X_{i,j}) = P(\omega_{i,j-1}, \omega_{i-1,j})P(\omega_{i,j}|\omega_{i,j-1}, \omega_{i-1,j})P(X_{i,j}|\omega_{i,j}). \quad (3.37)$$

En este punto se considera que la probabilidad de transición no depende de la ubicación en la imagen según la ecuación (3.11) y la figura 3.2.

$$\begin{aligned} & P(\omega_{i,j-1}, \omega_{i-1,j}) P(\omega_{i,j}|\omega_{i,j-1}, \omega_{i-1,j}) P(X_{i,j}|\omega_{i,j}) \\ &= P(\omega_{i,j-1}, \omega_{i-1,j}) a_{\omega_{i,j-1}, \omega_{i-1,j}, \omega_{i,j}} P(X_{i,j}|\omega_{i,j}). \end{aligned} \quad (3.38)$$

Note que hasta aquí se implementaron las mismas suposiciones que para el algoritmo PCVT. Ahora, es necesario hacer otra suposición para que el algoritmo CEP sea tratable. Por consiguiente, se asume que los dos píxeles del “pasado” son independientes. Recuerde que el PCVT necesita esta suposición para el primer paso donde se detectan las N secuencias más probables, pero no para el resto del algoritmo. Con la suposición mencionada se puede simplificar la ecuación (3.38).

$$\begin{aligned} & P(\omega_{i,j-1}, \omega_{i-1,j}) a_{\omega_{i,j-1}, \omega_{i-1,j}, \omega_{i,j}} P(X_{i,j}|\omega_{i,j}) \\ &= P(\omega_{i,j-1})P(\omega_{i-1,j}) a_{\omega_{i,j-1}, \omega_{i-1,j}, \omega_{i,j}} P(X_{i,j}|\omega_{i,j}). \end{aligned} \quad (3.39)$$

La ecuación (3.39) es la ecuación principal para el algoritmo CEP ya que sirve para

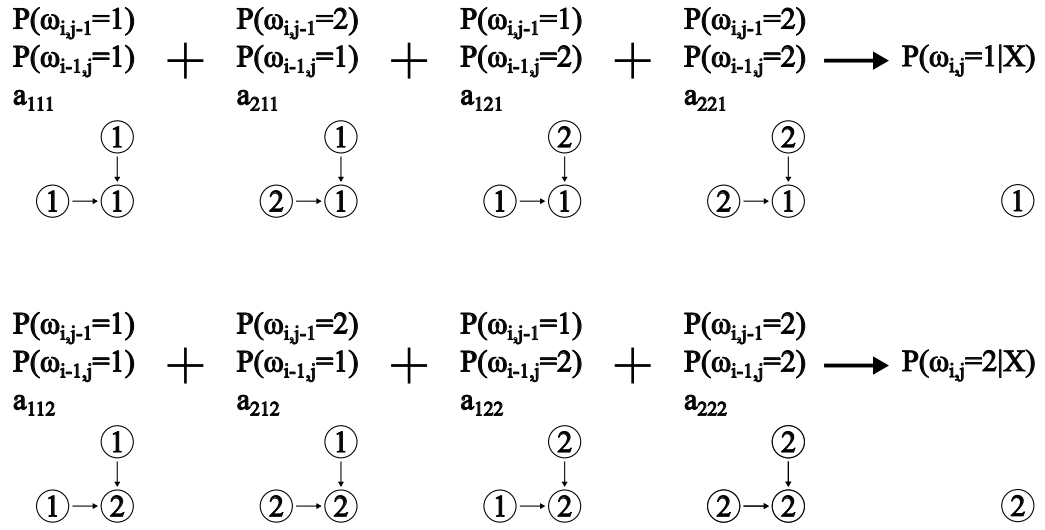


FIGURA 3.7: Ejemplo con dos estados para mostrar que significa la ecuación (3.41).

calcular las probabilidades de estado del píxel (i, j) dado la observación $X_{i,j}$, las probabilidades de transición y los estados del pasado $(i-1, j)$ y $(i, j-1)$.

El próximo paso del CEP consiste en computar $P(\omega_{i,j} = l|\mathbf{X}) \forall l = 1, 2, \dots, L$. Para eso resumimos las ecuaciones (3.36) hasta (3.39) y obtenemos

$$P(\omega_{i,j}|\omega_{i,j-1}, \omega_{i-1,j}, X_{i,j}) = P(\omega_{i,j-1})P(\omega_{i-1,j}) a_{\omega_{i,j-1}, \omega_{i-1,j}, \omega_{i,j}} P(X_{i,j}|\omega_{i,j}). \quad (3.40)$$

Lo interesante de la ecuación (3.40) es que para una finita cantidad de estados (en este caso se trabaja con L estados) los términos $P(\omega_{i,j-1})$, $P(\omega_{i-1,j})$ y $a_{\omega_{i,j-1}, \omega_{i-1,j}, \omega_{i,j}}$ toman un valor de un conjunto finito. Entonces, es posible enumerar todas las posibles combinaciones de estados del “pasado” que vienen dada por $\omega_{i,j-1} = m; \omega_{i-1,j} = n \forall m, n = 1, 2, \dots, L$. Sumando sobre todas las combinaciones y sus respectivas probabilidades se obtiene

$$P(\omega_{i,j} = l|\mathbf{X}) \propto \sum_{m=1}^L \sum_{n=1}^L P(\omega_{i,j-1} = m) P(\omega_{i-1,j} = n) a_{\omega_{i,j-1}=m, \omega_{i-1,j}=n, \omega_{i,j}=l} P(X_{i,j}|\omega_{i,j} = l). \quad (3.41)$$

Note que, la ecuación (3.41) representa una enumeración completa de todos los términos que son necesarios para calcular $P(\omega_{i,j} = l|\mathbf{X})$. En la figura 3.7 se muestra un ejemplo de la ecuación (3.41) para el caso particular de dos estados.

Después de calcular las probabilidades de estado del píxel (i, j) es necesario normalizar las probabilidades usando

$$\sum_{l=1}^L P(\omega_{i,j} = l | \mathbf{X}) = 1. \quad (3.42)$$

Resumiendo la teoría de CEP constatamos que se pueden calcular las probabilidades de estado de un píxel (i, j) usando las ecuaciones (3.40), (3.41) y (3.42). Luego se pueden usar estas probabilidades para computar las probabilidades de estado de los siguientes píxeles $(i + 1, j)$ y $(i, j + 1)$. Entonces, aplicando este método a todos los píxeles de la imagen sucesivamente (respetando el orden causal establecido por \prec), se obtiene un mapa de probabilidades.

Una vez computado el mapa de probabilidades es fácil encontrar el mapa de estados ω^* . Solo hay que elegir el estado con mayor probabilidad en cada píxel:

$$\omega_{i,j}^* = \arg \max_{k=1,2,\dots,L} \{P(\omega_{i,j} = k | \mathbf{X})\}. \quad (3.43)$$

Cabe destacar que por las suposiciones realizadas ω^* no es un óptimo global como definido en la ecuación (3.12). Sin embargo, lo consideramos el mapa más probable en la iteración actual para la configuración usada.

En el algoritmo 2 se presenta una descripción detallada del algoritmo CEP. Note que, la complejidad del algoritmo es del orden $\mathcal{O}((uv)L^3)$. Por lo tanto, el coste computacional de CEP depende fuertemente de la cantidad de estados. Esta y otras propiedades del algoritmo se discuten en la sección 3.7.

Algoritmo 2: Complete Enumeration Propagation (CEP)

1. Inicializar mapa de estados
 2. Estimar parámetros de las funciones de probabilidad (θ)
 3. Calcular a_{nml} según ecuación (3.21)
 4. Hallar nuevo mapa de estados usando las ecuaciones (3.40), (3.41), (3.42) y (3.43)
 5. Iterar los pasos 2), 3) y 4) hasta convergencia
-

3.6. Funciones de emisión

Como se ha mencionado en distintas oportunidades de este capítulo, el marco teórico de los modelos ocultos de Markov en dos dimensiones no requiere una cierta distribución de probabilidad. Cualquier función que cumpla con los requisitos de la ecuación (3.16) puede ser usada como función de emisión. Entonces surge la pregunta: ¿Cómo puedo encontrar la función de emisión óptimo?

Para algunos problemas específicos existen distribuciones particulares que han demostrado su capacidad de representar las observaciones. Un ejemplo, son las imágenes de radar que muchas veces son modeladas con distribuciones *Gamma* [76]. Sin embargo, es difícil determinar la óptima función de emisión para una tarea genérica – sobre todo porque muchas densidades pueden cambiar completamente su comportamiento al cambiar suavemente sus parámetros. En la figura 3.8 se muestra algunos ejemplos de distribuciones que cambian su curvatura para distintos parámetros.

En esta tesis se enfrenta el problema de la función de emisión desde dos puntos de vista. Por un lado, se corren todos los algoritmos con distintas distribuciones y luego se comparan los resultados. Por el otro, se desarrollan tres métodos para estimar la función de emisión “online” después de cada iteración.

Para entender cómo funciona el esquema “online” supongamos que finalizó la iteración i . En este punto tenemos un mapa de estados y las observaciones. Estas dos informaciones juntas se pueden usar como datos de entrenamiento porque el mapa de estados indica a que clase pertenece cada observación. La idea ahora es evaluar cuál es la densidad que mejor modela las observaciones de la clase uno, dos, etc. Hacemos hincapié que es admisible usar distintas distribuciones para cada clase. O sea, las observaciones de la clase uno podrían ser representadas por una distribución *Weibull* mientras las clase dos se modela usando una *inverse Gaussian*.

Después de elegir una función de emisión se estiman sus parámetros usando las correspondientes observaciones. Luego, se sigue el esquema general que consiste en calcular las probabilidades de transición y correr uno de los dos decodificadores de la sección 3.5.

Si bien hay una gran variedad de distribuciones en la literatura [73], nos enfocamos en siete densidades importantes que tienen características particulares. Se trata de las

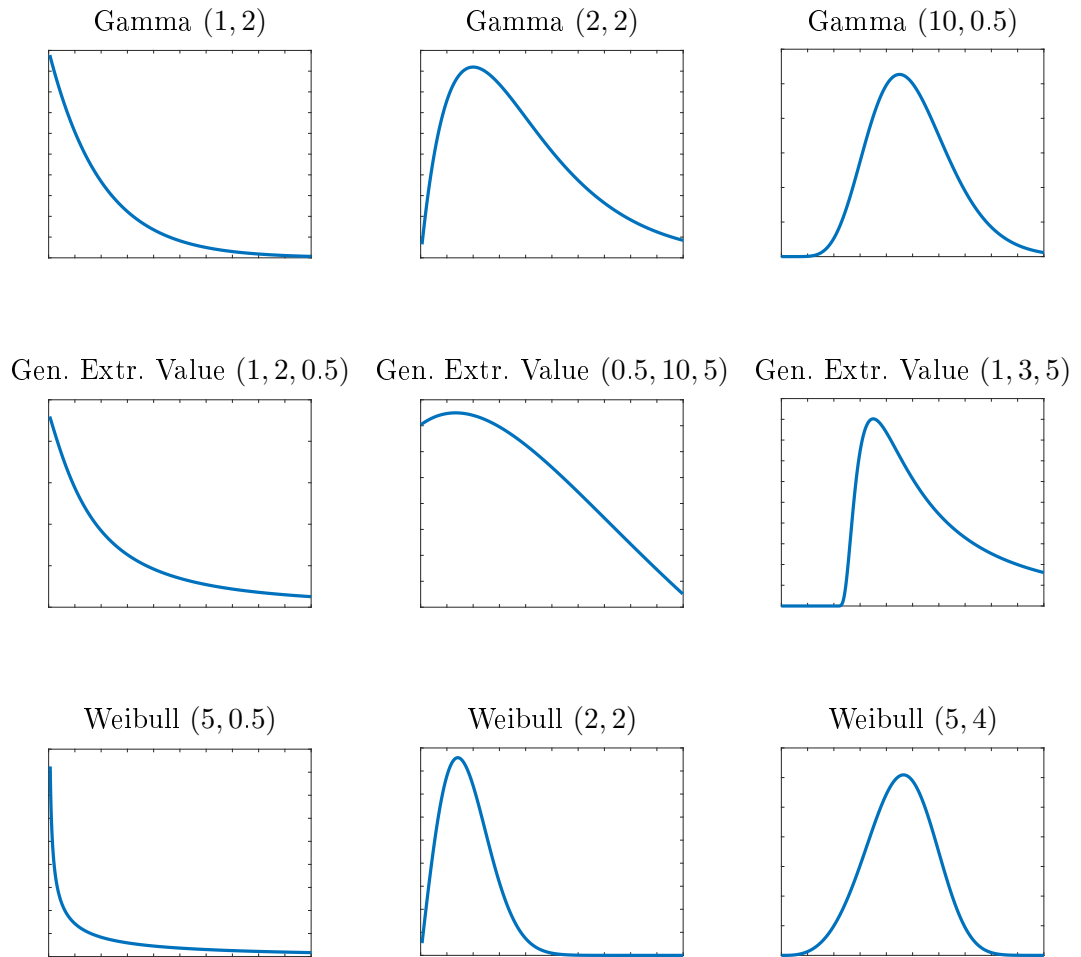


FIGURA 3.8: Curvaturas de tres distribuciones para distintos parámetros (en paréntesis). Primera línea: la distribución *Gamma*. Segunda línea: la distribución *Generalized Extreme Value*. Tercera línea: la distribución *Weibull*.

siguientes funciones: *Gamma*, *Generalized Extreme Value*, *inverse Gaussian* (también conocida como *Wald distribution*), *Kernel*, *Nakagami*, *Normal* y *Weibull*.

En lo siguiente se presentan tres métodos para encontrar la distribución óptima dado datos de entrenamiento. Primero, supongamos que contamos con un conjunto $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ de familias de distribuciones, donde cada familia es una de las siete densidades mencionadas en el párrafo anterior. Cada una de estas funciones tiene la forma $\mathcal{F}_i = f_{i,\theta} : \theta \in \Theta_i$, donde Θ_i es el espacio de parámetros de la familia \mathcal{F}_i . Por ejemplo, una \mathcal{F}_i puede ser la familia de distribuciones normales y Θ_i es el conjunto que contiene todos los posibles parámetros de la familia seleccionada.

3.6.1. Teorema de Bayes

El primer método para encontrar la densidad adecuada para un conjunto de datos está basado en el Teorema de Bayes. La selección de la función de probabilidad se realiza a partir de

$$f_l = \arg \max_{1 \leq i \leq n, \theta \in \Theta_i} \prod_{\omega_{i,j}=l} f_{i,\theta}(X_{i,j}), \quad (3.44)$$

siendo $f_{i,\theta}(X_{i,j})$ la densidad de probabilidad de observar $X_{i,j}$ dado que el modelo de emisión corresponde a la familia \mathcal{F}_i con parámetro $\theta \in \Theta_i$. Entonces, la función de emisión óptima según este criterio es la densidad f_l .

3.6.2. Test de Kolmogorov-Smirnov

La idea clave de este método es usar el valor p del test de Kolmogorov-Smirnov con dos muestras (KS) [78]. Primero, se ajustan los parámetros de las funciones candidatas a las observaciones.

$$f_i = \arg \max_{\theta \in \Theta_i} \prod_{\omega_{i,j}=l} f_{i,\theta}(X_{i,j}). \quad (3.45)$$

Luego, se toman muestras de las distribuciones ajustadas y se calculan los valores p entre las muestras y las observaciones. Al final, se elige la distribución cuyo valor p es el más alto.

3.6.3. Divergencia de Kullback-Leibler

El último método para seleccionar una distribución para cada clase hace uso de la divergencia de Kullback-Leibler (KL) [79]. Igual que en el test de Kolmogorov-Smirnov el primer paso consiste en ajustar las posibles funciones a las observaciones según la ecuación (3.45). Además, se construyen varios histogramas con distintos intervalos. Luego, se computa la divergencia de Kullback-Leibler entre las funciones ajustadas y los histogramas. Finalmente se promedian las divergencias de Kullback-Leibler para cada función candidato y se elige la función que tiene la divergencia promedio más chica. El razonamiento detrás este método es que mientras más chica es la divergencia entre la densidad y las observaciones (representadas por histogramas) mejor es el modelo.

En la siguiente sección se comparan los algoritmos PCVT y CEP con respecto a criterios teóricos como las suposiciones y su coste computacional. Además, se evalúan todos los métodos de esta sección con respecto a sus resultados experimentales en el capítulo 7.

3.7. Comparación de algoritmos

En este capítulo se presentó el esquema general de modelos ocultos de Markov en dos dimensiones y se puso énfasis en los algoritmos decodificadores PCVT y CEP. Ambos algoritmos usan las mismas variables auxiliares que son las funciones de emisión y las probabilidades de transición a_{nml} . Cabe destacar que la estimación de los parámetros de las densidades y el computo de a_{nml} tiene un coste computacional insignificante. Por lo tanto, no enfocamos en la complejidad del PCVT y del CEP.

El orden del PCVT es de $\mathcal{O}((uv - 1)N^2)$ para las N secuencias más probables mientras el orden de CEP es $\mathcal{O}((uv)L^3)$ para L estados ocultos. Note que el orden de PCVT depende de forma cuadrática de un parámetro N elegido por el usuario. Por un lado, esto se puede considerar una ventaja ya que el usuario puede elegir entre calidad del resultado (para N grande) y el coste computacional (para N chico). Por el otro, N es un extra parámetro que puede llevar a resultados malos o tiempos de computación muy largos. En este sentido, es preferible un algoritmo como el CEP que no necesita la interacción con un usuario. Sin embargo, el CEP tiene un coste computacional alto para tareas con muchos estados por el termino L^3 .

Otro aspecto de los algoritmos son las suposiciones que necesitan. El defecto de PCVT es claramente la pre-selección de N secuencias sobre cada diagonal. En este paso, es posible de descartar secuencias óptimas. No obstante, hay que tener en cuenta que se trata de un algoritmo iterativo. Por lo tanto, una secuencia óptima puede ser descartada en una iteración, pero no en la siguiente. Analizando el algoritmo CEP resalta la suposición de considerar los dos píxeles del “pasado” independientes a la hora de calcular las probabilidades de estado. Por otra parte, esta crítica puede ser debilitada por el hecho que CEP considera las probabilidades de transición a_{nml} que representan la probabilidad conjunta entre los dos píxeles del “pasado” y el píxel actual.

Como reflexión final se destaca que el algoritmo CEP es notablemente más fácil de implementar que el algoritmo PCVT. Mientras CEP puede ser programado con algunos

bucles *for*, PCVT requiere habilidades más avanzadas para procesar las secuencias sobre las diagonales.

En el próximo capítulo se introducen los modelos de Potts. Igual que en este capítulo se presenta el algoritmo estándar y un algoritmo competidor novedoso.

Capítulo 4

Realizaciones del modelo Potts

En 1925 Ernst Ising desarrolló en su tesis doctoral [80] el llamado *Ising model*. Este modelo describe las interferencias de las cargas $+$ y $-$ en el ferromagnetismo. Por lo tanto, el Ising model considera solo dos estados. Unos 27 años después, Renfrey Potts extendió la teoría del Ising model a un número finito de estados [81]. Basado en el trabajo de Potts se han desarrollado una gran cantidad de algoritmos y métodos para distintas tareas como la evolución de cáncer [82] o inestabilidad de espumas (importante para la lucha contra incendios) [83]. Sin embargo, una de las aplicaciones más frecuentes es el procesamiento de imágenes [84,85].

En la próxima sección se presenta el esquema general de modelos de Potts. Luego, se introduce el algoritmo estándar y un competidor desarrollado en esta tesis. Al final del capítulo se comparan los dos algoritmos.

4.1. Definición general

En la ecuación (2.8) de la sección 2.3.2 se introdujo el potencial $\Phi_{\mathcal{S}}^{\beta}(\boldsymbol{\omega})$ de un modelo de Potts isotrópico sin campo externo. La función de energía de este potencial (a veces llamado hamiltoniano) es la suma de $\Phi_{\mathcal{S}}^{\beta}(\boldsymbol{\omega})$ para todos los pares $\{s, t\} \in \mathcal{S}$

$$H_{\mathcal{S}}^{\beta}(\boldsymbol{\omega}) := \sum_{\{s,t\} \in \mathcal{S}} \Phi_{\{s,t\}}^{\beta}(\{\omega_s, \omega_t\}) = -\beta H_{\mathcal{S}}(\boldsymbol{\omega}), \quad (4.1)$$

donde $H_S(\boldsymbol{\omega})$ es la cantidad de pares de vecinos (sin importar el orden) con la misma etiqueta. Recuerde en este contexto que para la mayoría de los conjuntos $\{s, t\} \in \mathcal{S}$ vale $s \notin \mathcal{N}_t$. O sea, s y t no son vecinos y por lo tanto la cantidad de pares de vecinos con la misma etiqueta es cero.

De acuerdo a la función de energía de la ecuación (4.1) constatamos que la densidad condicional del campo aleatorio de Gibbs está dada por

$$\pi_S^\beta(\omega_S | \omega^S) \propto \exp\{\beta H_s(\boldsymbol{\omega})\}. \quad (4.2)$$

En la ecuación (4.2), ω^S representa todos los estados ocultos salvo ω_S .

En resumen, las ecuaciones (4.1) y (4.2) expresan formalmente lo que se podría articular de siguiente manera: Para conocer la probabilidad condicional de un píxel hay que mirar (y contar) los estados ocultos en su vecindad. Por ende, la definición de una vecindad es fundamental para un modelo de Potts. Pero en contrario a los modelos causales, no es necesario que esta vecindad cumpla con una relación causal. Entonces, cualquier vecindad es admisible. Sin embargo, el coste computacional aumenta al usar vecindades grandes. Por consiguiente, nos limitamos a las vecindades de la figura 2.3 en este trabajo.

En las próximas dos secciones se presentan dos algoritmos que son realizaciones del modelo de Potts. Primero se detalla el algoritmo estándar y luego se desarrolla un nuevo algoritmo.

4.2. Iterated Conditional Modes

Dado la función de energía de la ecuación (4.1) la pregunta es: ¿Cómo encontrar el máximo de la probabilidad conjunta del campo aleatorio de Gibbs? O, en otras palabras, ¿Cómo hallar un mapa de estados tal que se maximicen las probabilidades condicionales?

Para resolver este problema, Kirkpatrick desarrolló el algoritmo *Simulated Annealing* (SA) [86]. Si bien SA garantiza converger al máximo global, tiene un coste computacional muy alto. Por consiguiente, no se puede aplicar el algoritmo a imágenes medianas y grandes.

Para reducir la alta complejidad del algoritmo SA, Besag propuso la siguiente simplificación: en vez de considerar todos los posibles mapas de estados se puede implementar un esquema iterativo donde en cada iteración se tiene en cuenta solo el mapa de estados actual. Note que dicha simplificación es básicamente la misma que usan los modelos causales para reducir su complejidad computacional.

La ventaja del algoritmo de Besag – llamado *Iterated Conditional Modes* (ICM) [33] – es que reduce drásticamente el coste computacional. La desventaja es que ya no hay garantía de converger al máximo global. Por lo menos, se pueden derivar dos condiciones bajo las cuales ICM converge al máximo local más cercano del mapa inicial. Pero antes de presentar estas dos condiciones es necesario describir cómo funciona ICM.

La idea principal de ICM consiste en actualizar el estado de cada píxel (i, j) en cada iteración teniendo en cuenta dos factores. Primero, la evidencia radiométrica que es la probabilidad de la observación dado el estado oculto l : $p(X_{i,j}|l, \mu_l, \Sigma_l)$. Y segundo, la información contextual que son los estados ocultos de los píxeles vecinos. En este contexto, la función de energía de la ecuación (4.1) define la influencia de los píxeles vecinos.

Juntando la evidencia radiométrica y la información contextual se asigna al píxel (i, j) el estado l que maximiza

$$g(l) = p(X_{i,j}|l, \mu_l, \Sigma_l) \exp \beta U_{i,j}(l). \quad (4.3)$$

En la ecuación (4.3), $U_{i,j}(l)$ es la cantidad de píxeles de la vecindad del píxel (i, j) con estado oculto l . Entonces $U_{i,j}(l)$ incorpora información contextual junto con el parámetro de suavidad β . Para $\beta > 0$ vale: mientras más píxeles vecinos tengan el estado oculto l , más grande $U_{i,j}(l)$ y por lo tanto más probable que el píxel (i, j) también tome el estado l . Al mismo tiempo, el parámetro β juega un rol importante. Si $\beta = 0$, no se tiene en cuenta información contextual. En cambio, $\beta \rightarrow \infty$ implica que las observaciones no influyen en absoluto en el proceso de segmentación.

Como el parámetro β es generalmente desconocido, hay que determinarlo de alguna manera. Básicamente hay dos opciones.

1. El usuario fija algún valor para β

2. Se estima el valor de β

En el caso que el usuario elija β , es recomendable tener en cuenta la estructura esperada de la imagen. En grandes rasgos se puede decir que si se trata de una imagen con muchos segmentos chicos es preferible elegir $\beta < 0,5$. En cambio, si la imagen tiene parches grandes, el valor de β puede variar entre $1 < \beta < 2$.

Si se opta por estimar el valor de β hay que tener en cuenta que aparecen ciertos obstáculos en el camino. Como se mencionó en esta sección, β es una medida de suavidad de los estados ocultos. Pero como no conocemos el verdadero mapa de estados ocultos no podemos conocer β . La simplificación que aplica la mayoría de los algoritmos de estimación es el uso de un mapa inicial de estados. Sin embargo, no se puede usar ICM para obtener este mapa inicial de estados ya que no conocemos β . Por esto, el mapa inicial se obtiene generalmente con métodos no contextuales como mezcla de gaussianas. Cabe destacar que segmentaciones no contextuales son generalmente más granuladas que segmentaciones contextuales. Por lo tanto, hay una tendencia de estimar un valor de β más chico de lo que realmente es.

Note, que se puede mejorar la estimación de β si se re-estima el parámetro después de cada iteración. O sea, se usa un mapa inicial de estados que es no contextual en la primera iteración, pero a partir de la segunda iteración el mapa de estados ya tiene incorporado información contextual (por la ecuación (4.3)). Por lo tanto, el valor estimado es generalmente más alto desde la segunda iteración en adelante.

En la literatura existen dos estimadores de β que se basan en la estimación de pseudo máxima verosimilitud (se dice *pseudo* verosimilitud por los problemas de estimación que se detallaron arriba). Por un lado, Frery [87] y Levada [88] propusieron un estimador “a priori”. El nombre del estimador viene dado por su característica de usar solo un mapa inicial de estados, pero no las correspondientes observaciones. O sea, el estimador se basa en un modelo *a priori* sin conocer muestras de datos. En cambio, el estimador “a posteriori” – como propuesto por Gimenez et al. [89] – usa un mapa inicial de estados juntos con las observaciones.

En [90] se demuestra que el estimador “a posteriori” logra mejores resultados que el estimador “a priori” para una amplia variedad de problemas de segmentación. Por lo tanto, se usa el estimador “a posteriori” en este trabajo.

Hasta aquí se presentó el esquema general del algoritmo ICM. El paso fundamental consiste en actualizar el estado de cada píxel según la ecuación (4.3). Solo falta responder la siguiente pregunta: ¿En qué orden se actualizan los estados de los píxeles? O, en otras palabras: ¿Cómo definir el esquema de visita?

A primera vista, el esquema de visita no es una parte crucial de ICM. Sin embargo, el esquema de visita influye en el resultado final y juega un rol importante a la hora de determinar si ICM converge o no. Para que ICM converja a un máximo local el esquema de visita tiene que cumplir los siguientes puntos:

1. Hay que actualizar el estado de cada píxel por lo menos una vez por iteración
2. Dos píxeles visitados sucesivamente no pueden tener vecinos en común

Estas dos condiciones fueron comprobadas por Geman et al. en el Teorema A de su trabajo [91]. Si no se cumple una de las dos condiciones, se observa un comportamiento oscilatorio del mapa de estados a partir de una determinada iteración. Esto significa que, los estados ocultos de algunos píxeles van alternando entre dos o más estados.

En este trabajo se utiliza un esquema de visita clásica que consiste en una sucesión de barridas en grillas equidistantes. La cantidad de barridas necesarias se define en función del tamaño de la vecindad. Por ejemplo, para una vecindad 3×3 se usan 9 grillas, mientras que una vecindad del tamaño 5×5 requiere 25 grillas. Estos dos ejemplos se muestran en las figuras 4.1 y 4.2.

Algoritmo 3: Iterated Conditional Modes (ICM)

1. Inicializar mapa de estados
 2. Estimar parámetros μ_l y Σ_l para todos $l \in \Lambda$
 3. Estimar o definir parámetro β
 4. Fijar esquema de visita
 5. Actualizar el estado oculto de cada píxel (i, j) según ecuación (4.3)
 6. Opcional: re-estimar el parámetro β
 7. Iterar el paso 5) y posiblemente el paso 6) hasta convergencia
-

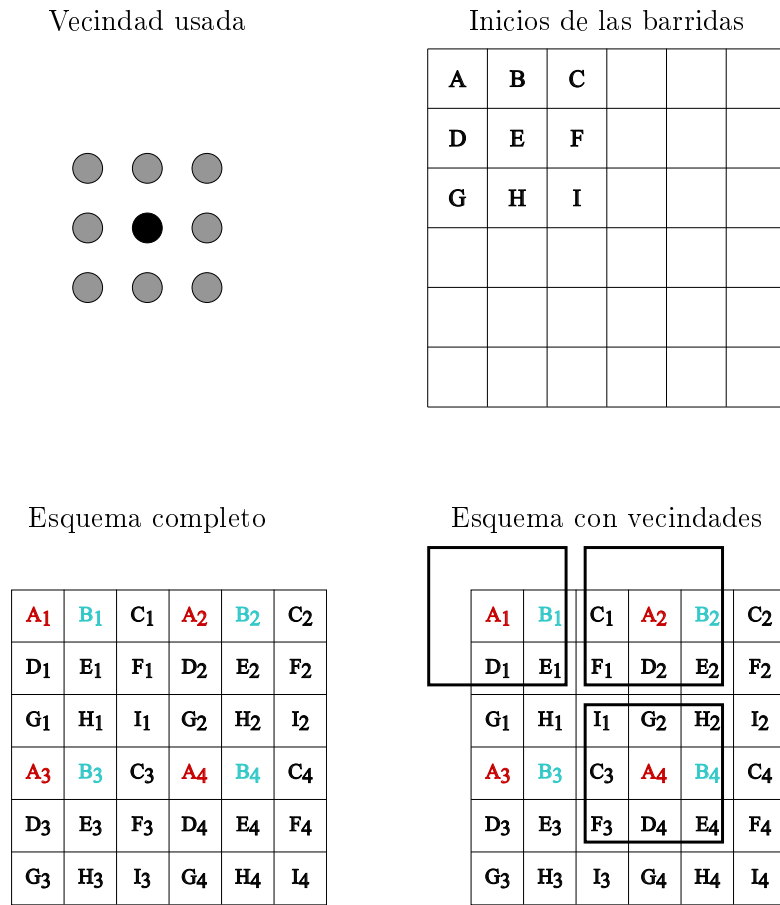


FIGURA 4.1: Un posible esquema de visitas para una vecindad 3×3 . En la primera barrida se actualizan los estados de los píxeles A_1, A_2, \dots . Después se visitan los píxeles B_1, B_2, \dots , etc. De esta manera se evita actualizar dos píxeles con vecinos en común sucesivamente.

Resumiendo la información de esta sección, se presentan los pasos principales de ICM en el algoritmo 3. En la siguiente sección se introduce un nuevo algoritmo que se basa en los mismos principios que ICM. Al final del capítulo se comparan los dos algoritmos.

4.3. Sequential Band Merging

En esta sección se presenta un nuevo algoritmo para la segmentación de imágenes basado en el modelo de Potts. El algoritmo denominado *Sequential Band Merging* (SBM) tiene varias características particulares que lo diferencian claramente del algoritmo ICM.

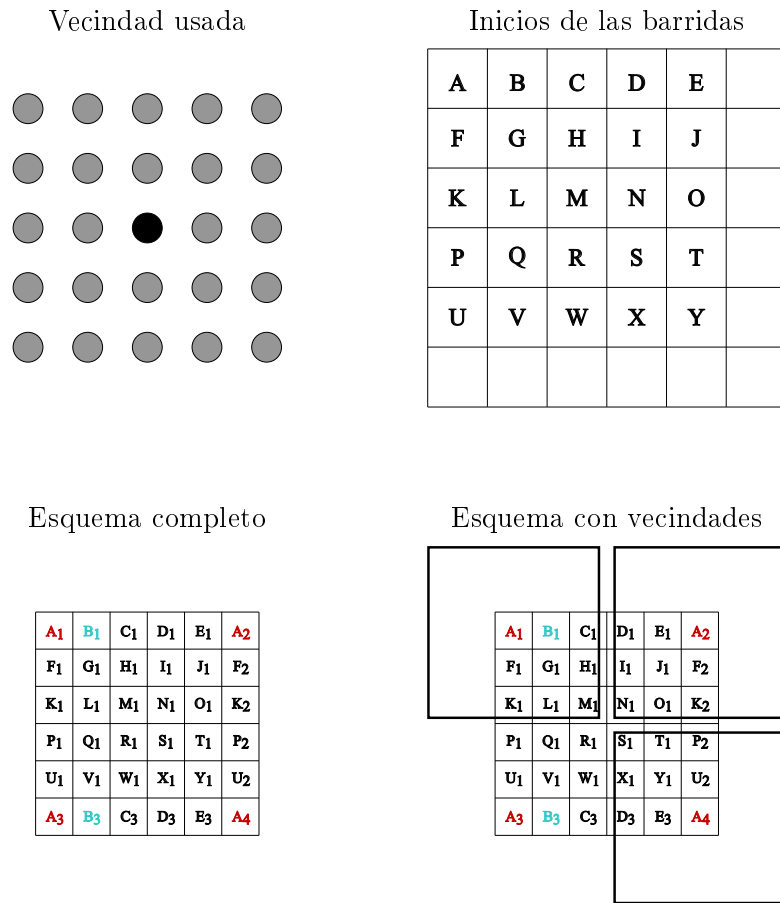


FIGURA 4.2: Un posible esquema de visitas para una vecindad 5×5 . En la primera barrida se actualizan los estados de los píxeles A_1, A_2, \dots . Después se visitan los píxeles B_1, B_2, \dots , etc. Note que para una vecindad 5×5 se necesitan más barridas que para vecindades chicas para cumplir con las condiciones de convergencia.

4.3.1. Motivación

Hasta aquí se presentaron dos modelos causales (PCVT y CEP) y una realización del modelo de Potts (ICM). A pesar de sus obvias diferencias, todos estos modelos tienen limitaciones en cuanto el uso de funciones de probabilidad como funciones de emisión. Por ejemplo, PCVT y CEP permiten la aplicación de distintas distribuciones de probabilidad, pero solo para imágenes de una banda (niveles de gris) mientras el uso de la distribución normal es obligatorio para ICM.

Entonces, surge un problema a la hora de trabajar con imágenes multiespectrales que no siguen la distribución normal. Un ejemplo son las imágenes de radar que usualmente

se modelan con distribuciones Gamma. En otras aplicaciones puede ser útil trabajar con densidades de kernel. O, más general, modelar cualquier clase en cualquier banda con cualquier distribución. Por ejemplo, la primera clase se podría modelar con una distribución Gamma en la primera banda, luego con una distribución Weibull en la segunda banda, etc. Esto sería el máximo de flexibilidad en cuanto funciones de probabilidad.

Conforme a nuestro conocimiento, no existe ningún algoritmo de segmentación en la literatura con tanta flexibilidad para las funciones de emisión. Entonces, el primer motivo para desarrollar el algoritmo SBM es la necesidad de un método que no tenga limitaciones para sus funciones de densidad.

El segundo desafío es diseñar un algoritmo que permita trabajar con vecindades de distintos tamaños en distintas bandas. Esta necesidad surge de una característica de imágenes de teledetección donde cada banda se observa con distintos niveles ruido. Generalmente las bandas de luz visible (con longitudes entre 380 nm y 750 nm) tiene una buena relación señal-ruido mientras otras bandas (sobre todo las bandas con longitudes mayor a 1000 nm) tiene incorporado mucho ruido.

En resumen, se busca desarrollar un algoritmo que se adapte perfectamente a las necesidades de segmentación de imágenes de teledetección. A parte de los de requerimientos mencionados, es deseable que la complejidad computacional del algoritmo no sea demasiado alta para poder segmentar imágenes satelitales con más de 1000×1000 píxeles.

4.3.2. Esquema general

El esquema general del algoritmo SBM consiste en tres pasos. Primero, transformar las observaciones (o sea la imagen) a un espacio característico (*feature space*). Segundo, estimar parámetros contextuales para cada banda y luego aplicar filtros no lineales en el espacio característico. Tercero, fundir la información de todas las bandas del espacio característico para obtener un mapa de estados (que se puede interpretar como una segmentación de la imagen). Para el tercer paso se desarrolla un novedoso método de fusión de datos.

4.3.3. Probabilidades marginales

En los últimos capítulos se denominó \mathbf{X} a las observaciones y $X_{i,j}$ hacía referencia al píxel (i, j) . Ahora se agrega la notación $X^{(k)}$ referida a los valores de la banda k de la imagen \mathbf{X} . Siguiendo esta notación, $X_{i,j}^{(k)}$ es el valor de gris del píxel (i, j) de la banda k donde $k \in 1, 2, \dots, K$.

Además, se supone que los valores de gris de las distintas bandas dependen solo del estado oculto de cada píxel y de la correspondiente función de emisión. Para formalizar esto, usamos la notación de la sección 3.6 donde se definió un conjunto $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ de familias de distribuciones con Θ_l siendo el espacio de parámetros de la familia \mathcal{F}_l . Si fijamos que el estado $\omega_{i,j}$ se modela en la banda k con una función de probabilidad de la familia \mathcal{F}_l obtenemos

$$P\left(X_{i,j}^{(k)}|\Theta, \boldsymbol{\omega}\right) = P\left(X_{i,j}^{(k)}|\Theta_l, \omega_{i,j}\right). \quad (4.4)$$

Note, que la ecuación (4.4) es una generalización de la suposición

$$P\left(X_{i,j}|\boldsymbol{\omega}\right) = P\left(X_{i,j}|\omega_{i,j}\right), \quad (4.5)$$

que usan PCVT, CEP y ICM.

Con la suposición de la ecuación (4.4) podemos aplicar el teorema de Bayes para calcular las probabilidades marginales de que el píxel (i, j) tenga estado $l \in 1, 2, \dots, L$ para los valores de gris de la banda k .

$$P\left(\omega_{i,j} = l|X_{i,j}^{(k)}\right) \propto P\left(X_{i,j}^{(k)}|\omega_{i,j} = l\right) P\left(\omega_{i,j} = l\right). \quad (4.6)$$

En la ecuación (4.6), el termino $P(\omega_{i,j} = l)$ es la probabilidad a priori de los estados. Esta probabilidad puede ser usada para ponderar las probabilidades de los estados antes de mirar las observaciones. En algunos casos, donde contamos con el conocimiento de expertos puede ser útil ponderar los estados, pero el enfoque de este trabajo se centra más en algoritmos universales. Por lo tanto, se trabaja con probabilidades a priori que son uniformes. Esto significa $P(\omega_{i,j} = l) = 1/L$. De esta manera no se perjudica ningún estado a priori.

Una vez calculado las marginales, es necesario normalizar las probabilidades de todos los píxeles en todas las bandas para convertir el símbolo “ \propto ” de la ecuación (4.6) en un “=”.

Por ende, se tiene que cumplir

$$\sum_{l=1}^L P\left(\omega_{i,j} = l | X_{i,j}^{(k)}\right) = 1 \quad (4.7)$$

$$\forall k \in 1, 2, \dots, K; \quad i \in 1, 2, \dots, u; \quad j \in 1, 2, \dots, v.$$

Analizando el cálculo de las probabilidades marginales, es destacable que no se hizo ninguna suposición acerca de las funciones de emisión. Lo único que se necesita en esta instancia son las probabilidades marginales que pueden ser calculadas para cualquier función de probabilidad. Por consiguiente, SBM tiene la flexibilidad de usar cualquier combinación de densidades en cada una de las K bandas.

Además, cabe destacar que en la imagen original cada píxel tuvo asociado un vector de K valores de gris. Entonces, calculando las marginales para los L estados se proyectan los datos originales a un hiperespacio de dimensión KL llamado espacio característico.

Por último, evaluamos el coste computacional de este primer paso. Las ecuaciones (4.6) y (4.7) son computacionalmente muy sencillas. Incluso para imágenes grandes, una computadora personal no necesita más que unos segundos para resolverlas.

Antes de seguir con el segundo paso de SBM se presenta un esquema completo del primer paso de SBM en la figura 4.3. En esta figura se usa un ejemplo concreto (imagen con tres bandas y dos clases) para explicar cómo transformar las observaciones al espacio característico. En las siguientes secciones se presenta un novedoso método que procesa los datos del espacio característico para llegar a una segmentación de la imagen original.

4.3.4. Aplicación del Bilateral Filter

En esta sección se presenta el segundo paso del algoritmo SBM que consiste en la aplicación de un filtro no paramétrico y no lineal. En concreto se trata del *Bilateral Filter* (BF) [92]. Pero, en vez de suavizar las observaciones, se corre el BF directamente en el espacio característico que está formado por las probabilidades marginales. De esta manera, se espera evitar el efecto de *blurring* de los valores de gris de la imagen original.

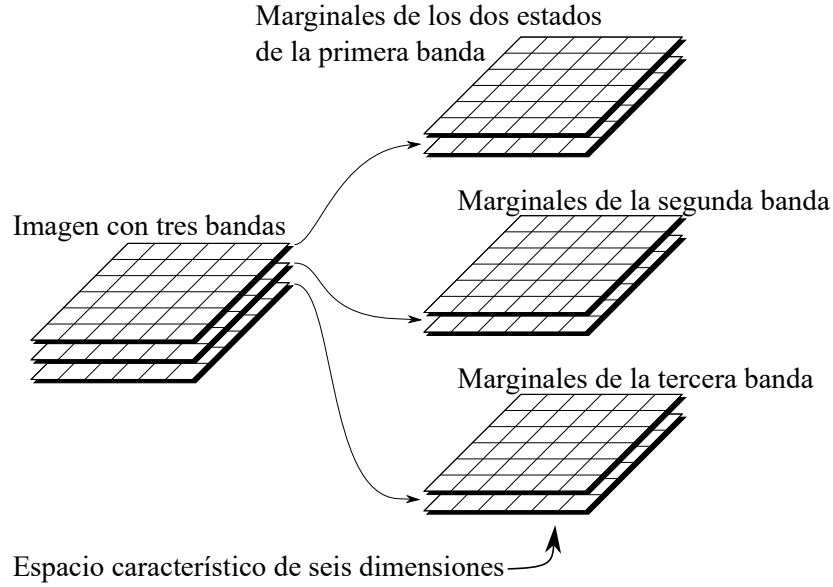


FIGURA 4.3: Esquema del primer paso del algoritmo SBM para $K = 3$ bandas y $L = 2$ estados. Las bandas de la imagen original se proyectan en un espacio de dimensión KL .

Al mismo tiempo, se aprovechan dos características del BF que son: promediado espacialmente sin suavizar bordes [93], o, en nuestros términos: suavización de las marginales sin blurring.

En lo siguiente, llamaremos $\mathbf{q}_{i,j}^{(k)}$ a las marginales del píxel (i, j) en la banda k según la definición en la ecuación (4.6):

$$\mathbf{q}_{i,j}^{(k)}(l) = P\left(\omega_{i,j} = l | X_{i,j}^{(k)}\right) \quad l = 1, 2, \dots, L. \quad (4.8)$$

Aplicando el algoritmo BF a los valores $\mathbf{q}_{i,j}^{(k)}$ se obtienen las marginales suavizadas $\mathbf{q}_{i,j}^{(k)*}$. Para esto, se utiliza la formula estándar del BF que es

$$\mathbf{q}_{i,j}^{(k)*} = \sum_{i',j' \in C} K_{i,j,i',j'}^{(k)} \mathbf{q}_{i',j'}^{(k)}. \quad (4.9)$$

En la ecuación (4.9), C representa la vecindad (a veces llamado *clique*) de la banda k . Además $K_{i,j,i',j'}^{(k)}$ es el *kernel* de los píxeles (i, j) e (i', j') .

La vecindad C consiste típicamente de todos los píxeles dentro de un cierto radio. Sin embargo, se estudian distintas vecindades en este trabajo, incluso vecindades cuadradas

o vecindades arbitrarias que son definidas por un usuario. Par mayor información acerca de vecindades refiérase al apéndice B o a [94].

Como función de kernel se elige el clásico kernel gaussiano que viene dado por

$$K_{i,j,i',j'}^{(k)} = \exp \left(-\frac{\|(i,j)-(i',j')\|^2}{h_x^2} - \frac{\|X_{i,j}^{(k)} - X_{i',j'}^{(k)}\|^2}{h_y^2} \right). \quad (4.10)$$

Note en la ecuación (4.10), que el kernel $K_{i,j,i',j'}^{(k)}$ depende por un lado de la distancia euclidiana de los píxeles (i,j) e (i',j') y por otro lado de los valores de gris de los dos píxeles $(X_{i,j}^{(k)}$ y $X_{i',j'}^{(k)})$. Ambos componentes – la distancia euclidiana y la diferencia de los valores de gris – son ponderados por los parámetros de kernel h_x y h_y . Si h_x es chico, el BF considera solo píxeles muy cerca del píxel (i,j) , mientras para $h_x \rightarrow \infty$ se asigna el mismo peso a todos los píxeles que se encuentran dentro de la vecindad C . Lo mismo vale para los valores de gris. Mientras más chico h_y , más sensitivo el kernel con respecto a diferencias entre los valores de gris.

Antes de seguir con la descripción del BF es necesario normalizar $K_{i,j,i',j'}^{(k)}$ para que valga

$$\begin{aligned} \sum_{i',j' \in C} K_{i,j,i',j'}^{(k)} &= 1 \\ \forall k \in 1, 2, \dots, K; \quad i \in 1, 2, \dots, u; \quad j \in 1, 2, \dots, v. \end{aligned} \quad (4.11)$$

Con la introducción del kernel gaussiano en la ecuación (4.10) aparecieron tres parámetros. Primero la vecindad C que es fundamental para todas las realizaciones del modelo de Potts. Por lo tanto, tomamos C como un parámetro que fija el usuario (igual que en el caso ICM). Los otros dos parámetros (h_x y h_y) son propios de BF. Para evitar que una mala elección de estos dos parámetros empeore el resultado final de SBM proponemos dos métodos novedosos para estimar sus valores.

Empecemos con el parámetro h_x . El objetivo es poner mucho peso en los píxeles vecinos que están cerca del píxel (i,j) sin dejar de lado los píxeles vecinos más lejanos. Para fijar un buen compromiso entre los pesos de los píxeles cercanos y lejanos se propone calcular una variable auxiliar d_c . Esta variable es la distancia euclidiana máxima que existe dentro de la vecindad C . Entonces, se propone el siguiente estimador para el parámetro h_x :

$$h_x = \frac{\sqrt{2}}{3} d_c. \quad (4.12)$$

El estimador de la ecuación (4.12) tiene una interpretación geométrica relacionada con el kernel gaussiano. Recuerde que la parte espacial del kernel viene dada por

$$K_{i,j,i',j'}^{espacial} = \exp\left(\frac{-\|(i,j) - (i',j')\|^2}{h_x^2}\right). \quad (4.13)$$

Note que, aparte de unas constantes, la ecuación (4.13) describe una función normal en dos dimensiones. Y el punto de reflexión de esta función se encuentra exactamente a un tercio de d_c si estimamos h_x de acuerdo a la ecuación (4.12). Ahora es fácil calcular el peso mínimo que asigna el kernel a los píxeles en el borde de C .

$$\begin{aligned} K_{i,j,i',j'}^{espacial_{min}} &= \exp\left(\frac{-\|(i,j) - (i',j')\|^2}{\left(\frac{\sqrt{2}}{3}d_c\right)^2}\right) \\ &= \exp\left(\frac{-d_c^2}{\left(\frac{\sqrt{2}}{3}d_c\right)^2}\right) \\ &= \exp\left(-\frac{9}{2}\right) \approx 0,0111. \end{aligned} \quad (4.14)$$

Lo que dice la ecuación (4.14) es que el valor del kernel espacial para los píxeles en el borde de C es un poco más que 0,01. Este valor parece ser razonable para los píxeles lejanos, sobre todo si recordamos que el valor máximo del kernel espacial es 1 para $\|(i,j) - (i',j')\| = 0$.

El segundo parámetro del kernel gaussiano es h_y . Este valor pondera la distancia fotométrica entre dos píxeles. Aquí, el objetivo principal es elegir un valor para h_y tal que dos observaciones que pertenecen al mismo estado tengan un kernel alto, mientras que dos observaciones de distintos estados tengan un kernel chico. Por lo tanto, es conveniente estimar h_y en función del mapa de estados actual. Además, hay que tener en cuenta que la distribución de los valores de gris puede variar mucho entre las distintas bandas. Por consiguiente, se introduce el término $h_y^{(k)}$ que representa el kernel fotométrico de la banda k .

Como mencionamos en el párrafo anterior, los parámetros $h_y^{(k)}$ están fuertemente relacionados con el mapa de estados. Incluso se puede desarrollar teóricamente un método óptimo para estimar los $h_y^{(k)}$. Este método consiste en analizar la vecindad de cada píxel junto con los correspondientes valores de gris. Con esta información se puede calcular el valor óptimo de $h_y^{(k)}$ maximizando el kernel fotométrico esperado para dos píxeles vecinos

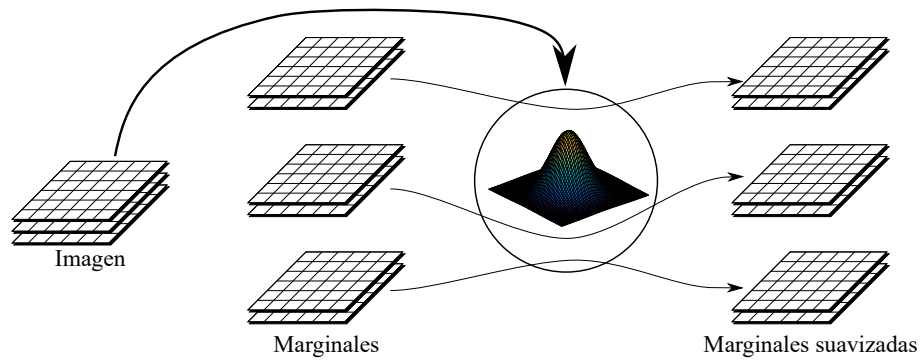


FIGURA 4.4: Esquema del segundo paso del algoritmo SBM para $K = 3$ bandas y $L = 2$ estados. Las bandas de la imagen original determinan el grado de suavización de las probabilidades marginales.

con el mismo estado. Lamentablemente este procedimiento es muy costoso y poco práctico. Por lo tanto, se propone un método aproximativo que estima un valor $h_y^{(k)}(l)$ para cada estado $l \in 1, 2, \dots, L$ y luego promedia los valores auxiliares $h_y^{(k)}(l)$ para obtener $h_y^{(k)}$.

La idea principal del estimador aproximativo es maximizar el kernel esperado para dos píxeles arbitrarios que contienen el mismo estado en vez de considerar la vecindad de cada píxel por separado. Esto es equivalente a calcular la máxima verosimilitud de $h_y^{(k)}(l)$ para cada estado l . La resultante fórmula para el estado l en la banda k viene dada por

$$h_y^{(k)}(l) = \sqrt{\frac{2 \sum_{\omega_{i,j}=l} (X_{i,j}^{(k)} - \mu(l))^2}{\sum_{\omega_{i,j}=l} 1}}. \quad (4.15)$$

Luego, se obtiene $h_y^{(k)}$ como el promedio de los valores $h_y^{(k)}(l)$.

$$h_y^{(k)} = \frac{\sum_{l=1}^L h_y^{(k)}(l)}{L}.$$

En la figura 4.4 se presenta un esquema del segundo paso de SBM. Cabe destacar que las observaciones originales definen el grado de suavización, aunque no participan activamente en el proceso de filtrado.

Por último, se introduce una nueva variable que surge de las ecuaciones de esta sección y que facilita la descripción del tercer paso de SBM en la siguiente sección. Recuerde que, $\mathbf{q}_{i,j}^{(k)*}$ son las probabilidades marginales suavizadas de la banda k que se calculan

en la ecuación (4.9). En este contexto, denominamos $\mathbf{Q}_{i,j}^*$ al conjunto de probabilidades marginales suavizadas de todas las bandas, con $\mathbf{Q}_{i,j}^* \in \mathcal{R}^{KL}$.

$$\mathbf{Q}_{i,j}^* = \left[\mathbf{q}_{i,j}^{(1)*}, \mathbf{q}_{i,j}^{(2)*}, \dots, \mathbf{q}_{i,j}^{(K)*} \right]. \quad (4.16)$$

La siguiente sección se refiere a los valores $\mathbf{Q}_{i,j}^*$ de la ecuación (4.16) como vectores característicos. Esto se debe a que el vector $\mathbf{Q}_{i,j}^*$ contiene toda la información necesaria para segmentar o clasificar el píxel (i, j) . Además se llama $\mathbf{Q}_{i,j}^{(l)*}$ a las L entradas correspondientes a la banda l .

4.3.5. Clasificación

El tercer paso de SBM consiste en asignar un estado $l \in 1, 2, \dots, L$ a cada píxel. Para esto se propone un algoritmo novedoso que trabaja con L vectores de base $b_1, b_2, \dots, b_L \in \mathcal{R}^{KL}$. Aparte, b_l^k hace referencia a los valores de b_l que corresponden a la banda k .

En grandes rasgos, el algoritmo determina los estados de los píxeles de acuerdo a la proximidad euclidiana de los vectores característicos $\mathbf{Q}_{i,j}^*$ a los vectores de base. O sea, se asigna el estado l al píxel (i, j) si la distancia entre $\mathbf{Q}_{i,j}^*$ y b_l es mínima.

El problema que aparece en este contexto es que no se conocen los vectores de base. Además, no existe información a priori acerca de la homogeneidad de los estados ni de la cantidad de píxeles que tienen estado l . Para superar estos obstáculos se desarrolla un esquema iterativo capaz de hallar los vectores de base y clasificar los píxeles de la imagen.

El propuesto proceso iterativo aprovecha una particularidad del vector característico $\mathbf{Q}_{i,j}^*$. Recuerde que, $\mathbf{Q}_{i,j}^*$ está compuesto por las probabilidades marginales de K bandas, donde las primeras L entradas vienen de la banda uno, los siguientes L entradas son de la banda dos y así sucesivamente. Por lo tanto, se puede tratar cada banda por separado, empezando con la primera banda.

Dado una segmentación inicial ω , se pueden fijar los primeros L valores de los vectores de base como el promedio de las correspondientes probabilidades marginales.

$$b_l^{(1)} = \frac{\sum_{\omega_{i,j}=1} \mathbf{Q}_{i,j}^{(1)*}}{\sum_{\omega_{i,j}=1} 1} \quad \forall l \in 1 \dots L. \quad (4.17)$$

Después de aplicar la ecuación (4.17), los vectores de base tienen la forma

$$\begin{aligned}
 b_1 &= [\mathbf{v}_{111}, \mathbf{v}_{112}, \dots, \mathbf{v}_{11L}, \text{nan}, \text{nan}, \dots, \text{nan}]^T \\
 b_2 &= [\mathbf{v}_{211}, \mathbf{v}_{212}, \dots, \mathbf{v}_{21L}, \text{nan}, \text{nan}, \dots, \text{nan}]^T \\
 \vdots &= [\vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \vdots]^T \\
 b_L &= [\mathbf{v}_{L11}, \mathbf{v}_{L12}, \dots, \mathbf{v}_{L1L}, \text{nan}, \text{nan}, \dots, \text{nan}]^T
 \end{aligned} \tag{4.18}$$

En la ecuación (4.18), *nan* significa “not a number”. O sea, estas entradas de los vectores b_1, b_2, \dots, b_L no tienen asignado un valor hasta ahora.

El siguiente paso consiste en calcular las distancias euclidianas de los vectores característicos a los vectores de base que surgen de la ecuación (4.17). Luego, se asigna un estado a cada píxel de acuerdo a

$$\omega = \left[\arg \min_{l \in \{1, 2, \dots, L\}} \left(\mathbf{Q}_{i,j}^{(1)*} - b_l^{(1)} \right) \right]_{i,j} . \tag{4.19}$$

Una vez evaluada la ecuación (4.19), se puede agregar la segunda banda al proceso de segmentación. Para esto se introduce la notación $b_l^{(1:k)}$ que representa los valores de b_l correspondientes a las bandas $1, 2, \dots, k$. Siguiendo esta lógica, llamamos $\mathbf{Q}_{i,j}^{(1:k)*}$ a los correspondientes valores de $\mathbf{Q}_{i,j}^*$ de las bandas $1, 2, \dots, k$. Ahora podemos extender la ecuación (4.17) a

$$b_l^{(1:2)} = \frac{\sum_{\omega_{i,j}=1} \mathbf{Q}_{i,j}^{(1:2)*}}{\sum_{\omega_{i,j}=l} 1} \quad \forall l \in \{1 \dots L\}. \tag{4.20}$$

Note en la ecuación (4.20) que se obtienen los valores $b_l^{(2)}$ como promedios de $\mathbf{Q}_{i,j}^{(2)*}$ usando un mapa de estados ω que viene de la primera banda de la ecuación (4.19). Entonces, se conectaron las primeras dos bandas a pesar de que se consideraban independientes en las últimas dos secciones.

Para determinar el nuevo mapa de estados, se adapta la ecuación (4.19) a dos bandas y se obtiene

$$\omega = \left[\arg \min_{l \in \{1, 2, \dots, L\}} \left(\mathbf{Q}_{i,j}^{(1:2)*} - b_l^{(1:2)} \right) \right]_{i,j} . \tag{4.21}$$

Recuerde que ahora los vectores de base de la ecuación (4.18) quedan como

$$\begin{aligned}
 b_1 &= [v_{111}, v_{112}, \dots, v_{11L}, \mathbf{v}_{121}, \mathbf{v}_{122}, \dots, \mathbf{v}_{12L}, nan, \dots]^T \\
 b_2 &= [v_{211}, v_{212}, \dots, v_{21L}, \mathbf{v}_{221}, \mathbf{v}_{222}, \dots, \mathbf{v}_{22L}, nan, \dots]^T \\
 \vdots &= [\vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \vdots, \vdots]^T \\
 b_L &= [v_{L11}, v_{L12}, \dots, v_{L1L}, \mathbf{v}_{L21}, \mathbf{v}_{L22}, \dots, \mathbf{v}_{L2L}, nan, \dots]^T
 \end{aligned} \tag{4.22}$$

Lo que visualiza la ecuación (4.22) es que ya se llenaron las entradas $b_l^{(2)}$ de los vectores de base. La idea ahora es seguir con este procedimiento para completar los valores faltantes que son $b_l^{(3:K)}$.

Generalizando las ecuaciones (4.20) y (4.21) se pueden derivar dos fórmulas que permiten la incorporación sucesiva de las bandas 3 a K de la imagen. Como se mencionó anteriormente se calculan primero los vectores de base de la banda 1 hasta la banda k

$$b_l^{(1:k)} = \frac{\sum_{\omega_{i,j}=1} \mathbf{Q}_{i,j}^{(1:k)*}}{\sum_{\omega_{i,j}=l} 1} \quad \forall l \in 1 \dots L, \tag{4.23}$$

y luego se determina el mapa de estados de acuerdo a

$$\omega = \left[\arg \min_{l \in 1,2,\dots,L} \left(\mathbf{Q}_{i,j}^{(1:k)*} - b_l^{(1:k)} \right) \right]_{i,j}. \tag{4.24}$$

Aplicando las ecuaciones (4.23) y (4.24) sucesivamente hasta la banda K , se obtienen vectores de base completos (sin entradas “nan”) y un mapa de estados ω que corresponde a todas las bandas de la imagen.

Una vez terminada la incorporación de todas las bandas, hay que comprobar si el mapa de estados ha convergido en el último paso (de la banda $(K-1)$ a la banda K). Si esto no ocurre, hay que empezar de nuevo con la ecuación (4.17). La única diferencia con respecto a la primera iteración, es que ahora se usa el mapa de estado que surgió de la ecuación (4.24) como mapa inicial. Este procedimiento se repite hasta que el mapa de estados converja.

Después de hallar el mapa de estados, se termina la primera iteración de SBM. Entonces, hay que empezar nuevamente con el primer paso de SBM, pero esta vez se usa como mapa inicial el mapa que salió de la ecuación (4.24).

En resumen, hemos usado todas las bandas de la imagen para hallar un consistente mapa de estados. Además, obtuvimos los correspondientes vectores de base $b_1, b_2, \dots, b_L \in \mathcal{R}^{KL}$. Cabe destacar que en todo este proceso no se ha recurrido a funciones de probabilidades multidimensionales para conectar los valores de gris de distintas bandas.

El coste computacional de este paso de SBM depende obviamente de la cantidad de iteraciones necesarias. Sin embargo, el coste por iteración es relativamente chico porque solo hay que calcular LK veces un promedio (que tiene un coste casi igual a cero) y $LKuv$ veces una distancia (que es computacionalmente muy simple). Entonces, para una imagen de tamaño $(u \times v)$, el orden del método propuesto es de $\mathcal{O}(L^2K^2uv)$. Recuerde en este contexto que la cantidad de estados L y de bandas K es generalmente menor a diez. Por lo tanto, los términos L^2 y K^2 no juegan un rol importante.

Generalmente el algoritmo SBM converge dentro de pocas iteraciones. Solo en algunos casos aislados se necesitan más de 50 iteraciones para determinar el mapa de estados. Además, los resultados de SBM son muy prometedoras como se muestra en el capítulo 7 donde se presentan los resultados experimentales. Para una descripción esquemática del SBM refiérase al algoritmo 4 y a la figura 4.5.

Algoritmo 4: Sequential Band Merging (SBM)

1. Inicializar mapa de estados
 2. Ajustar los parámetros Θ de las funciones de probabilidad
 3. Calcular probabilidades marginales de cada píxel usando la ecuación (4.6)
 4. Aplicar el filtro bilateral según la ecuación (4.9)
 5. Hallar un mapa de estados usando sucesivamente las ecuaciones (4.23) y (4.24)
 6. Si el mapa de estados no convergió volver a empezar con el paso 2)
-

En la siguiente sección se comparan los dos modelos basados en el modelo de Potts – ICM y SBM – con respecto a sus características formales. Después se presenta una extensión de SBM en el capítulo 5. Dicha extensión permite clasificar datos que no se encuentran en una grilla equidistante.

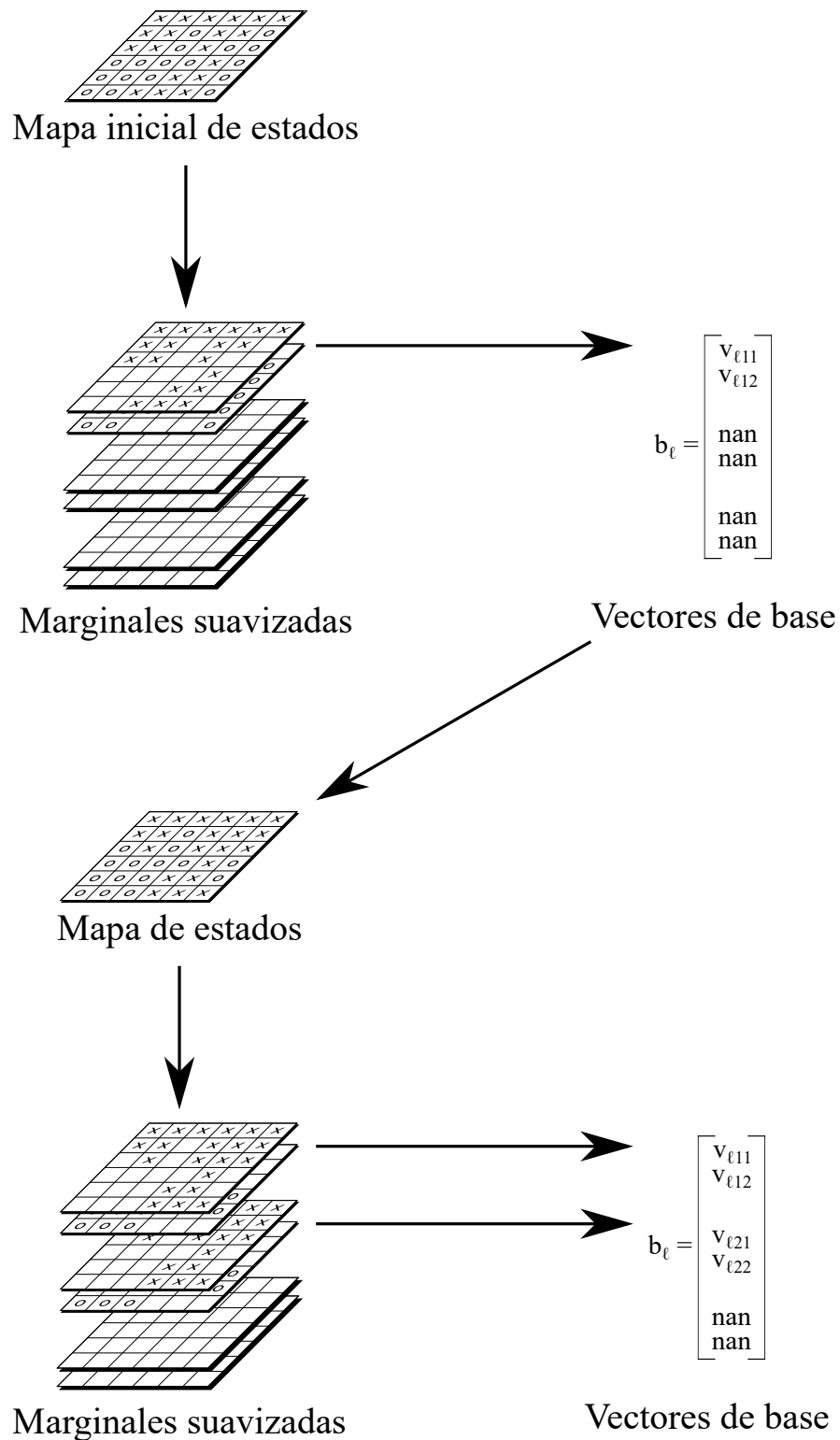


FIGURA 4.5: Esquema del tercer paso del algoritmo SBM para $K = 3$ bandas y $L = 2$ estados. Se muestra el procedimiento de la primera iteración que empieza con un mapa inicial y termina con un nuevo mapa de estados. Este mapa es el punto inicial para la segunda iteración.

4.4. Comparación de algoritmos

En este capítulo se presentaron dos realizaciones del modelo de Potts. En esta sección se compara los fundamentos teóricos de los dos algoritmos. Luego, en el capítulo 7, se evalúa el desempeño de los dos modelos para datos experimentales.

Ambos modelos usan una vecindad no causal que generalmente es elegida por el usuario. Sin embargo, hay grandes diferencias dependiendo de cómo se aplique dicha vecindad.

Por un lado, ICM considera solo el actual estado oculto de los vecinos sin importar los valores de gris ni la evidencia a favor de este estado. Por ejemplo, podría pasar que un píxel vecino está en el estado 1 con 50,1% de probabilidad mientras el estado 2 tiene una probabilidad de 49,9%. Y sin embargo ICM descartaría completamente el estado 2 a la hora de calcular las probabilidades condicionales. Además, ICM no distingue entre vecinos cercanos y lejanos. Todos los píxeles de la vecindad tienen el mismo peso con independencia de su ubicación. En otras palabras, ICM usa $h_x \rightarrow \infty$ en la ecuación (4.10).

Por el otro lado, SBM asigna un peso individual a cada píxel en la vecindad. Este peso se compone por la distancia entre dos píxeles y por la similitud de los valores de gris. Si bien la forma de SBM de asignar un peso individual a cada píxel vecino parece más exacto, hay que analizar la situación detalladamente.

La forma sencilla de ICM de mirar solo los estados ocultos tiene dos ventajas con respecto al método de SBM. Por un lado, el coste computacional es menor porque no hay que calcular pesos individuales para cada píxel. Por el otro, hay estudios que demuestran que en imágenes con mucho ruido a veces es ventajoso usar solo los estados de los píxeles sin incorporar más información [90]. Una explicación de esto es que, si las observaciones son defectuosas, entonces las conclusiones que se pueden sacar también tienen una tendencia a ser incorrectas. Por lo tanto, se refuerzan los efectos de ruido en esquemas con pesos individuales como implementa SBM. Sin embargo, cabe destacar que en la mayoría de los casos la forma avanzada de SBM lleva a mejores resultados – aunque a un coste computacional más alto.

Otra gran diferencia entre ICM y SBM es la forma como y en qué momento se tiene en cuenta información contextual. En el caso de ICM, se realizan las tareas de incorporar información contextual y clasificar en un solo paso al ejecutar la ecuación (4.3). Para

entender esto hay que ver que dicha ecuación reúna las probabilidades de observación (información no contextual) con las probabilidades condicionales (información contextual) para determinar el estado de cada píxel (clasificación).

En contrario al ICM, el algoritmo SBM necesita un paso para incorporar información contextual (mediante el BF) y luego, otro paso para realizar la clasificación de cada píxel. Vale mencionar que en el algoritmo SBM se implementa el BF de tal forma que los valores de gris de la imagen sirvan solo para fijar los pesos de cada vecino. Luego el filtro trabaja en el espacio característico. Por lo tanto, no es necesario fijar un esquema de visita en el caso de SBM.

Mientras SBM no requiere de un esquema de visita, ICM solo converge si se aplica un esquema de visita que cumpla con ciertas condiciones (ver sección 4.2). Si bien el esquema de visita es una tarea extra a la hora de implementar ICM, al menos nos da una garantía de que el algoritmo converge. En cambio, no existe tal garantía de convergencia en el caso de SBM. Por esto es necesario en algunos pocos experimentos cortar el SBM porque alcanzó el máximo de iteraciones permitidas que en este trabajo se fija en 200.

En cuanto al coste computacional es fácil comparar los dos algoritmos. ICM requiere menos recursos que SBM. Sin embargo, la complejidad computacional de SBM es notablemente menor que la de los modelos causales PCVT y CEP que se presentaron en el capítulo 3. Además, hay que constatar que el coste de ICM como de SBM crece linealmente con la cantidad de píxeles de la imagen y con la cantidad de vecinos. Por lo tanto, ambos algoritmos tienen sus limitaciones para segmentar imágenes satelitales enteras, ya que una imagen típica – como una Landsat 8 – tiene $7751 \times 7831 = 60,698,081$ píxeles.

Por último, se analizan los métodos de clasificación de los algoritmos ICM y SBM. Como se mencionó anteriormente, ICM usa un procedimiento sencillo para determinar el estado de un píxel (ver ecuación (4.3)). En cambio, SBM aplica un algoritmo más avanzado que permite fusionar los datos de distintas bandas, como se muestra en la figura 4.5. Para este procedimiento no influye si los puntos de observación son píxeles equidistantes u observaciones en sitios arbitrarios como mediciones de humedad de suelo o fotosíntesis.

Entonces, la idea para el siguiente capítulo consiste en la generalización del algoritmo SBM de tal forma que pueda procesar datos de sitios arbitrarios. En este contexto, el

objetivo es desarrollar un algoritmo con suficiente flexibilidad como para unir datos de imágenes satelitales y datos de mediciones *in situ* para obtener un único mapa de estados.

Capítulo 5

Fusión de datos

En este capítulo se presenta un algoritmo capaz de procesar y fusionar datos de distintos sensores. El algoritmo denominado *Markov Data Fusion* (MDF) se puede entender como una extensión del algoritmo SBM del capítulo 4. Pero antes de presentar los fundamentos teóricos de MDF y de posibles algoritmos competidores, es necesario introducir el tema de fusión de datos, destacar los desafíos más urgentes y diferenciar las distintas líneas de investigación.

5.1. Introducción

El termino «fusión de datos» abarca muchos algoritmos distintos y técnicas desarrolladas en diferentes áreas. Algunos ejemplos son redes de sensores, sistemas de defensa, robótica, sistemas inteligentes y procesamiento de imágenes [95].

Por un lado, el concepto principal de fusión de datos es el mismo en todas las áreas de investigación: Procesar datos de distintos sensores para mejorar el desempeño de un sistema. Por otro lado, hay grandes diferencias entre las diversas aplicaciones en cuanto tipo de datos, métodos de procesamiento y objetivos. Antes de estudiar estas cuestiones en profundidad, presentamos la definición clásica de fusión de datos del *JDL Data Fusion Subgroup*.

Definición 5.1. Data Fusion. A process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined

position and identity estimates, and complete and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results [96].

Note que, la definición 5.1 es muy general y no especifica la tarea principal (asociación, correlación y combinación de datos e información) ni el objetivo final (lograr mejores resultados). Este enfoque universal refleja la forma como se trabaja con algoritmos de fusión de datos. Por ejemplo, el *Data Fusion Contest* organizado por la *Geoscience and Remote Sensing Society* (IEEE) [97] dispone de datos de múltiples sensores, pero no especifica ningún objetivo concreto. Lo único que pide el jurado para ganar la competencia es desarrollar y describir un algoritmo novedoso que saque provecho de los datos.

Entonces, para aclarar el panorama, es necesario introducir criterios de clasificación que ayuden a distinguir los algoritmos de fusión de datos. Por ejemplo, Dasarathy [98] propuso cinco categorías de algoritmos que se diferencian por sus tipos de datos de entrada y salida.

1. *data in – data out* (DAI-DAO)
2. *data in – feature out* (DAI-FEO)
3. *feature in – feature out* (FEI-FEO)
4. *feature in – decision out* (FEI-DEO)
5. *decision in – decision out* (DEI-DEO)

La primera categoría (DAI-DAO), corresponde a algoritmos donde se busca refinar o completar un conjunto de datos crudos. Los métodos FEI-FEO tienen el mismo objetivo (refinar y completar) pero trabajan a nivel de características (*features*) en vez de datos crudos. En cambio, las categorías FEI-DEO y DEI-DEO se implementan típicamente en sistemas de soporte de decisiones. Por consiguiente, los algoritmos de este capítulo corresponden claramente a la segunda categoría ya que se usan datos provenientes de distintos sensores (*data in – DAI*) y el objetivo es encontrar patrones, o, en otras palabras, características (*features – FEO*).

En vez de usar los tipos de datos de entrada y salida para la clasificación de algoritmos, Castanedo [99] sugirió considerar las arquitecturas subyacentes de los algoritmos. Según él, existen cuatro arquitecturas.

1. *Arquitectura centralizada*. Hay un procesador central que recibe y procesa toda la información.
2. *Arquitectura descentralizada*. Hay una red de nodos donde cada nodo procesa una parte de la información. Generalmente los nodos se comunican entre ellos a través de medidas de información de Fisher o de Shannon.
3. *Arquitectura distribuida*. Hay nodos independientes que procesan los datos correspondientes antes de mandar los resultados a un procesador central. Luego, el procesador central realiza la fusión de datos.
4. *Arquitectura jerárquica*. Hay una jerarquía en el proceso de fusión de datos. Por ejemplo, se procesan los datos de dos nodos y luego se agrega otro nodo al resultado.

Para ubicar el algoritmo MDF en una de las cuatro categorías, es necesario adelantar su esquema principal. Recuerde que el MDF es una extensión del algoritmo SBM del capítulo 4, donde se calculaban las probabilidades marginales en un primer paso. En el caso del MDF también se pre-procesan los datos de los distintos sensores de la misma manera. Luego, se realiza la fusión de datos con un procesador central. Por lo tanto, el algoritmo MDF tiene una arquitectura distribuida.

En la siguiente sección se presenta el estado de arte de fusión de datos. Además, se explica porque las técnicas de fusión de datos son muy importantes para la agricultura de precisión.

5.2. Estado del arte y desafíos

Como se mencionó en la sección anterior, fusión de datos es un campo de investigación muy amplio con relativamente poca literatura que cubre el espectro completo [100, 101]. Por lo tanto, es difícil hablar de un estado de arte general. Sin embargo, se pueden identificar áreas donde la implementación de algoritmos de fusión de datos está más

avanzada que en otras. Por ejemplo, en aplicaciones relacionadas con el rastreo de objetos hay una muy importante línea de investigación que utiliza filtros de Kalman [102, 103].

En cambio, en la fusión de distintas imágenes satelitales se opta generalmente por modelos que procesan los datos de cada píxel individualmente [104] o en bloques [105]. Una posible aplicación para este tipo de fusión de datos es la reunión de imágenes ópticas e imágenes de radar [106]. Otro desafío es la incorporación de una banda pancromática a una imagen multispectral [107].

Para fusionar datos de sensores locales como sensores de humedad de suelo se usan generalmente modelos estadísticos [108] o modelos de correlación [109]. Además, hay muchos investigadores que buscan extender los modelos espaciales a modelos espacio-temporales [110, 111].

Aparte de trabajar en áreas bien marcadas como se describieron en los últimos párrafos (rastreo, imágenes y suelo) hay una tendencia importante de fusionar datos de distintas áreas. La idea subyacente es que, al usar datos de diferentes sensores, se pueden compensar fallos o inexactitudes de un sensor particular. Por ejemplo, hay una línea de investigación que busca medir el latido del corazón con mayor exactitud al fusionar datos de distintos sensores [112]. Sin embargo, la agricultura de precisión es el campo donde la fusión de datos multisensoriales tiene el mayor impacto. En este contexto, el objetivo principal es aprovechar la cobertura de imágenes satelitales y luego refinar los resultados con mediciones locales *in situ*.

En los últimos años, se ha generado una gran cantidad de publicaciones como el importante trabajo de Grunwald et al. [113]. En este trabajo se comparan distintos modelos que encuentran patrones de suelo. En total se estudian tres tipos de modelos. Primero, modelos de datos locales (por ejemplo, humedad de suelo), segundo, modelos que procesan imágenes satelitales y tercero, modelos donde se procesan todos los datos disponibles mediante técnicas de fusión de datos. Una de las conclusiones es que la fusión de datos (datos locales e imágenes satelitales) muestran un mejor desempeño al encontrar patrones de suelo. Por otro lado, los autores destacan que es necesario extender los existentes modelos de fusión de datos para mejorar su desempeño (por ejemplo, mejorar la detección de propiedades biogeoquímicas).

Otra publicación destacable es la de Casa et al. [114], donde se subrayan los efectos sinérgicos que pueden tener modelos de fusión de datos. Los modelos usados en dicho trabajo se basan en las técnicas de kriging y la regresión de mínimos cuadrados parciales. Un resultado importante es que es preferible usar imágenes multispectrales junto con observaciones *in situ* en vez de usar solo imágenes hiperespectrales.

Partiendo de esta base teórica, el objetivo de este capítulo es desarrollar un algoritmo de fusión de datos capaz de procesar imágenes satelitales y datos locales para descubrir patrones en un campo agrícola. Cabe destacar que el desafío de este trabajo es meramente teórico. Para aprovechar los resultados obtenidos es necesario un ingeniero agrónomo que interprete el mapa de patrones.

En resumen, la fusión de datos es una línea de investigación activa e importante para la agricultura de precisión. En la siguiente sección, se detallan los datos usados en este trabajo. Luego, se presentan distintos métodos de fusión de datos.

5.3. Sensores, datos y objetivos

Como se mencionó en la sección anterior, hay muchos efectos sinérgicos que surgen del uso de datos multisensoriales. La idea detrás de estos efectos sinérgicos es que las ventajas de un sensor compensan las desventajas de otro y viceversa. Entonces, se propone el uso de cinco tipos de sensores con características muy diferentes: imagen óptica, imagen de radar, humedad de suelo, fotosíntesis, potencial de tallo. En el capítulo 6 se introducen estos sensores con más detalle, pero en este capítulo se hace hincapié en las ventajas y desventajas de cada sensor. Por lo tanto, se presenta un resumen de los sensores en la tabla 5.1.

Aparecen varios problemas a la hora de querer fusionar los datos de los sensores de la tabla 5.1. Primero, hay que tener en cuenta que un píxel de una imagen satelital representa el promedio de un área cuadrada. Por ejemplo, una imagen Landsat 8 tiene una resolución de 30×30 metros. O sea, cada píxel es el valor medio de una superficie de 900 m^2 , mientras cada medición *in situ* es una observación puntual. Entonces, la primera pregunta es: ¿Cómo tratar los distintos tipos de datos? La respuesta más sencilla es considerar el valor de un píxel como una medición puntual en el centro de la superficie cubierta.

CUADRO 5.1: Comparación entre distintas fuentes de información de agricultura de precisión.

	Qué se mide?	Resolución	desventajas	ventajas
Imagen óptico	reflexión de luz (pasivo)	equidistante (~ 5 a 30 m)	nubes resolución	multiespectral disponibilidad
Imagen de radar	ondas de radar (activo)	equidistante (~ 5 a 30 m)	costo incoherencias	independiente de nubes
Humedad de suelo	cant. de agua por vol. de tierra	arbitraria	var. indirecta impreciso	fácil rápido
Fotosíntesis	producción de CO_2	arbitraria	costoso impreciso	var. importante no invasivo
Potencial de tallo	conductividad eléctrica tallo	arbitraria	invasivo lento	var. importante económico

El segundo problema de los datos es más difícil de resolver que el primero. Si usamos una imagen multiespectral con K bandas, significa que hay K valores en cada píxel. Por otro lado, las observaciones locales son datos unidimensionales. Entonces, surge un problema a la hora de querer fusionar ambos tipos de datos. En otras palabras: ¿Cómo procesar datos con diferentes dimensiones?

El tercer problema tiene que ver con los valores observados. Supongamos que investigamos un campo con dos patrones A y B . Es posible que midiendo la humedad de suelo en un lugar que corresponde a A nos dé un valor más alto que en B , mientras la fotosíntesis en B es más alta que en A . O sea, la pregunta es: ¿Cómo encontrar patrones si los valores de un patrón pueden ser altos para algunos sensores y bajos para otros sensores? Esto es claramente un problema de ajustar los parámetros correspondientes. Si bien hay algoritmos para acomodar las clases (o patrones) no hay que subestimar este problema.

Entonces, ¿cuál es el objetivo concreto de este capítulo? Supongamos que hay un conjunto de observaciones \mathbf{X} realizadas en los sitios \mathcal{S} . Pues, el objetivo principal es asignar un estado ω_i a cada medición x_i . De esta forma, se obtiene un mapa de patrones que indica que partes del campo estudiado están en las mismas condiciones. A partir de este mapa (mapa de variabilidad) un especialista puede definir un tratamiento especial para cada sección [115].

En la siguiente sección, se presentan tres algoritmos estándar de fusión de datos. Luego se introduce el algoritmo MDF en la sección 5.5.

5.4. Algoritmos estándar

En esta sección, se presentan tres algoritmos clásicos que pueden ser usados para obtener un mapa de variabilidad partiendo de observaciones en sitios arbitrarios. Los tres algoritmos son: *K-medias* (KM), *Mezcla de Gaussianas* (MG) y *Agrupación Jerárquica* (AJ). Antes de introducir cada uno de estos algoritmos, es necesario ajustar los datos observados para poder aplicar los algoritmos.

En caso de trabajar con imágenes multiespectrales y datos unidimensionales es inevitable ajustar los datos para que todas las observaciones tengan la misma dimensión. Existen dos posibles soluciones a este problema. Primero, supongamos que se asigna a cada píxel una coordenada (método estándar). Entonces, hay K observaciones (provenientes de las K bandas) en cada coordenada. En cambio, si asignamos a cada banda una coordenada suavemente distinta (pero aún cerca del centro del píxel) obtenemos K sitios para cada píxel. Note que de esta forma hay exactamente una observación en cada sitio, o sea logramos transformar los datos multiespectrales en datos unidimensionales. Sin embargo, este método no es preferible ya que la transformación de N píxeles a NK sitios hace que consideramos cada banda una medición independiente de las otras bandas. Además, la selección de los sitios es arbitraria y puede tener mucha influencia en el resultado final.

El segundo enfoque consiste en calcular las máximas verosimilitudes de cada observación para cada clase. Entonces, para L clases transformamos cada observación (independiente si es multiespectral o no) a un vector de L probabilidades. Este método es preferible al primer método porque al aplicar distribuciones normales multidimensionales se puede conservar la dependencia que existe entre las distintas bandas de una imagen multiespectral.

Note en este contexto, que calcular las verosimilitudes de cada clase es una herramienta útil para transformar los datos, pero también hay una desventaja: No se tiene en cuenta la relación espacial de las observaciones. Para solucionar este problema, se agregan las coordenadas de cada sitio a su vector observado. De esta forma se garantiza que se conserve la relación espacial de las observaciones.

En resumen, hay un vector de dimensión $L + 2$ para cada observación – L probabilidades y dos coordenadas. Con estos pasos preparativos nos encontramos en condiciones para

aplicar distintos algoritmos a los datos. En lo siguiente, se detallan los fundamentos teóricos de los algoritmos estándar.

5.4.1. K-medias

El algoritmo K-medias es un algoritmo clásico que se basa en el trabajo fundamental de MacQueen [74]. El objetivo del algoritmo es asignar a cada observación $x_i \in \mathbf{X}$ un estado $\omega_i \in \boldsymbol{\omega}$ tal que cada clase sea lo más homogéneo posible. Esto es igual a minimizar para todas las clases la suma de las distancias de cada elemento de una clase a su punto medio.

$$\boldsymbol{\omega} = \arg \min_{\boldsymbol{\omega}} \sum_{\ell=1}^L \sum_{x_i \in \mathbf{X}} \|x_i - \mu_{\ell}\|^2 \quad (5.1)$$

En la ecuación (5.1), μ_{ℓ} es el punto medio de la clase ℓ y $\|\cdot\|$ es la distancia euclidiana.

Para encontrar la configuración que minimiza la ecuación (5.1) se aplica un procedimiento iterativo que consiste en dos pasos. Primero, se asigna a cada observación un estado de acuerdo a la media μ_{ℓ} más cercana. Este paso de asignación se puede formalizar de la siguiente manera.

$$\omega_i^{(p)} = \arg \min_{\ell} \|x_i - \mu_{\ell}^{(p)}\|. \quad (5.2)$$

En la ecuación (5.2), $\omega_i^{(p)}$ se refiere al estado de la observación i en la iteración (p) . Una vez determinado los estados $\boldsymbol{\omega}^{(p)}$, el segundo paso consiste en actualizar las medias de cada clase para la siguiente iteración.

$$\mu_{\ell}^{(p+1)} = \frac{\sum_{x_{\eta} \in \omega_{\ell}} x_{\eta}}{\sum_{x_{\eta} \in \omega_{\ell}} 1}. \quad (5.3)$$

Entonces, $\mu_{\ell}^{(p+1)}$ es la media de la clase ℓ en la iteración $(p+1)$.

Iterando las ecuaciones (5.2) y (5.3) hasta convergencia se obtiene un estado para cada observación. Cabe destacar que el algoritmo siempre converge hacia un mínimo local y por lo tanto no hay garantía teórica de encontrar una configuración de estados $\boldsymbol{\omega}$ que cumple con la ecuación (5.1).

5.4.2. Mezcla de Gaussianas

Otro método para fusionar datos es el algoritmo de Mezcla de Gaussianas (MG). Para hallar los parámetros óptimos de MG (o sea los parámetros de cada distribución gaussiana) se usa un algoritmo iterativo llamado *Expectation Maximization* (EM) [71].

Cabe destacar que los algoritmos MG (juntos con EM) y K-medias son muy parecidos. Solo hay dos diferencias notables. Primero, en vez de usar distancias euclidianas se usa la distribución normal multivariante como medida de similitud. Y segundo, se asigna un peso a cada clase como probabilidad a priori. En otras palabras, las probabilidades a priori pueden ser no-uniformes como expresado en la ecuación (5.4).

$$P(\omega_i = \ell) \neq \frac{1}{L}. \quad (5.4)$$

En el esquema MG se supone que cada clase se observa a través una distribución normal multivariante con media μ y desviación estándar Σ . Entonces, la probabilidad de observar x_i dado $\omega_i = \ell$ es

$$P(x_i|\omega_i = \ell) = \frac{1}{(2\pi)^{k/2}|\Sigma_\ell|^{1/2}} \exp \left\{ -\frac{1}{2}(x_i - \mu_\ell)^T \Sigma_\ell^{-1} (x_i - \mu_\ell) \right\}. \quad (5.5)$$

Usando la verosimilitud de la ecuación (5.5) y la probabilidad a priori se puede aplicar la fórmula de Bayes para obtener la probabilidad a posteriori de cada observación.

$$P(\omega_i = \ell|x_i) \propto P(x_i|\omega_i = \ell)P(\omega_i = \ell). \quad (5.6)$$

Por ende, es fácil determinar el estado de cada observación usando la ecuación (5.6). Lo único que hay que hacer es determinar las medias μ y las desviaciones estándar Σ de cada clase. Para resolver este problema, es necesario correr el algoritmo EM.

Igual que K-medias el algoritmo EM consiste de dos pasos. Primero, se asigna un estado a cada observación de acuerdo a

$$\omega_i^{(p)} = \arg \max_{\ell} P \left(x_i | \mu_\ell^{(p)}, \Sigma_\ell^{(p)} \right) c_\ell^{(p)}. \quad (5.7)$$

En la ecuación (5.7), $c_\ell^{(p)}$ es la probabilidad a priori de la clase ℓ en la iteración (p). Luego, se actualizan las probabilidades a priori usando

$$c_\ell^{(p+1)} = \frac{\sum_{x_\eta \in \omega_\ell} 1}{\sum_{x_\nu \in \mathbf{X}} 1}. \quad (5.8)$$

En el segundo paso, se estiman los valores de $\mu^{(p+1)}$ y de $\Sigma^{(p+1)}$ para la siguiente iteración. El parámetro $\mu^{(p+1)}$ se obtiene usando la ecuación (5.3) mientras el estimador de $\Sigma^{(p+1)}$ viene dado por

$$\Sigma_\ell^{(p+1)} = \frac{\sum_{x_\eta \in \omega_\ell} (x_i - \mu_\ell^{(p+1)}) (x_i - \mu_\ell^{(p+1)})^T}{\sum_{x_\eta \in \omega_\ell} 1}. \quad (5.9)$$

Finalmente, hay que iterar las ecuaciones (5.7), (5.8), (5.3) y (5.9) hasta que no se registren más cambios de estados de una iteración a la siguiente. Entonces, el algoritmo EM convergió. Pero igual que en el caso de K-medias, EM solo converge a un óptimo local.

5.4.3. Agrupación Jerárquica

Por último, se presenta el algoritmo de Agrupación Jerárquica (AJ). Este algoritmo apunta a encontrar las similitudes y disimilitudes de cada par de observaciones de forma aglomerativa. Esto quiere decir que al principio cada observación comienza en su propia clase y de a poco se van juntando clases en cada escalón jerárquico hasta que todas las observaciones se agrupan en una sola clase.

En el algoritmo 5 se presenta un esquema general de AJ.

Algoritmo 5: Agrupación jerárquica (AJ)

1. Encontrar las similitudes y disimilitudes de cada par de observaciones
 2. Agrupar las observaciones en un árbol binario, jerárquico como en la figura 5.1
 3. Elegir dónde cortar el árbol
-

En el primer paso de AJ se aplica la distancia euclidiana para determinar la similitud de dos observaciones. Obviamente, es posible usar otras medidas como la distancia

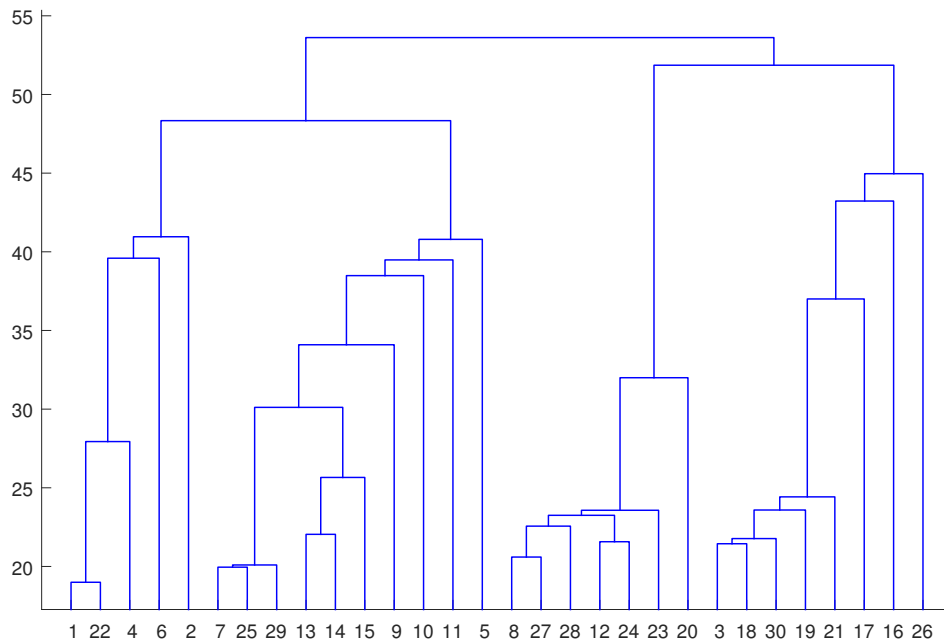


FIGURA 5.1: Ejemplo de un dendrograma. El eje vertical representa la distancia entre dos clases mientras el eje horizontal corresponde a las observaciones.

de Chebychev o la distancia de Mahalanobis, pero estas medidas tienen características particulares y por lo tanto no sirven para un algoritmo universal de referencia.

El segundo paso de AJ funciona de la siguiente forma. Se anotan todas las distancias entre las observaciones en una lista de tal forma que la primera entrada es la distancia más chica y la última entrada es la distancia más grande. Ahora, se forma una clase uniendo las dos observaciones más cercanas. Luego, se calculan las distancias de esta nueva clase a todas las observaciones restantes y se agregan los resultados a la lista de distancias. A partir de ahí se van uniendo los elementos segundo más cercanos, luego los tercero más cercanos y así sucesivamente. De esta forma se obtiene un dendrograma que indica cómo se agrupan las observaciones. En la figura 5.1 se muestra el ejemplo de un dendrograma.

Note que, todas las entradas de la lista (excepto la primera entrada) pueden ser dos observaciones individuales, una clase y una observación, o dos clases. O sea, es posible que en los primeros pasos se generen clases con varios elementos, mientras algunas observaciones quedan aisladas.

Para completar el segundo paso de AJ falta definir como se calcula la distancia entre dos clases. En este trabajo se aplica la medida de mínimo enlace, que significa que la

distancia D entre dos clases A y B viene dada por

$$D(A, B) = \min_{a \in A, b \in B} \{dist(a, b)\}. \quad (5.10)$$

Finalmente, hay que decidir dónde cortar el árbol. Si la cantidad de clases es desconocida, es un desafío saber dónde cortar el árbol (dendrograma) pero como en este trabajo la cantidad de clases es fija (L clases) es fácil obtener la configuración final. Simplemente hay que cortar el dendrograma a la altura donde existan L clases.

5.5. Makorv Data Fusion

En esta sección se presenta un nuevo método para la fusión de datos. El algoritmo denominado *Markov Data Fusion* (MDF) es una extensión de SBM del capítulo 4 a observaciones en sitios arbitrarios.

En comparación con los algoritmos de referencia, MDF tiene dos ventajas importantes. Primero, las coordenadas de cada observación son procesadas independientemente del valor observado. Entonces, hay una clara separación entre información contextual (observaciones cercanas) y pertenencia a una clase por el valor medido. La segunda ventaja es que MDF permite usar cualquier función de densidad para modelar datos unidimensionales. Para datos multidimensionales es necesario aplicar una distribución multivariante como la distribución normal o implementar una cópula [116].

Recuerde que el primer paso de SBM consiste en calcular las probabilidades marginales para cada observación de acuerdo a la ecuación (4.6) de la sección 4.3.3. Este paso es igual para MDF porque aquí solo se evalúan las observaciones de forma independiente sin tener en cuenta las coordenadas.

En el segundo paso de SBM, se utiliza el BF como especificado en la ecuación (4.9) para incorporar información contextual. El kernel del BF viene dado por la ecuación (4.10). En caso de MDF, las formulas son las mismas. La única diferencia es que para sitios arbitrarios hay que calcular la distancia de cada observación a todas las otras observaciones para encontrar sitios vecinos. En cambio, para píxeles equidistantes, como se usan en SBM, la configuración de vecinos es fija (ver figura 2.3 para las posibles configuraciones).

En contrario a los primeros dos pasos, hay que implementar ajustes significantes al tercer paso de MDF para poder procesar observaciones de sitios arbitrarias. En caso de SBM es suficientes seguir el esquema de la figura 4.5 porque se aprovecha implícitamente el hecho que cada píxel tiene las mismas coordenadas. Entonces, es fácil transferir los resultados de la banda k a la banda $k + 1$, porque el estado de un píxel tiene que ser el mismo en todas las bandas. En cambio, MDF necesita relacionar matemáticamente los sitios (y las observaciones) de un sensor con los sitios de otro sensor.

Recuerde que el esquema del tercer paso de SBM consistía en iterar las ecuaciones (4.23) y (4.24). Ahora, para MDF hay que modificar estas dos ecuaciones. Además, es necesario agregar una ecuación para conectar los estados de los distintos sensores.

Antes de presentar las fórmulas de MDF hay que definir algunas variables. En lo siguiente se llama $\omega^{(k)}$ a los estados del sensor k y $\mathbf{X}^{(k)}$ a las correspondientes observaciones. A parte, se denomina $s_i^{(k)}$ al sitio de medición i del sensor k . Por último, llamamos $q_i^{(k)*}$ a las probabilidades suavizadas (con el BF) del sensor k en el sitio i . Usando esta notación, el tercer paso de MDF viene dado por las siguientes ecuaciones.

Primero, se calcula una estimación de los estados del primer sensor (o sea $\omega^{(1)}$) usando el algoritmo K-medias o MG. Con estos estados se determina la relación espacial que tiene cada medición del segundo sensor con las mediciones del primer sensor, separados por clases.

$$d_i^{(2)}(\ell) = \sum_{\omega_j^{(1)}=\ell} dist(s_i^{(2)}, s_j^{(1)}). \quad (5.11)$$

En la ecuación (5.11) $d_i^{(2)}(\ell)$ es una medida que indica cuán relacionado está el sitio $s_i^{(2)}$ con el estado ℓ del primer sensor usado como función de distancia $dist$ la distancia euclidiana. Luego, se asigna al sitio $s_i^{(2)}$ un estado de acuerdo a

$$\omega_i^{(2)} = \arg \max_{l \in \{1, 2, \dots, L\}} d_i^{(2)}(l). \quad (5.12)$$

En el siguiente paso, se calcula una variable auxiliar c que representa la influencia de información contextual de las observaciones del mismo sensor.

$$c_i^{(2)}(\ell) = \sum_{\omega_n^{(2)}=\ell, n \neq i} dist(q_i^{(2)*}, q_n^{(2)*}). \quad (5.13)$$

En la ecuación (5.13) $c_i^{(2)}(\ell)$ representa la evidencia en favor de la clase ℓ de acuerdo a los sitios vecinos del sitio i .

Finalmente, se refinan los estados de la ecuación (5.12) usando información contextual de la ecuación (5.13). Entonces, tanto la información contextual como la relación con los estados del sensor anterior influyen en los estados del segundo sensor. (5.12)

$$\omega_i^{(2)} = \arg \max_{\ell \in \{1,2,\dots,L\}} \left\{ c_i^{(2)}(\ell) * d_i^{(2)}(\ell) \right\}. \quad (5.14)$$

Note que, la ecuación (5.14) es una extensión de la ecuación (4.24) de SBM.

Una vez evaluada la ecuación (5.14) se puede generalizar el procedimiento de MDF de la siguiente manera. Primero, se calcula la influencia de los sitios ya procesados:

$$d_i^{(k)}(\ell) = \sum_{l=1}^{k-1} \sum_{\omega_j^{(l)}=\ell} dist\left(s_i^{(k)}, s_j^{(l)}\right). \quad (5.15)$$

Luego, se asigna un estado provisorio a cada observación del sensor k .

$$\omega_i^{(k)} = \arg \max_{\ell \in \{1,2,\dots,L\}} d_i^{(k)}(\ell). \quad (5.16)$$

Y finalmente, se incorpora información contextual.

$$c_i^{(k)}(\ell) = \sum_{\omega_n^{(k)}=\ell, n \neq i} dist\left(q_i^{(k)*}, q_n^{(k)*}\right). \quad (5.17)$$

Entonces, los estados del sensor k vienen dados por

$$\omega_i^{(k)} = \arg \max_{\ell \in \{1,2,\dots,L\}} \left\{ c_i^{(k)}(\ell) * d_i^{(k)}(\ell) \right\}. \quad (5.18)$$

En resumen, se usan las ecuaciones (5.15), (5.16), (5.17) y (5.18) para ir incorporando sensores al proceso de asignación de estados. Después de procesar el último sensor K se empieza de nuevo con el primer sensor hasta que todos los estados converjan. Pero como todas las observaciones de todos los sensores ya tienen asignada un estado, se usa la siguiente ecuación en vez de empezar de nuevo con la ecuación (5.11).

$$d_i^{(k)}(\ell) = \sum_{l \neq k} \sum_{\omega_j^{(l)}=\ell} dist\left(s_i^{(k)}, s_j^{(l)}\right). \quad (5.19)$$

En el algoritmo 6 se presenta un esquema de MDF. Además, se muestra un diagrama del último paso de MDF en la figura 5.2.

Algoritmo 6: Markov Data Fusion (MDF)

1. Iniciar estados del primer sensor
 2. Aplicar las ecuaciones (5.11), (5.12), (5.13) y (5.14)
 3. Procesar sucesivamente los sensores usando las ecuaciones (5.15), (5.16), (5.17) y (5.18)
 4. Iterar las ecuaciones (5.19), (5.16), (5.17) y (5.18) hasta convergir
-

En la siguiente sección, se comparan los algoritmos K-medias, MG, AJ y MDF con respecto a sus suposiciones y su coste computacional. En el capítulo 7, se evalúan todos los métodos para datos reales y simulados.

5.6. Comparación de algoritmos

Los algoritmos de referencia (K-medias, MG y AJ) y el algoritmo propuesto en este trabajo se diferencian notablemente por sus suposiciones. Mientras K-medias y MG asumen que existe una distribución oculta para cada clase, AJ se basa principalmente en la idea de unir observaciones de acuerdo a su similitud que es expresado por la distancia euclidiana. Para subrayar esto, hacemos hincapié en que K-medias es una simplificación de MG suponiendo que las matrices de covarianza son matrices identidad. En resumen, K-medias y MG son métodos generativos y AJ es un algoritmo jerárquico, aglomerativo.

Comparando MDF con los algoritmos de referencia, se observa que MDF es un punto intermedio en cuanto al modelado de los datos. Por un lado, se tienen en cuenta las distancias euclidianas de los sitios de distintos sensores, pero por otro lado se respetan las probabilidades de cada clase para las observaciones del mismo sensor.

Otro aspecto importante es el costo computacional. En la tabla 5.2 se muestran los órdenes de los distintos algoritmos. Como cada algoritmo depende de distintos parámetros (como la cantidad de mediciones, la cantidad de clases, etc.) es difícil determinar cuál es el algoritmo más eficiente. Sin embargo, es evidente que MDF es más complejo que K-medias y MG para cualquier configuración y más complejo que AJ si la cantidad de

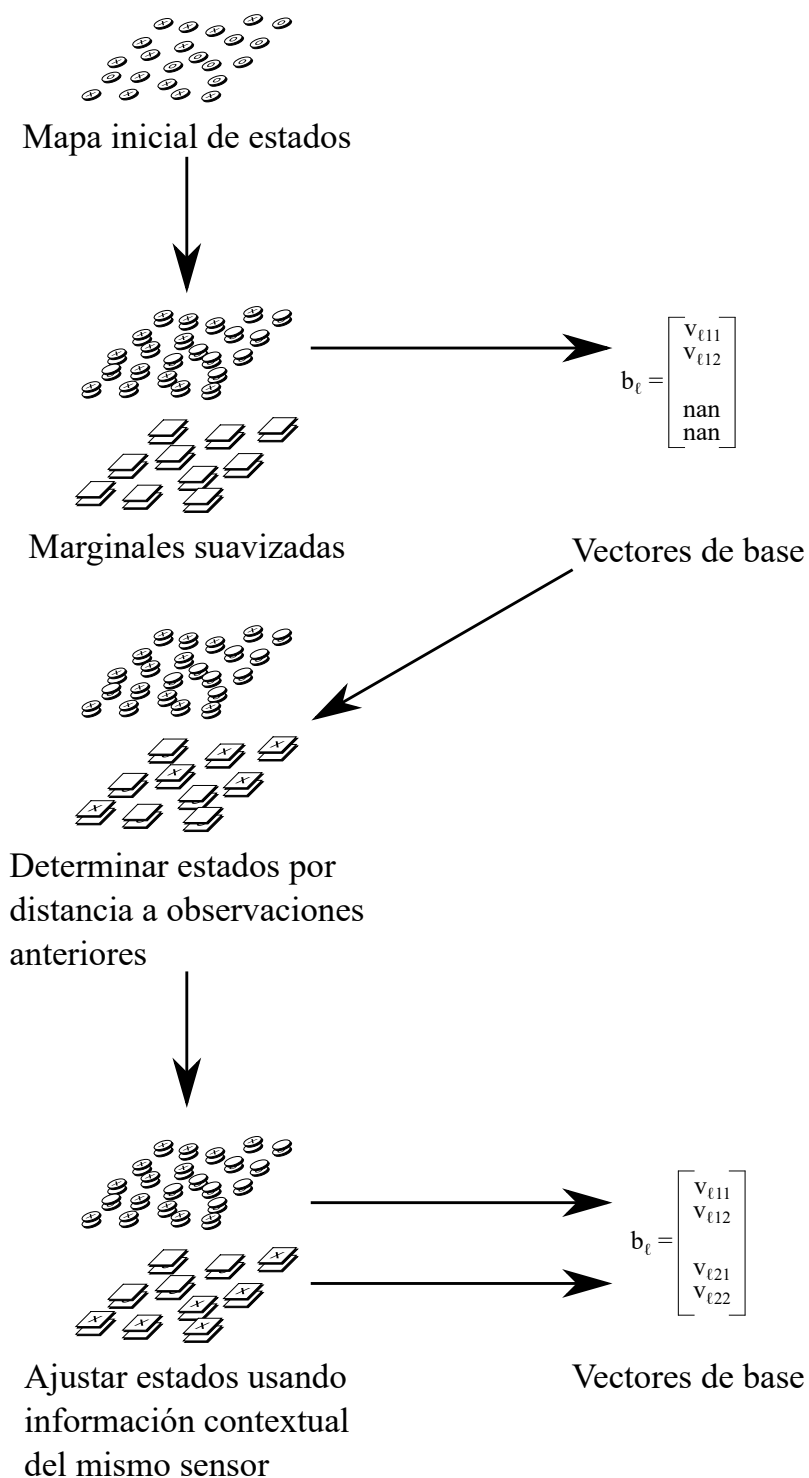


FIGURA 5.2: Esquema del tercer paso del algoritmo MDF para dos sensores y estados.

CUADRO 5.2: Complejidad de algoritmos para la fusión de datos suponiendo que hay i iteraciones, L clases y n mediciones.

K-medias	$\mathcal{O}(inL^2)$
MG	$\mathcal{O}(in^2L^2)$
AJ	$\mathcal{O}(n^3)$
MDF	$\mathcal{O}(in(L^2 + n + 4))$

mediciones es baja o mediana. En los experimentos con datos reales y simulados de este trabajo (ver capítulo 7) se estableció el siguiente orden de tiempos computacionales: El algoritmo más rápido fue AJ seguido por K-medias, MG y finalmente MDF.

En el siguiente capítulo se introducen los sensores usados en este trabajo. Luego, se presentan los resultados experimentales en el capítulo 7

Capítulo 6

Sensores para agricultura de precisión

En este capítulo se introducen algunos sensores típicos de agricultura de precisión. Además de detallar las técnicas y los subyacentes procesos físicos, se destacan las particularidades de cada sensor. Por lo tanto, se puede entender este capítulo como una introducción al capítulo siguiente, donde se presentan los resultados experimentales.

6.1. Teledetección

La palabra teledetección se refiere a la adquisición de información de un objeto o fenómeno a distancia sin contacto físico. Las aplicaciones más comunes en esta área son imágenes capturadas por aviones o satélites. Sin embargo, existen otras maneras de detección remota, como, por ejemplo, a través de barcos o aeronaves.

Hay dos formas de llevar a cabo la colección de información: teledetección pasiva y teledetección activa. En la teledetección activa se emite una radiación hacia el objeto de interés y luego se graban las ondas reflejadas. Un ejemplo para este tipo de teledetección son imágenes de radar donde primero se emiten ondas electromagnéticas y después se observa la respuesta.

En contrario a la teledetección activa no se emiten ondas en la teledetección pasiva. Solo se observa la radiación que emite el objeto de interés. Entonces, es necesario tener otra

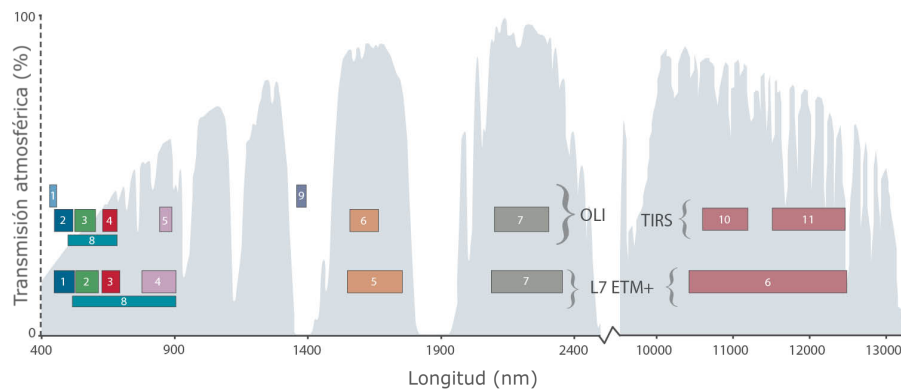


FIGURA 6.1: Comparación entre las bandas de Landsat 7 (sensor ETM+) y Landsat 8 (sensores OLI y TIRS).

Fuente: L. Rocchio y J. Barsi. Imagen tomada del sitio <http://landsat.gsfc.nasa.gov/?p=3186> y modificada por J. Baumgartner (traducción a castellano)

fuente de energía para que el objeto observado emita radiación. Típicamente esta fuente es la luz solar. Como consecuencia, la teledetección pasiva no es aplicable durante la noche.

6.1.1. Teledetección pasiva

Para especificar las ventajas y desventajas de la teledetección pasiva y activa es importante conocer las propiedades de las ondas en cuestión. Empezamos con la teledetección pasiva. Como se mencionó, la teledetección pasiva necesita una fuente de energía que es generalmente la luz solar. Por consiguiente, las ondas capturadas corresponden aproximadamente a las ondas visibles y por lo tanto no pueden pasar por obstáculos como nubes.

Por ejemplo, el satélite Landsat 8 está equipado con un sensor llamado *Operational Land Imager* (OLI). Este sensor captura ondas con longitudes entre $0,433 \mu\text{m}$ y $1,390 \mu\text{m}$. Además del sensor OLI el satélite Landsat 8 lleva otro sensor pasivo que registra ondas termales del espectro infrarrojo. Este sensor se llama *Thermal InfraRed Sensor* (TIRS). En la figura 6.1 se muestra una comparación entre las bandas cubiertas por Landsat 7 y Landsat 8 en el contexto de la transmisión atmosférica.

Otro ejemplo de un sensor pasivo es el *Airborne Visible/Infrared Imaging Spectrometer* (AVIRIS). Este sensor vuela a bordo de un avión ER-2 jet (altura 20,000 m, velocidad 730 km/h) o una Twin Otter International's turboprop (altura 4,000 m, velocidad 130 km/h). El sensor es capaz de diferenciar 224 bandas en el espectro de 0,4 μm a 2,5 μm .

6.1.2. Teledetección activa

En la teledetección activa se aplica principalmente la tecnología de radar que permite longitudes de onda de aproximadamente 1 mm a 1 m. La ventaja de estas ondas electromagnéticas es que atraviesan nubes. Además, la teledetección activa no depende de la luz solar ya que se trabaja con ondas reflejadas.

La mayoría de la teledetección activa se lleva a cabo usando un *Synthetic-Aperture Radar* (SAR). En este tipo de radar se emiten ondas electromagnéticas perpendicularmente a la dirección de vuelo hacia la tierra y luego se capturan las ondas reflejadas. Hay cuatro factores que influyen en el resultado final:

1. Factores físicos, como el constante dieléctrico del suelo que depende entre otras cosas de la humedad de suelo
2. Factores geométricos, como la rugosidad del suelo o pendientes
3. El tipo de suelo (vegetación, concreto, etc.)
4. Frecuencias y polarización de la onda aplicada

6.1.3. Imágenes satelitales vs. imágenes aéreas

Tanto los sensores pasivos como los sensores activos pueden ser montados a satélites o aviones. En esta sección se presentan algunas ventajas y desventajas de los dos métodos.

En cuanto imágenes satelitales, hay que destacar que los satélites pasan por un sitio específico generalmente en un intervalo de tiempo fijo. Por ejemplo, el satélite Landsat 8 toma una imagen de un sitio cada ocho días. En cambio, un avión puede tomar imágenes en cualquier momento (teniendo en cuenta que sensores pasivos no funcionan de noche).

Otro aspecto importante es que, al usar un avión, se puede variar la altura de vuelo y luego es posible evitar ciertos tipos de nubes. Por ejemplo, las nubes de las familias

Cirrus, Cirrostratus y Cirrocúmulus se encuentran en alturas de 7 km hasta 10 km y por lo tanto interfieren en imágenes satelitales, pero no en imágenes tomadas desde un avión.

Además, los sensores montados en aviones tienden a tener una mejor resolución que las imágenes satelitales – tanto espacial como espectral. Por otro lado, las imágenes satelitales cubren áreas mucho mayores que las imágenes tomadas desde aviones.

Por último, cabe destacar que generalmente es más fácil acceder a imágenes satelitales que a imágenes aéreas. Por ejemplo, las imágenes Landsat 7 y Landsat 8 se pueden bajar gratuitamente de la web, mientras la mayoría de las imágenes aéreas tienen costo.

En la siguiente sección, se presentan sensores locales para mediciones *in situ*. Estas mediciones se pueden entender como un complemento de las imágenes de teledetección.

6.2. Sensores locales

En las últimas dos décadas se ha desarrollado una gran variedad de sensores locales para la agricultura de precisión [117]. Algunos sensores, como el medidor de flujo de savia o el medidor del grosor de la hoja, son diseñados para aplicaciones muy particulares. Otros sensores, como mini rizotrones, tienen un alto costo, mientras sensores como dendrómetros electromecánicos requieren mucho tiempo y dedicación para obtener datos útiles [118]. Por lo tanto, dejamos de lado estos tipos de sensores y nos enfocamos en los siguientes tres sensores: medidor de humedad de suelo, medidor de fotosíntesis y medidor de potencial de tallo.

6.2.1. Humedad de suelo

El índice de humedad de suelo se refiere a la cantidad de agua por volumen de tierra. Por lo tanto, este valor es sumamente importante para determinar el momento y la cantidad del riego teniendo en cuenta la variabilidad espacial del campo. Además, la humedad de suelo permite sacar conclusiones acerca del estado del cultivo [29]. Si la humedad de suelo es baja, la planta sufre de estrés hídrico mientras que, en caso de exceso de agua, se arrastran los nitratos a una profundidad superior al alcance de las raíces [119].

La forma más exacta para medir la humedad de suelo consiste en pesar una muestra de tierra antes y después del secado. Sin embargo, hay otro tipo de sensor que mide la

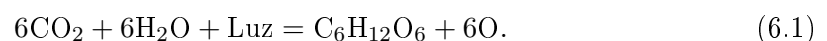


FIGURA 6.2: Medidor de humedad de suelo.

conductividad del suelo. Esta variable se puede transformar en humedad de suelo ya que la conductividad aumenta a medida que aumente la humedad de suelo. La ventaja de este tipo de sensor es que el resultado está disponible instantáneamente. A parte, no es necesario llevar muestras de tierra. Por lo tanto, se prefiere este tipo de sensor de humedad de suelo en este trabajo. En la figura 6.2 se muestra dicho sensor.

6.2.2. Fotosíntesis

Otro método para conocer el estado de un cultivo es medir la fotosíntesis. Recuerde que la fotosíntesis de una planta viene dada por



Para acceder a la tasa de O de la ecuación (6.1) se mide la concentración de CO_2 y H_2O en un flujo de aire que pasa por una hoja expuesta al rayo del sol. Mientras más alta la concentración de CO_2 , más fotosíntesis está haciendo la planta.

Si bien el índice de fotosíntesis de un cultivo es muy importante para optimizar su crecimiento, hay dos desventajas destacables con respecto al aparato. Primero, el medidor tiene un alto coste de adquisición y segundo los resultados obtenidos son muy imprecisos.

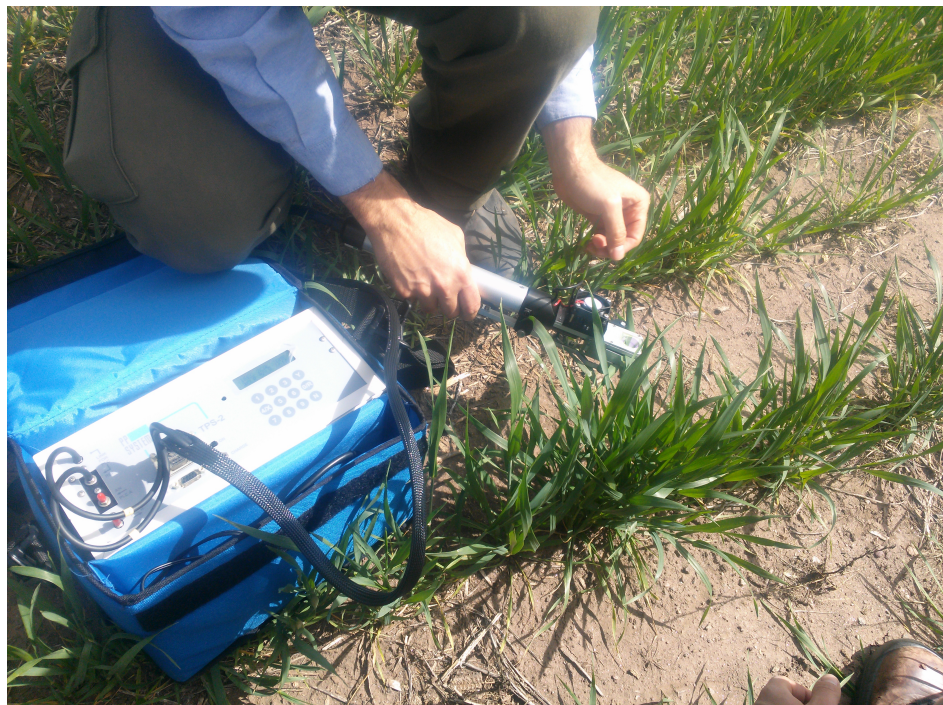


FIGURA 6.3: Medidor de fotosíntesis TPS-2.

La razón de la mala relación señal/ruido es que se trata de un complejo procesos químico que depende de muchos factores como la intensidad de la luz o el flujo de aire. En este trabajo se usa un medidor TPS-2 como se muestra en la figura 6.3.

6.2.3. Potencial de tallo

Por último, se desarrolla un nuevo sensor de bajo costo denominado *Medidor de potencial de tallo*. Este medidor pincha el tallo de una planta en dos lugares cercanos y luego aplica una corriente alterna con distintas frecuencias (una señal *chirp*) a uno de los dos pinchazos. Al mismo tiempo, se mide el potencial eléctrico en el otro pinchazo. De esta manera se obtiene una especie de diagrama de Bode del tallo. En primer lugar, este diagrama contiene información acerca de la conductividad eléctrica del tallo. En segundo lugar, se pueden sacar conclusiones acerca del estado del tallo con el diagrama Bode. Por ejemplo, una planta que sufre estrés hídrico tiende a tener una resistencia más alta que una planta sana. Además, se pretende usar los datos observados para comprobar si a la planta le faltan minerales.

En la literatura existen varios métodos para medir propiedades eléctricas de la planta, pero la gran mayoría de estos medidores son no-invasivos [120, 121]. Esto quiere decir

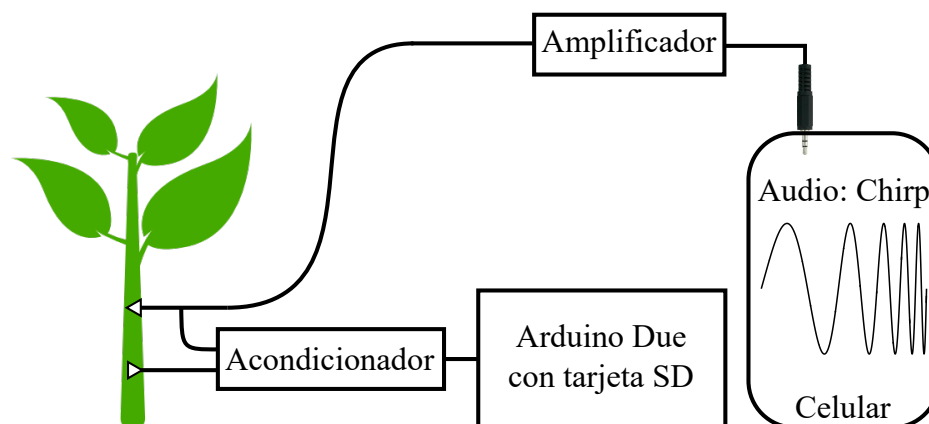


FIGURA 6.4: Esquema principal del medidor de potencial de tallo.

que los electrodos solo se apoyan en distintas partes de la planta (generalmente en las hojas). En cambio, el medidor desarrollado en esta tesis es un aparato invasivo. Por otro lado, se espera adquirir más información acerca del estado de la planta ya que se estudia directamente la parte más importante de cualquier cultivo: el tallo.

A la hora de construir un medidor de potencial de tallo aparecen múltiples problemas. Primero, no es fácil generar una onda cuya frecuencia varía entre ciertos límites. Segundo, hay que procesar y grabar los resultados en tiempo real.

Para resolver el primer problema se usa un celular con reproductor de música y salida de 3,5 mm jack. De esta manera se puede generar una señal analógica con frecuencias entre 20 Hz y 20 kHz. Amplificando esta señal, se obtienen los voltajes para el primer pinchazo (entrada). Para grabar la señal de entrada y la medición en el segundo pinchazo se utiliza un microprocesador Arduino Due[®] con una frecuencia de 84 MHz juntos con una tarjeta SD. La alimentación del Arduino Due[®] y la energía del amplificador viene de dos baterías de 9 Volt (una batería alimenta el Arduino y la otra es para el amplificador). En la figura 6.4 se presenta el esquema general del medidor de potencial de tallo y en la figura 6.5 se muestra como se usa el medidor en un lote agrícola.

Para acceder a los datos grabados en formato csv en la tarjeta SD se usa Matlab[®]. Cabe destacar que, después de realizar una medición de potencial de tallo es necesario anotar las respectivas coordenadas de GPS manualmente.

En el siguiente capítulo, se presentan los resultados experimentales, tanto de las imágenes de teledetección como de sensores locales. Luego, se evalúan los resultados en el capítulo 8.

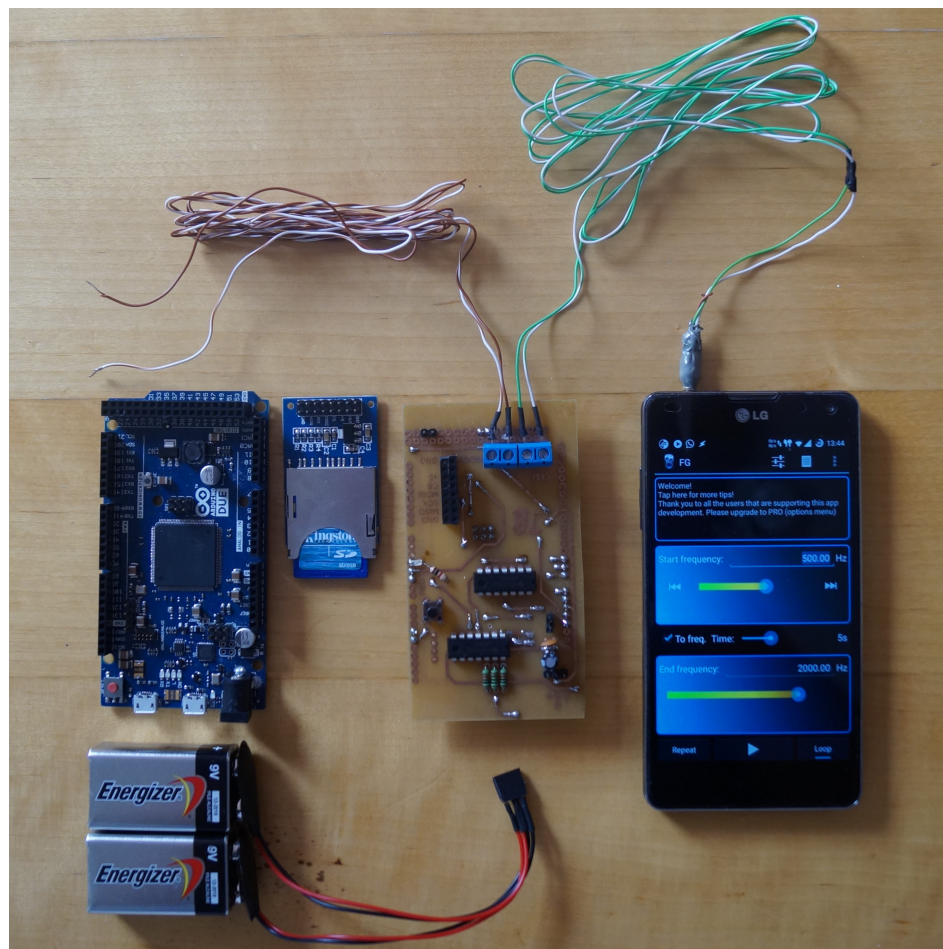


FIGURA 6.5: Medidor de potencial de tallo con sus componentes: Arduino Due[®], celular, placa, cable 3,5 mm jack (verde-blanco), cable para pinchar la planta (marrón-blanco), tarjeta SD, dos baterías de 9 V (no conectadas en este momento).

Capítulo 7

Resultados experimentales

El objetivo principal de este capítulo es evaluar y comparar los algoritmos de la literatura con los algoritmos desarrollados en este trabajo. A nivel de segmentación de imágenes, los algoritmos de referencia son PCVT (basado en modelos ocultos de Markov de dos dimensiones) e ICM (basado en el modelo de Potts). Los algoritmos competidores desarrollados en esta tesis son CEP (basado en modelos ocultos de Markov de dos dimensiones) y SBM (basado en el modelo de Potts).

En cambio, para la fusión de datos se usan los algoritmos K-medias, MG y AJ como referencia para evaluar el desempeño de MDF. Recuerde que el algoritmo MDF fue desarrollado en esta tesis como extensión de SBM.

Cabe destacar que tanto para la segmentación de imágenes como para la fusión de datos se agrega otro método de referencia llamado Máxima Verosimilitud (MV). Este algoritmo solo tiene en cuenta la probabilidad de cada observación sin usar datos contextuales. Por lo tanto, el algoritmo MV viene dado por

$$\omega_i = \arg \max_{\ell \in \mathcal{L}} \left\{ p(O_i | \ell, \Theta_\ell) \right\}. \quad (7.1)$$

En la ecuación (7.1) O_i es el valor observado en el sitio i (un píxel de una imagen o una medición *in situ*) y Θ_ℓ es la función de probabilidad de la clase ℓ .

Este capítulo se divide en cuatro partes. Primero se introducen los parámetros usados y los coeficientes de evaluación en la sección 7.1. Luego se presentan los resultados de imágenes de teledetección (pasiva y activa) en la sección 7.2. A continuación, se evalúan

los algoritmos para imágenes de referencia en la sección 7.3. Las imágenes usadas son imágenes de rayos X, imágenes simuladas y una imagen de referencia estándar. Por último, se presentan los resultados de los algoritmos de fusión de datos para datos reales y simulados en la sección 7.4.

7.1. Parámetros y coeficientes

En esta sección se fijan los parámetros de los algoritmos estudiados. Además, se introduce un coeficiente que sirve para evaluar los resultados en caso de contar con la verdad de campo.

7.1.1. Parámetros de los algoritmos

Para los algoritmos PCVT, CEP y SBM no existe una garantía teórica de converger a un resultado final (ya sea un óptimo local o global). Solo en caso de ICM se puede demostrar que el algoritmo siempre converge a un óptimo local, mientras el algoritmo MV no es un método iterativo y por lo tanto no es afectado por problemas de convergencia.

Entonces, surge la pregunta: ¿Después de cuantas iteraciones se corta el proceso iterativo de PCVT, CEP y SBM en caso de no converger? Para todos los experimentos de este trabajo se fija un máximo de 150 iteraciones.

Cabe destacar que PCVT, CEP y SBM siempre logran asignar un estado definitivo a casi todas las observaciones dentro de 100 iteraciones. Solo en algunos casos aislados se observa que a partir de una determinada iteración dos a cinco píxeles van cambiando estados en un ciclo repetitivo. Para entender este comportamiento hay que recordar que después de cada iteración se re-estiman las distribuciones de observación. Este pequeño ajuste en las densidades puede ser suficiente para que el píxel vuelva al estado anterior en la siguiente iteración. Por lo tanto, el algoritmo se mantiene en un ciclo de rotar los estados de algunos pocos píxeles hasta llegar a la iteración 150.

Los algoritmos ICM y SBM requieren una vecindad especificada por el usuario. En este trabajo se usa por estándar una vecindad 5×5 (24 vecinos). Solo en algunos casos se aplica una vecindad 3×3 (8 vecinos) o 7×7 (48 vecinos) para investigar la influencia de la vecindad, pero siempre el texto hace referencia a cualquier cambio de vecindad.

El algoritmo PCVT tiene como único parámetro la máxima cantidad de secuencias por diagonal. En este trabajo se fija un máximo de 200 secuencias. Este número es lo suficientemente alto para no descartar secuencias importantes, pero al mismo tiempo no implica un coste computacional demasiado alto. Cabe destacar que, al reducir la máxima cantidad de secuencias probables, el resultado de PCVT se acerca al de MV. Incluso PCVT se convierte en MV al permitir solo una secuencia por diagonal.

En algunos experimentos se muestra la imagen original y también la misma imagen con contraste aumentado. Esto se hace para facilitar la interpretación de los resultados para el lector. Sin embargo, se usan exclusivamente las imágenes originales para correr los algoritmos MV, PCVT, CEP, ICM y SBM.

Todos los algoritmos fueron implementados en el lenguaje de programación Matlab[®]. Los experimentos se corrieron en una PC con Windows[®] 8.1/10 con procesador i3 y 4 GB RAM. Se usó la versión 2015a de Matlab[®] junto con las cajas de herramientas *Image Processing Toolbox*, *Mapping Toolbox* y *Statistics and Machine Learning Toolbox*.

Por último, se fija que todos los algoritmos se inician con el método MG usando EM. En caso de utilizar datos de entrenamiento para iniciar los algoritmos se hace referencia en el texto.

7.1.2. El coeficiente κ de Cohen

En la mayoría de los experimentos de las secciones 7.2, 7.3 y 7.4 está disponible la verdad de campo. Por lo tanto, se pueden calcular distintos coeficientes que expresan la concordancia entre los resultados de los algoritmos y la verdad de campo.

Algunos ejemplos son, el coeficiente de correlación de Pearson [122] o el coeficiente de correlación de Spearman [123]. Sin embargo el coeficiente κ de Cohen [34] es uno de los coeficientes más usados en los últimos años [124, 125]. El coeficiente viene dado por

$$\hat{\kappa} = \frac{P_O - P_E}{1 - P_E}. \quad (7.2)$$

En la ecuación (7.2), P_O es la concordancia relativa entre el resultado de un algoritmo y la verdad de campo, mientras P_E es la probabilidad hipotética de concordancia por azar.

Para aclarar cómo se calculan los términos P_O y P_E se presenta un pequeño ejemplo basado en la tabla 7.1.

CUADRO 7.1: Ejemplo como calcular κ para tres clases. En esta tabla, n_{ij} es la cantidad de píxeles con estado j al que el algoritmo asigno el estado i .

		Verdad de campo			
		Clase 1	Clase 2	Clase 3	Σ
Resultado	Clase 1	n_{11}	n_{12}	n_{13}	n_{1+}
	Clase 2	n_{21}	n_{22}	n_{23}	n_{2+}
	Clase 3	n_{31}	n_{32}	n_{33}	n_{3+}
	Σ	n_{+1}	n_{+2}	n_{+3}	N

Usando los datos de la tabla 7.1 se obtiene P_0 y P_E de acuerdo a

$$\begin{aligned}
 P_0 &= \frac{\sum_{i=1}^3 n_{ii}}{N} \\
 P_E &= \frac{\sum_{i=1}^3 n_{+i}n_{i+}}{N^2}.
 \end{aligned}
 \tag{7.3}$$

En la ecuación (7.3), N es la suma de todos los estados, o sea $N = \sum_{i=1}^3 \sum_{j=1}^3 n_{ij}$. En las siguientes secciones se usa el coeficiente κ para evaluar y comparar los algoritmos MV, ICM, PCVT, CEP y SBM.

7.2. Imágenes de teledetección

En esta sección se presentan los resultados de segmentación de imágenes aéreas. En primer lugar, se usan imágenes ópticas obtenido a través de teledetección pasiva. Luego se aplican los algoritmos a una imagen de radar.

7.2.1. Imágenes ópticas (teledetección pasiva)

En la sección 6.1 se detallaron las ventajas y desventajas de teledetección pasiva. Ahora se presentan los resultados para tres imágenes con características muy diferentes.

Landsat 7

La imagen de esta sección fue tomada por el sensor *Enhanced Thematic Mapper Plus* (ETM+) que vuela a bordo del satélite Landsat 7 en una altura de 705 km. El ETM+ es

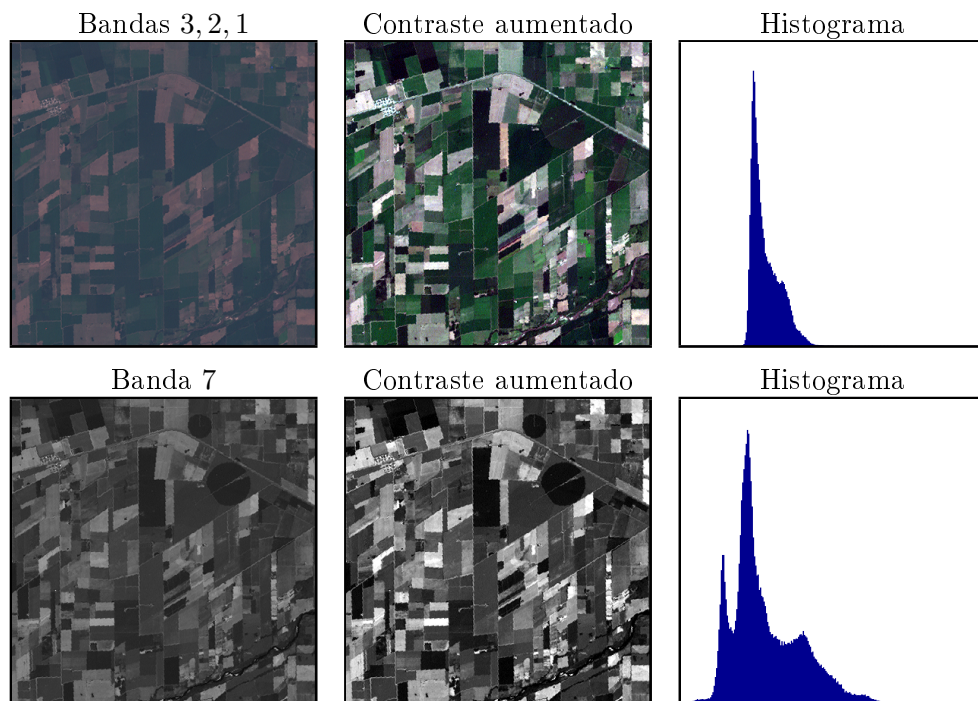


FIGURA 7.1: Imagen Landsat 7 de una zona agrícola. Solo en la banda 7 resaltan algunos detalles como los dos pivotes de riego que se encuentran en la parte superior de la imagen.

un sensor pasivo que captura ondas con longitudes entre $0,45 \mu\text{m}$ y $2,35 \mu\text{m}$ distribuidos en seis bandas multiespectrales, dos bandas termales y una banda pancromática. Cabe destacar que la resolución espacial varía para los distintos tipos de bandas. Mientras las seis bandas multiespectrales tienen una resolución de $30 \text{ m} \times 30 \text{ m}$, las bandas termales trabajan con píxeles de $60 \text{ m} \times 60 \text{ m}$. La banda pancromática obviamente tiene la mejor resolución espacial, precisamente $15 \text{ m} \times 15 \text{ m}$. En este esquema, es conveniente trabajar solo con las seis bandas multiespectrales para evitar problemas de resolución. Para mayor información acerca de las bandas del ETM+ refiérase a la figura 6.1.

La imagen de interés con el número de identificación 228 – 082 muestra la provincia de Córdoba, Argentina. De esta imagen se corta una zona de $1,2 \text{ km} \times 1,2 \text{ km}$ que se encuentra aproximadamente dos kilómetros al norte de la localidad de Arroyito. En la figura 7.1 se muestra la imagen usada en colores rojo-verde-azul usando las bandas 3, 2 y 1. Además, se presentan los niveles de gris de la banda 7 de la misma imagen donde se pueden distinguir algunos lotes que en las bandas 1, 2 y 3 parecen ser un solo lote grande.

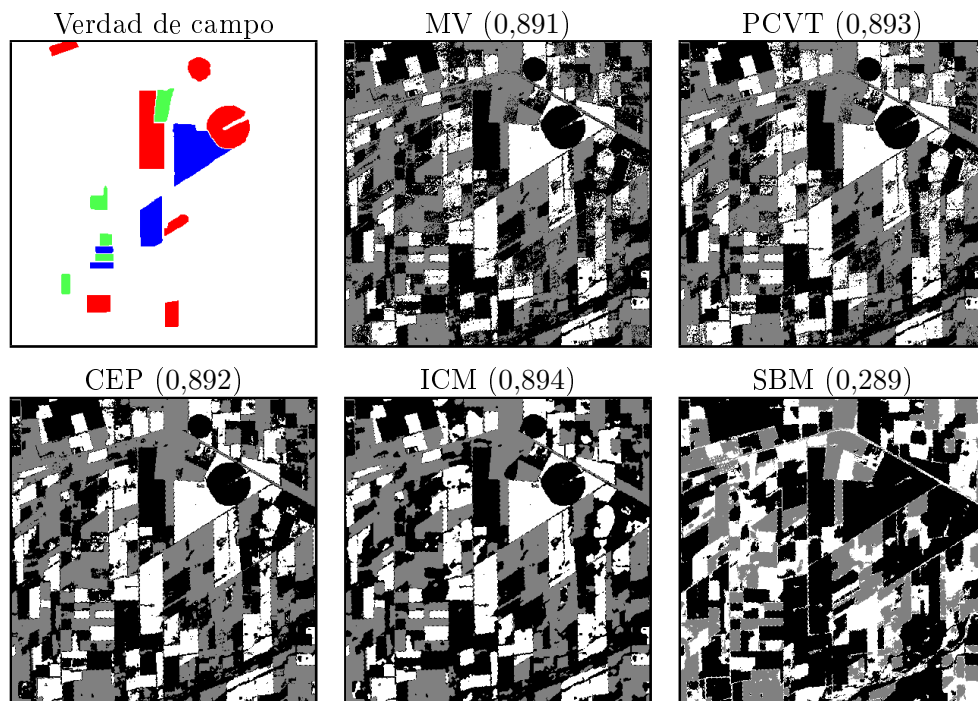


FIGURA 7.2: Segmentaciones de la imagen Landsat 7 con seis bandas multiespectrales (coeficientes κ en paréntesis). La verdad de campo está disponible solo en las partes marcadas. Se observa que SBM no logra distinguir las clases rojo y azul. Como consecuencia tiene un coeficiente κ bajo.

En la figura 7.2 se introduce la verdad de campo y se presentan los resultados experimentales. En este caso MV, PCVT, CEP y ICM usan una distribución normal multivariante mientras SBM aplica distribuciones normales unidimensionales en cada banda. Se observa que SBM no logra distinguir dos de las cuatro clases y por ende tiene un coeficiente κ bajo.

Luego se corren los algoritmos con los datos de la banda 7 exclusivamente. En este experimento unidimensional se pueden aplicar distintas funciones de probabilidad para los algoritmos PCVT y CEP. En la figura 7.3 se presentan algunos resultados que muestran que MV, ICM y PCVT mezclan dos de las cuatro clases. Solo en el caso de aplicar una distribución que no sea la normal, PCVT logra distinguir correctamente las cuatro clases. En cambio, CEP y SBM discriminan las cuatro clases para cualquier función de probabilidad.

En la figura 7.4 se presenta un resumen de los resultados de esta sección. En la misma figura se muestran los tiempos de computación. Se observa que MV e ICM tienen un costo computacional mucho menor al de los algoritmos PCVT, CEP y en partes SBM.

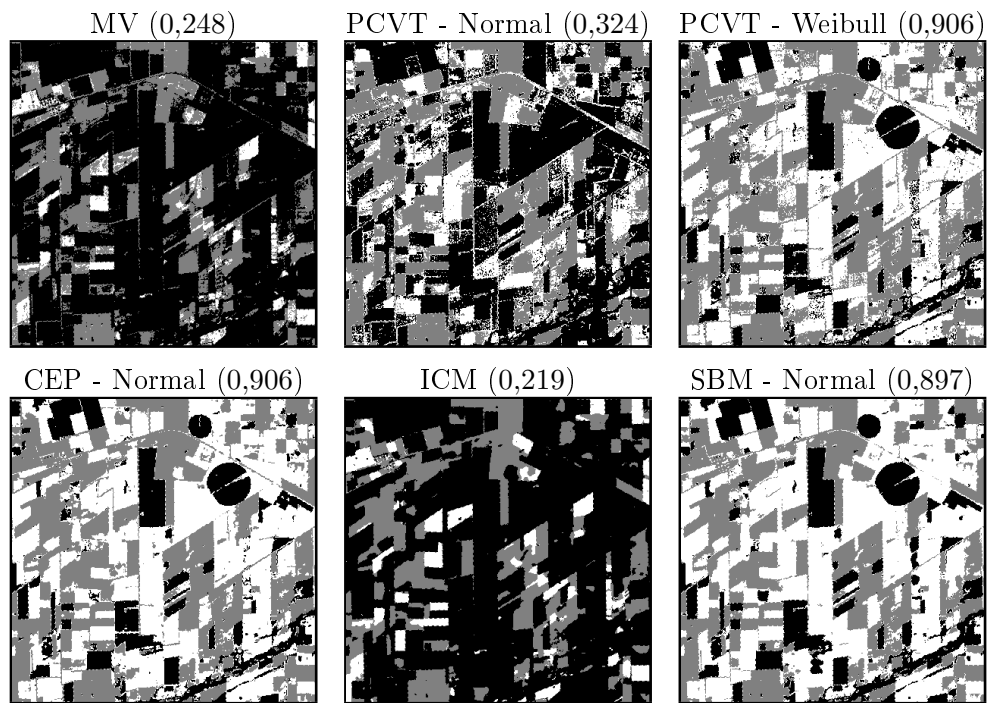


FIGURA 7.3: Segmentaciones de la banda 7 de la imagen Landsat 7. MV, ICM y PCVT juntos con una distribución normal no logran distinguir las cuatro clases mientras CEP, SBM y PCVT juntos con una distribución Weibull muestran un excelente desempeño.

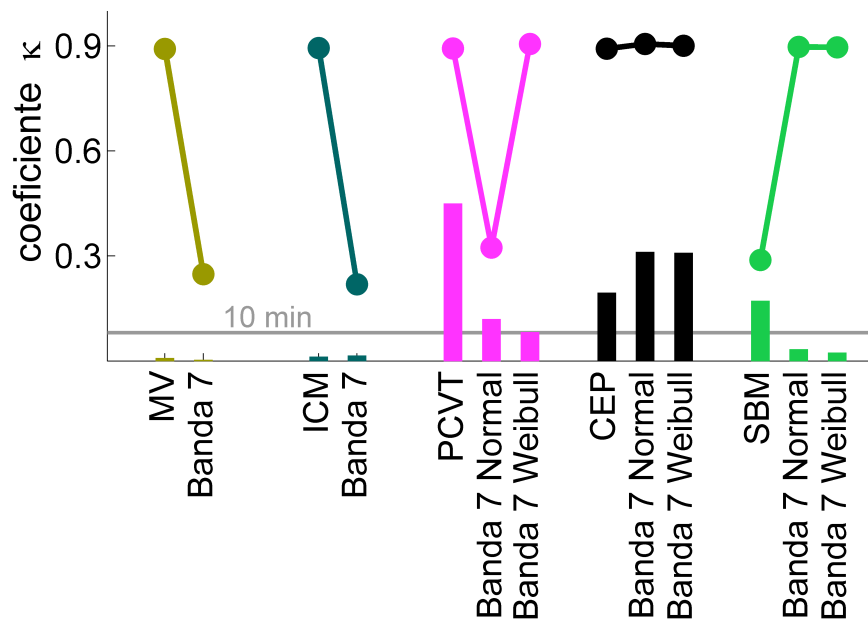


FIGURA 7.4: Evaluación de la imagen Landsat 7. Se observa que MV, ICM y PCVT (juntos con una distribución normal) empeoran fuertemente su desempeño al usar solo la banda 7 mientras SBM mejora su rendimiento. EL algoritmo CEP muestra el mejor desempeño general obteniendo altos valores de κ en cualquier configuración. Por otro lado, CEP (igual que PCVT y SBM en partes) tiene un alto coste computacional en este experimento.

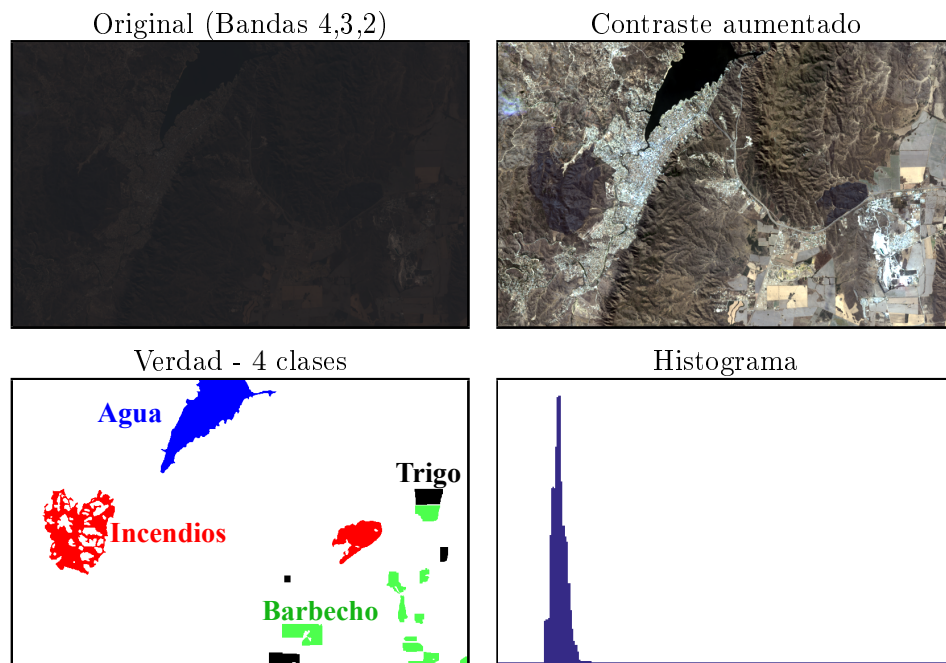


FIGURA 7.5: Configuración inicial del experimento Landsat 8. En la imagen con contraste aumentado se puede observar las dos partes quemadas por incendios forestales.

Landsat 8

Otro ejemplo para un sensor de teledetección pasiva es el *Operational Land Imager* (OLI) que está montado en el satélite Landsat 8. Igual que el satélite Landsat 7, Landsat 8 se encuentra en una órbita sincrónica al sol a 705 km de altura. Esto significa que el satélite pasa por un punto específico siempre al mismo tiempo solar local. Obviamente la ventaja de esta órbita es que, de esta forma, se evita pasar por una zona de interés durante la noche. Cabe destacar que el sensor OLI captura ondas con longitudes entre $0,43 \mu\text{m}$ y $1,38 \mu\text{m}$. Además del OLI, el satélite está equipado con el *Thermal Infrared Sensor* (TIRS) que registra ondas termales entre $10,60 \mu\text{m}$ y $12,51 \mu\text{m}$.

La imagen usada en este trabajo fue tomada el 7 de septiembre de 2013 a las 10 : 34 horas de la mañana. La imagen muestra la ciudad de Carlos Paz y sus alrededores. Cabe destacar que, en dos zonas de la imagen se registraban incendios forestales a la hora de la captura. Por ende, el objetivo principal de este experimento es distinguir cuatro clases: incendios forestales, plantaciones de maíz, barbecho y el agua del dique San Roque. En la figura 7.5 se muestra la imagen original y la verdad de campo.

En este experimento se pone énfasis en el siguiente problema. A priori no se sabe con cuantas clases hay que correr los algoritmos para distinguir las cuatro clases de la verdad de campo. El resultado de este estudio es que MV y los modelos ocultos de Markov (PCVT y CEP) logran diferenciar incendios, maíz, barbecho y agua con solo cuatro clases mientras los modelos de Potts (ICM y SBM) necesitan cinco clases para cumplir con el objetivo. Por otro lado, vale mencionar que MV, PCVT y CEP consiguen muy altos coeficientes de κ para cuatro clases, pero solo valores medios en caso de correr los algoritmos con más de cuatro clases.

En la figura 7.6 se muestran las segmentaciones para cuatro y cinco clases. Se observa que ICM y SBM confunden los incendios con agua cuando se corre los algoritmos con tan solo cuatro clases. Luego de cinco clases en adelante todos los algoritmos distinguen bien las cuatro clases de la verdad de campo.

En la figura 7.7 se presenta la evolución de κ al aumentar la cantidad de clases. En caso de ICM se muestran los máximos y mínimos coeficientes de κ al aplicar vecindades de distintos tamaños. En cambio, para SBM se muestran los máximos y mínimos coeficientes de κ al aplicar distintas funciones de probabilidad como *Gamma*, *inverse Gaussian*, *Logistic*, *Nakagami* y *Weibull*.

Aviris

El último ejemplo de una imagen de teledetección pasiva es una imagen aérea. Se trata de una imagen capturada por el *Airborne Visible/Infrared Imaging Spectrometer* (AVIRIS) que vuela en uno de estos cuatro aviones:

1. ER-2 jet
2. Twin Otter International's turboprop
3. Scaled Composites' Proteus
4. WB-57

El rango de longitudes de onda cubierto por el sensor AVIRIS se ubica entre $0,37 \mu\text{m}$ y $2,50 \mu\text{m}$. Note que, este rango de longitudes es parecido al de Landsat 8. Sin embargo,

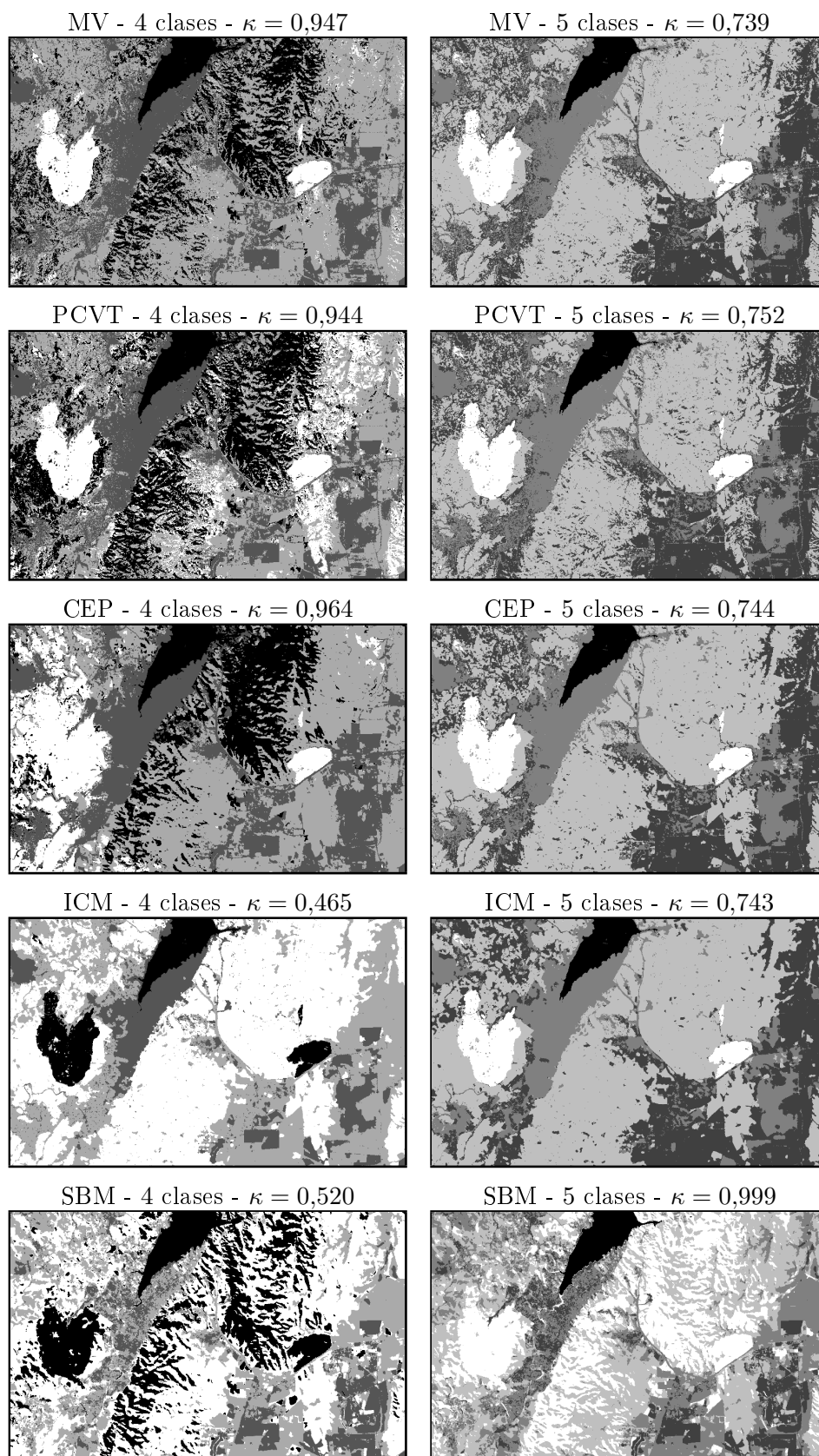


FIGURA 7.6: Comparación entre cuatro y cinco clases. Note que ICM y SBM no logran distinguir agua y la parte de incendios para cuatro clases. Por otro lado, MV, PCVT y CEP no consiguen un alto valor de κ para cinco clases.

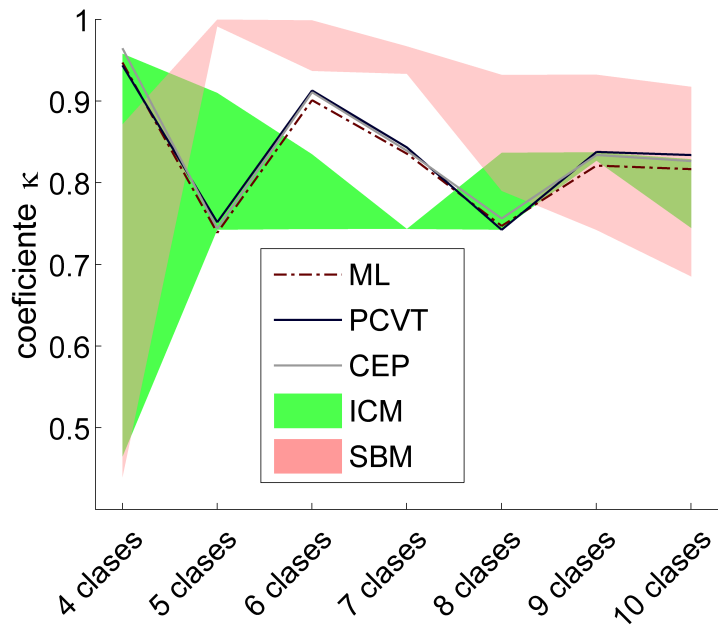


FIGURA 7.7: Evaluación del experimento Landsat 8. Se muestra cómo cambia el coeficiente κ al aumentar la cantidad de clases. Para ICM se presentan los valores máximos y mínimos de κ para las vecindades 3×3 , 5×5 y 7×7 . Al mismo tiempo se presentan los κ máximos y mínimos de SBM para distintas funciones de densidad.

hay una importante diferencia con respecto a las imágenes Landsat. En vez de registrar solo unas ocho bandas, AVIRIS captura 224 bandas simultáneamente.

La gran cantidad de bandas enfrenta a los algoritmos a una prueba difícil, sobre todo SBM cuyo coste computacional depende fuertemente de esta característica. Además, es importante mencionar que algunas de las bandas pueden tener valores negativos. Esto ocurre en la corrección atmosférica cuando los valores de infrarrojo son más pequeños que los valores de reflectancia en el rojo. Como consecuencia, es imposible correr el algoritmo SBM con funciones de densidad que no están definidas para valores negativos. Ejemplos para esta clase de distribuciones son *Gamma*, *inverse Gaussian* y *Weibull*.

La imagen estudiada tiene el número de identificación *f140528t01p00r10* y muestra el *Alameda Runway* cerca de San Francisco, EEUU. Las coordenadas exactas son $N37^{\circ}47'10''$, $W122^{\circ}19'19''$. La resolución espacial es de 16.40 m. En la figura 7.8 se usan las bandas 29, 20 y 12 para obtener una imagen de colores rojo-verde-azul. Aparte se muestra la verdad de campo y una versión de la imagen original con contraste aumentado.



FIGURA 7.8: Se usan las bandas 29, 20, 12 de la imagen AVIRIS para obtener una imagen rojo-verde-azul. Como el contraste de la imagen original es muy malo se presenta la misma imagen con contraste aumentado y la verdad de campo.

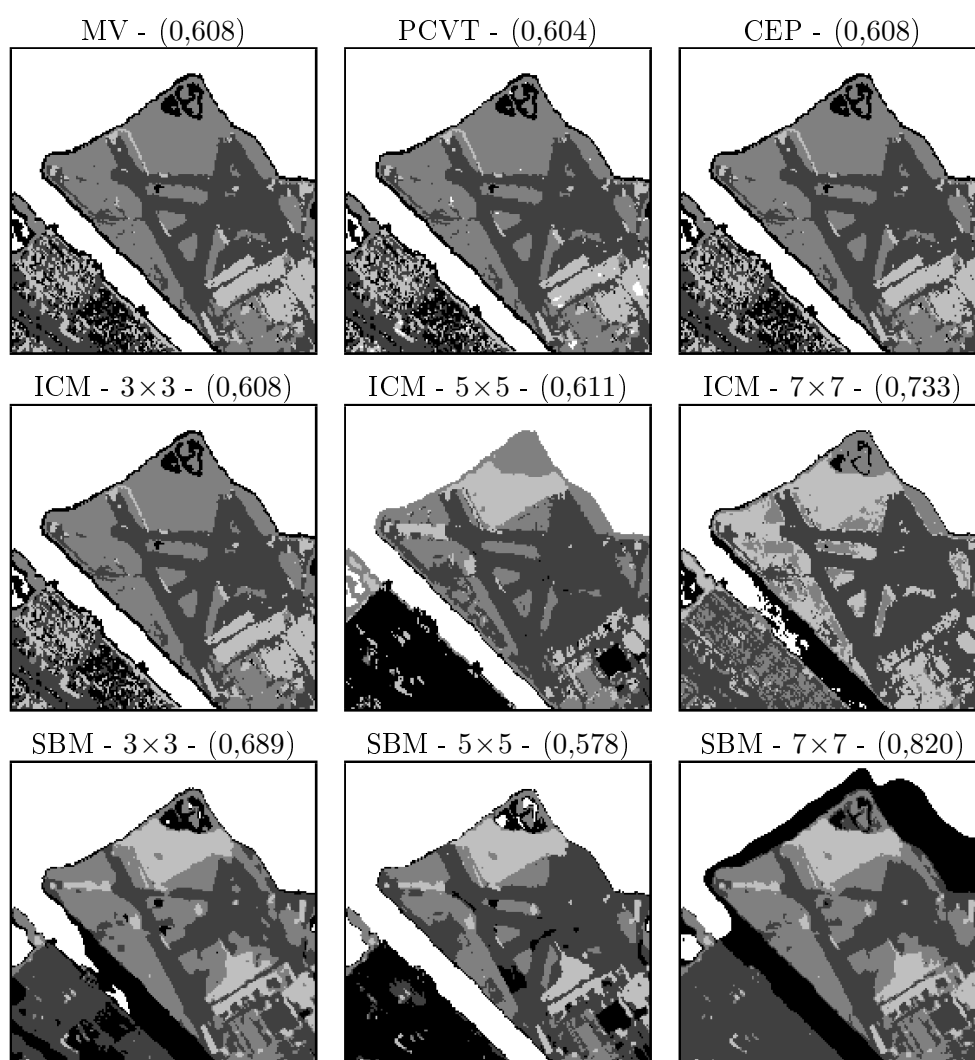


FIGURA 7.9: Segmentaciones de la imagen AVIRIS. Note que MV, PCVT, CEP y ICM con vecindad 3×3 no logran distinguir la parte de arena que se encuentra al norte de la pista de aterizaje.

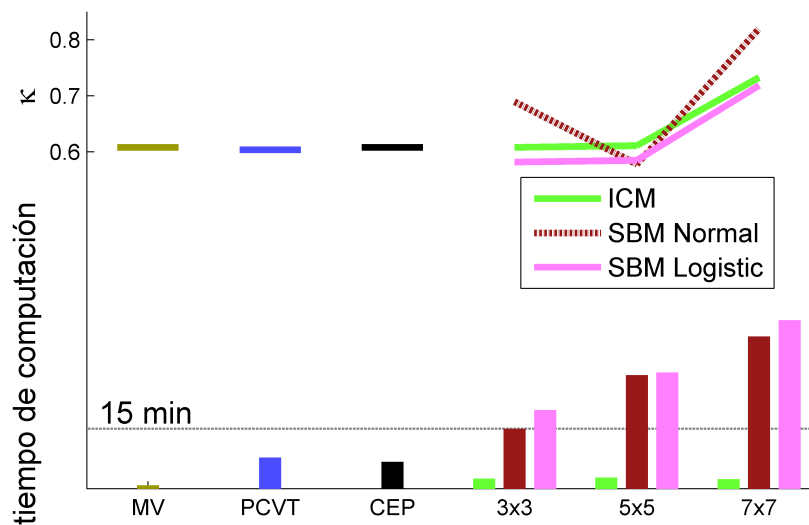


FIGURA 7.10: Evaluación de la imagen AVIRIS. Se puede observar que los algoritmos MV, PCVT y CEP tienen un desempeño parecido a los métodos ICM y SBM con vecindades chicas. Solo para vecindades grandes (7×7) ICM y SBM logran sacar una ventaja. Además, es destacable que el tiempo de computación de SBM es muy alto, especialmente para la distribución logística.

El objetivo en este experimento es distinguir agua poco profunda, agua profunda, arena y la pista de aterrizaje. En la figura 7.9 se muestran los resultados de los distintos algoritmos. Además, se evalúan los resultados y los tiempos de computación en la figura 7.10. En este contexto, se hace hincapié en la influencia de la vecindad usada por los modelos de Potts y en el efecto de correr SBM con una densidad que no sea la normal pero que permite valores negativos, como la distribución logística.

7.2.2. Imágenes de radar (teledetección activa)

En contrario a la teledetección pasiva, la teledetección activa no depende de la luz solar, ya que primero se emiten ondas y luego se registran las ondas reflejadas. Por lo tanto, se pueden tomar imágenes durante la noche o con cielo nublado.

En esta sección se usa una imagen de radar de apertura sintética (SAR) para evaluar las capacidades de los algoritmos para segmentar imágenes de teledetección activa. Precisamente, se trata de una imagen *COSMO-SkyMed* de banda *X* que trabaja con polarización dual (HH/VV donde H-horizontal y V-vertical) en modo *PingPong* con un nivel de procesamiento *L1A Single-look Complex Slant*. En otras palabras, se trata de una imagen con poco pre-procesamiento (el máximo nivel de procesamiento es *L1D*) con mucho ruido.

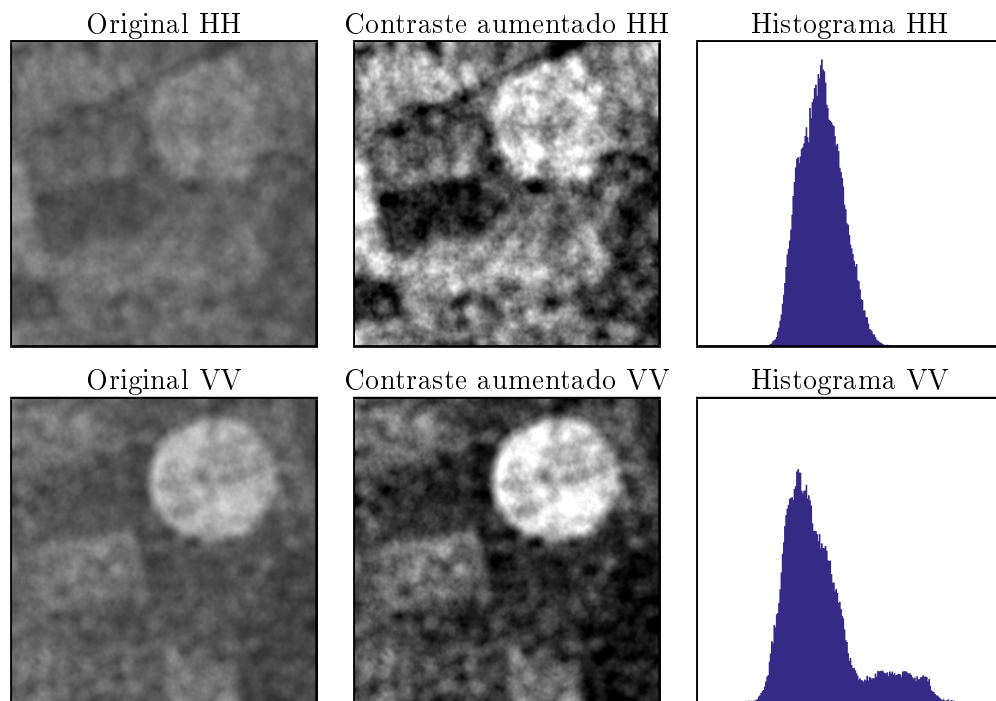


FIGURA 7.11: Polarizaciones horizontal-horizontal (HH) y vertical-vertical (VV) de una imagen *COSMO SkyMed*. Comparando los histogramas se observa que la polarización VV distingue mejor los tipos de suelo.

Cabe destacar que este tipo de imagen es frecuentemente usada para distinguir distintos tipos de vegetación [126, 127].

El satélite *COSMO-SkyMed* vuela en una órbita sincrónica al sol a 620 km de la tierra. Desde esta altura manda ondas con frecuencias entre 7.25 GHz y 7.75 GHz a la tierra y recibe ondas con frecuencias entre 7.9 GHz y 8.4 GHz.

La imagen en cuestión tiene el número de identificación:

CSKS1_DGM_B_PP_0A_CO_RD_SF_20140223213602_20140223213608.

Esta imagen fue tomada el 23 de febrero del 2014. Pero en vez de usar la imagen completa, se seleccionó una parte que muestra un pivote de riego que se encuentra aproximadamente 3 km al sur de la localidad de Río Segundo de la provincia de Córdoba. La resolución espacial de la imagen es 15 m.

En la figura 7.11 se muestran las dos polarizaciones HH y VV junto con sus respectivos histogramas. Se observa que VV tiene un mayor contraste que HH.

Lamentablemente no hay una verdad de campo en este experimento. Por consiguiente, no se pueden calcular los coeficientes κ . Sin embargo, este experimento sirve para investigar

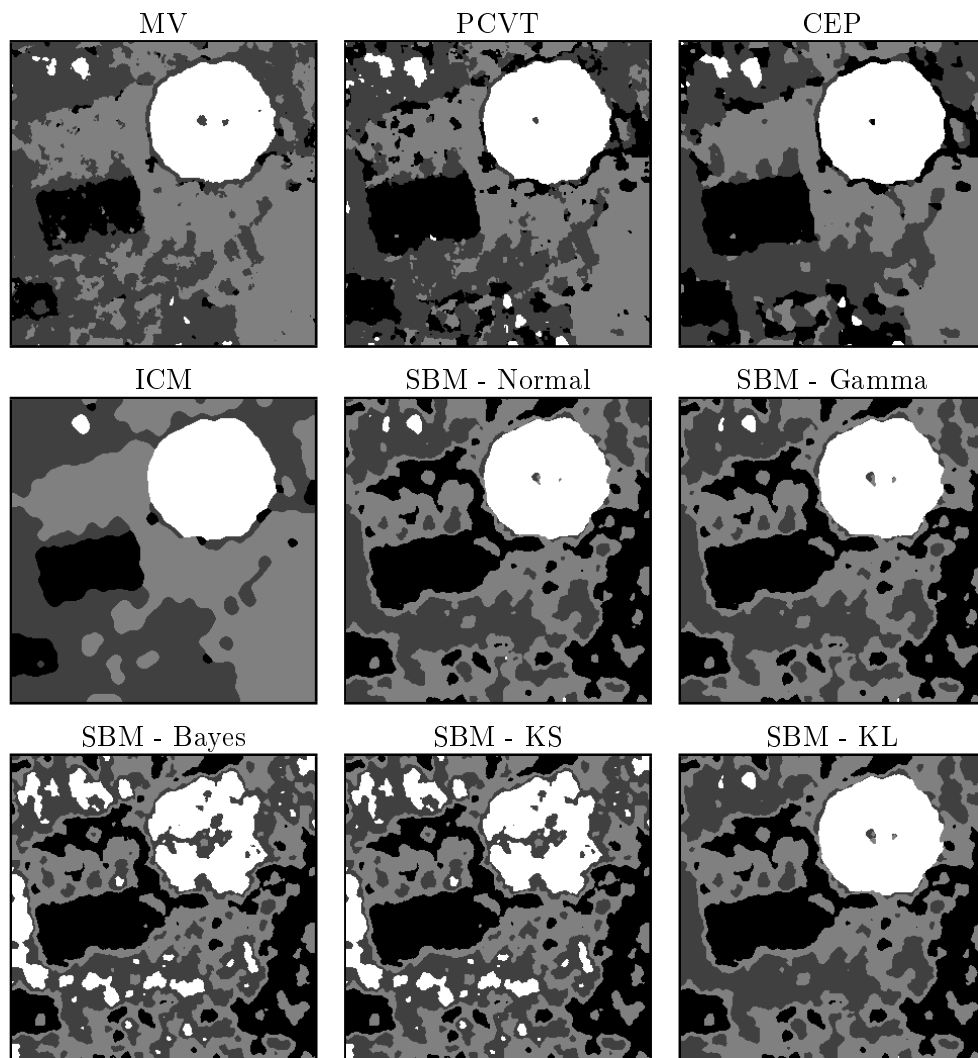


FIGURA 7.12: Segmentación de una imagen *COSMO SkyMed*. Se observa que ICM es el algoritmo que más suaviza la imagen mientras SBM (sobre todo en conjunto con los métodos Bayes y KS) conserva más detalles.

dos aspectos importantes de los algoritmos MV, PCVT, CEP, ICM y SBM. Primero, se compara la homogeneidad de los segmentos encontrados por los algoritmos. Por ejemplo, ICM suaviza mucho los datos observados. Esto significa que encuentra segmentos grandes que no son muy homogéneos. En cambio, SBM aplica una suavización mucho más leve. Por ende, se obtienen segmentos chicos, pero con mayor homogeneidad, como se ve en la figura 7.12.

El segundo punto que se estudia en esta sección es la pregunta: ¿Después de cuantas iteraciones convergen los algoritmos? Para investigar este asunto se corren todos los algoritmos veinte veces con condiciones iniciales aleatorias. En la figura 7.13 se presentan los resultados promedios de este experimento en escala logarítmica. Hacemos hincapié

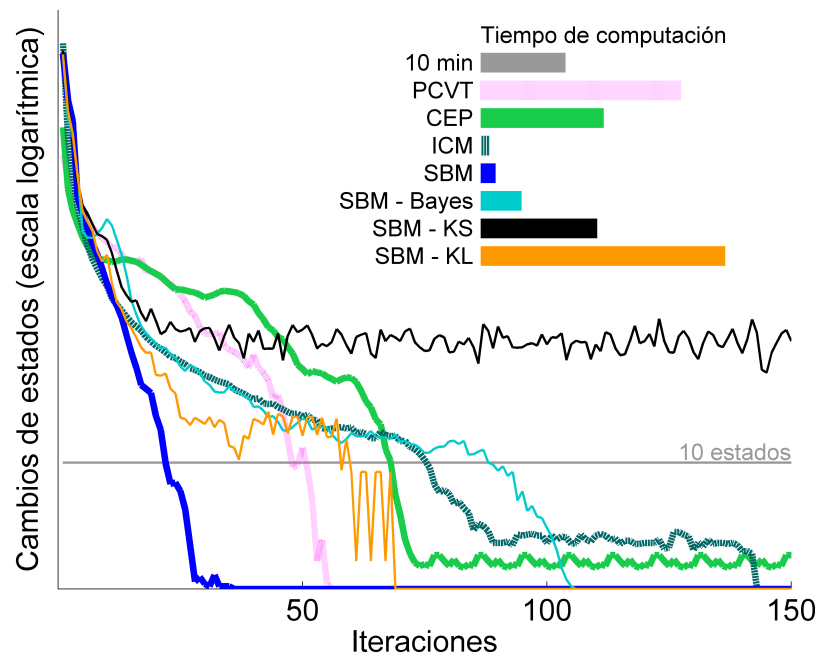


FIGURA 7.13: Promedio de cantidad de iteraciones necesarias para que los algoritmos PCVT, CEP, ICM, SBM, SBM-Bayes, SBM-KS y SBM-KL converjan. Cabe destacar el alto costo computacional que implican los métodos Bayes, KS y sobre todo KL. Además, es notable que solo SBM-KS y CEP no convergieron en algunas de las 20 corridas.

que una escala lineal no permitiría interpretar fácilmente la figura 7.13, ya que en las primeras iteraciones se cambian los estados de muchos píxeles.

7.3. Imágenes de referencia

En esta sección se estudian imágenes de referencia para evaluar los algoritmos propuestos fuera del área de la teledetección. En este contexto, se trata de cubrir los siguientes puntos:

1. Investigar las capacidades de generalización de los algoritmos
2. Evaluar los métodos de determinar la función de probabilidad “online”
3. Determinar la sensibilidad de los algoritmos con respecto al ruido gaussiano
4. Estudiar la influencia de la función de emisión

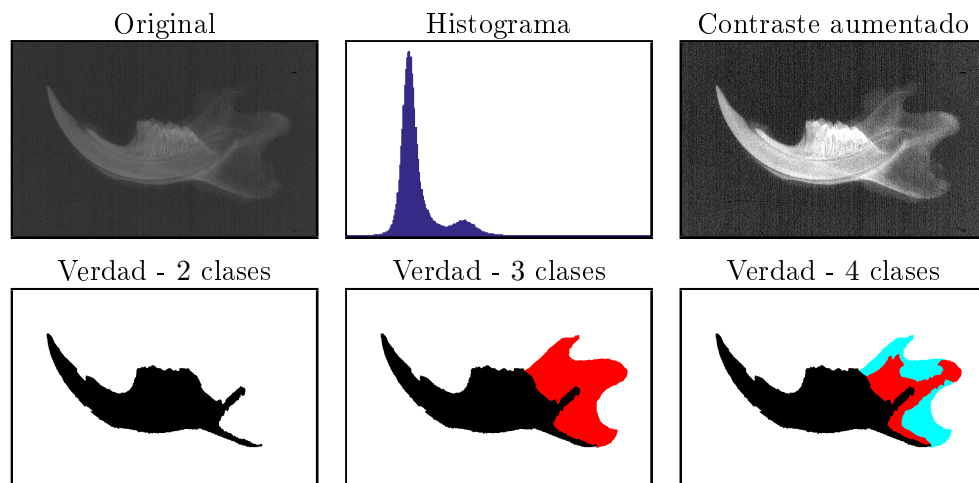


FIGURA 7.14: Tomografía de rayos X que muestra el diente de una rata junto con su histograma. La imagen consiste en cuatro clases: 1) huesos y dientes, 2) tejido y carne, 3) cartílago y 4) fondo. Estas cuatro clases se pueden agrupar para obtener dos o tres clases.

Para lograr estos objetivos, se utilizan cuatro imágenes de distintos ámbitos: Una imagen de rayos X, una imagen artificial, 200 imágenes simuladas y una imagen estándar de procesamiento de imágenes.

7.3.1. Imagen rayo X de una rata

La primera imagen de referencia es una tomografía de rayos X que muestra el diente de una rata *Winstar*. Esta imagen forma parte de un estudio de crecimiento de ratas bajo distintas condiciones que se realizó en el departamento de odontología de la Universidad Nacional de Córdoba [128]. En el caso particular de este trabajo se trata del diente de una rata que – junto con su madre – fue expuesto a la luz 24 horas por día.

Para evaluar el crecimiento de la rata es esencial separar las partes duras (hueso y diente) de las partes blandas como el tejido, la carne y el cartílago. Cabe destacar que, la tomografía tiene un alto nivel de ruido, en parte por las dimensiones chicas del diente y en parte por la digitalización mediante un escáner. En la figura 7.14 se muestra la imagen original, un histograma y la verdad de campo.

En este experimento se estudian tres aspectos importantes de los modelos markovianos. Primero, el desempeño de PCVT, CEP y SBM al aplicar la misma función de probabilidad. Segundo, la influencia del tamaño de la vecindad en los modelos de Potts. Y, por

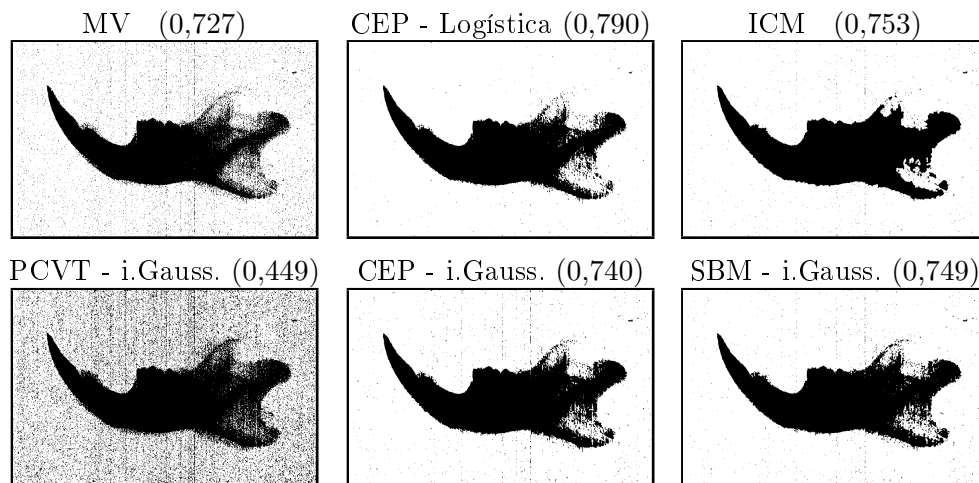


FIGURA 7.15: Imagen de rayos X con dos clases (coeficientes κ en paréntesis). En la primera fila se muestran los algoritmos de referencias (MV y ICM) y el mejor resultado de todos que fue obtenido corriendo CEP con una distribución logística. En la segunda fila se presentan los algoritmos PCVT, CEP y SBM con la inversa gaussiana como densidad. Se observa que la misma distribución puede llevar a resultados muy diferentes para distintos algoritmos.

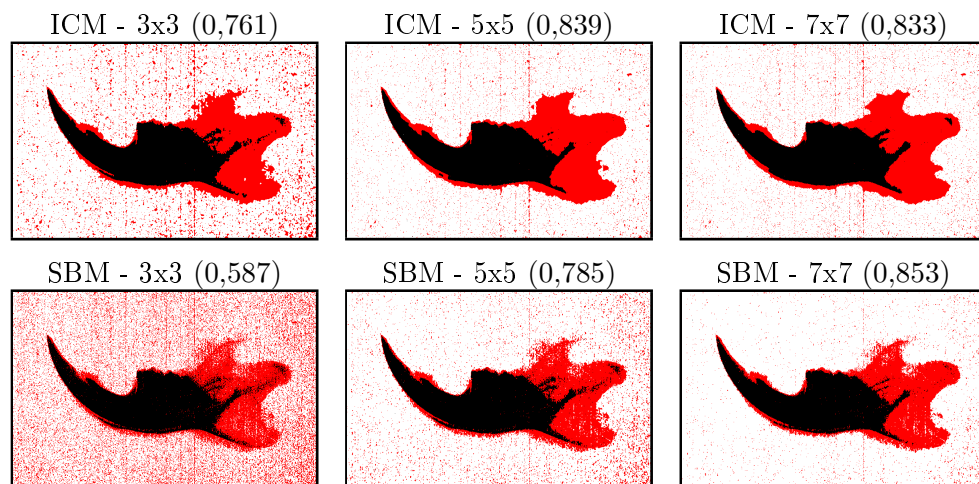


FIGURA 7.16: Imagen de rayos X con tres clases (valores de κ en paréntesis). Se compara la influencia de la vecindad para los métodos ICM y SBM. En este contexto 3x3 se refiere a la vecindad de 9 vecinos que es equivalente a la segunda vecindad de la figura 2.3.

último, la influencia del método de inicialización que puede ser el algoritmo EM o el uso de datos de entrenamiento.

En la figura 7.15 se presentan los resultados para distintas funciones de emisión. Luego, se comparan ICM y SBM para distintas vecindades en la figura 7.16. Después, se muestran los resultados de PCVT, CEP e ICM para una inicialización con EM y datos de entrenamiento en la figura 7.17.

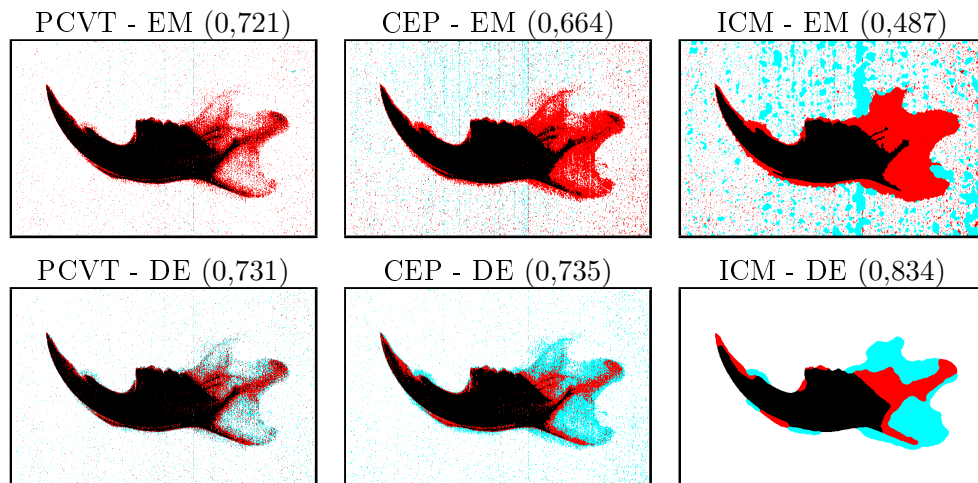


FIGURA 7.17: Imagen de rayos X con cuatro clases (coeficientes κ en paréntesis). Se muestran los resultados finales partiendo de distintos métodos de iniciación. Por un lado, se corre EM y por otro lado se usan datos de entrenamiento (DE). Se observa que los tres algoritmos son sensibles con respecto al método de inicialización. Todos los algoritmos de esta figura utilizan la densidad normal.

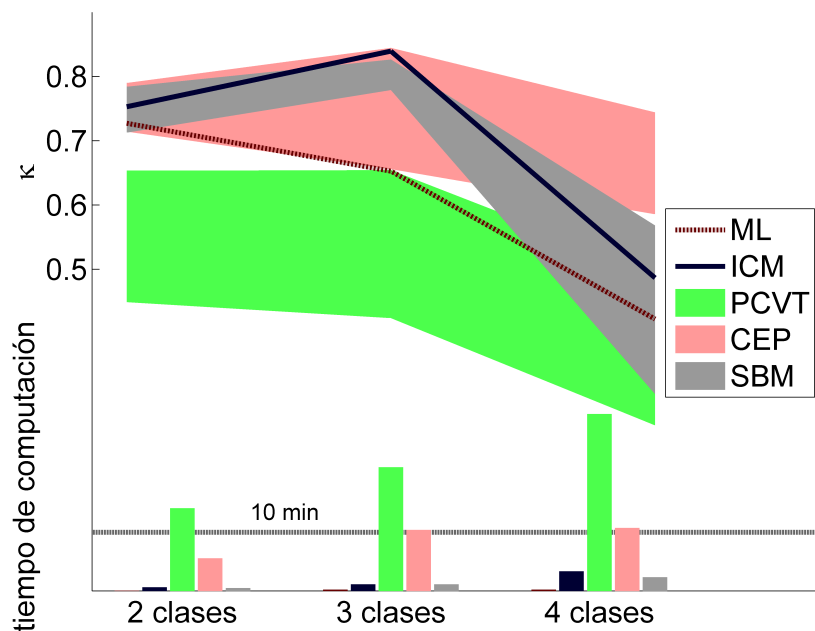


FIGURA 7.18: Evaluación de la imagen rayos X. En caso de PCVT, CEP y SBM se muestran los valores máximos y mínimos de κ para las distintas densidades (*Normal*, *Gamma*, *inverse Gaussian* y *Weibull*). Además, se comparan los tiempos computacionales. Como la imagen tiene una sola banda de niveles de gris, SBM tiene un tiempo computacional muy chico (comparable con ICM) mientras PCVT requiere un tiempo de computación largo.

Al final, se evalúan todos los resultados de la tomografía en la figura 7.18. En la misma figura se presentan los tiempos computacionales de los distintos algoritmos, siendo PCVT el algoritmo con coste computacional más alto. La razón de esto es que el orden de SBM depende principalmente de la cantidad de bandas (en este caso hay una sola banda) mientras el orden CEP es la cantidad de clases al cubo (solo hay cuatro clases en este experimento).

7.3.2. Logo de la UTN

Para investigar la sensibilidad de los algoritmos con respecto al ruido se lleva a cabo un experimento que involucra el logotipo de la Universidad Tecnológica Nacional (UTN). Dicho logotipo es una figura formada por líneas rectas y curvas sobre un fondo blanco. Por lo tanto, la imagen contiene solo dos clases: logotipo y fondo.

La configuración de este experimento es muy distinta a la de los otros experimentos. La idea principal es tomar el logotipo con fondo como verdad de campo y luego observar cada una de las dos clases con una distribución normal. Entonces, el experimento tiene cuatro parámetros: las medias de las dos clases y las desviaciones estándar de las dos clases.

Si fijamos la media de una clase en 100 y además definimos que las dos desviaciones estándar sean iguales y fijas ($\sigma = 25$) nos quedamos con un solo parámetro: la media de la segunda clase. Si la media de la segunda clase es 100,5, obtenemos una imagen donde el logotipo y el fondo prácticamente no se distinguen porque sus funciones de probabilidad son muy parecidas. En cambio, si la media de la segunda clase es 120, se va a notar mucho la diferencia entre logotipo y fondo.

Entonces, vamos a aumentar la media de la segunda clase sucesivamente de 100,5 a 120 con un paso de 0,5. En cada escala se corren todos los algoritmos y se calculan los coeficientes de κ . De esta forma, se obtienen curvas de κ que muestran el desempeño de cada algoritmo en función de la complejidad de la tarea.

Para evitar problemas estadísticos se observan 10 imágenes distintas en cada escala y luego se calculan los valores κ como promedios de las diez recorridas. En la figura 7.19 se muestra la verdad de campo y los resultados cuando la segunda media vale 109,5. Luego, se presenta la evolución de κ para los distintos algoritmos en la figura 7.20.

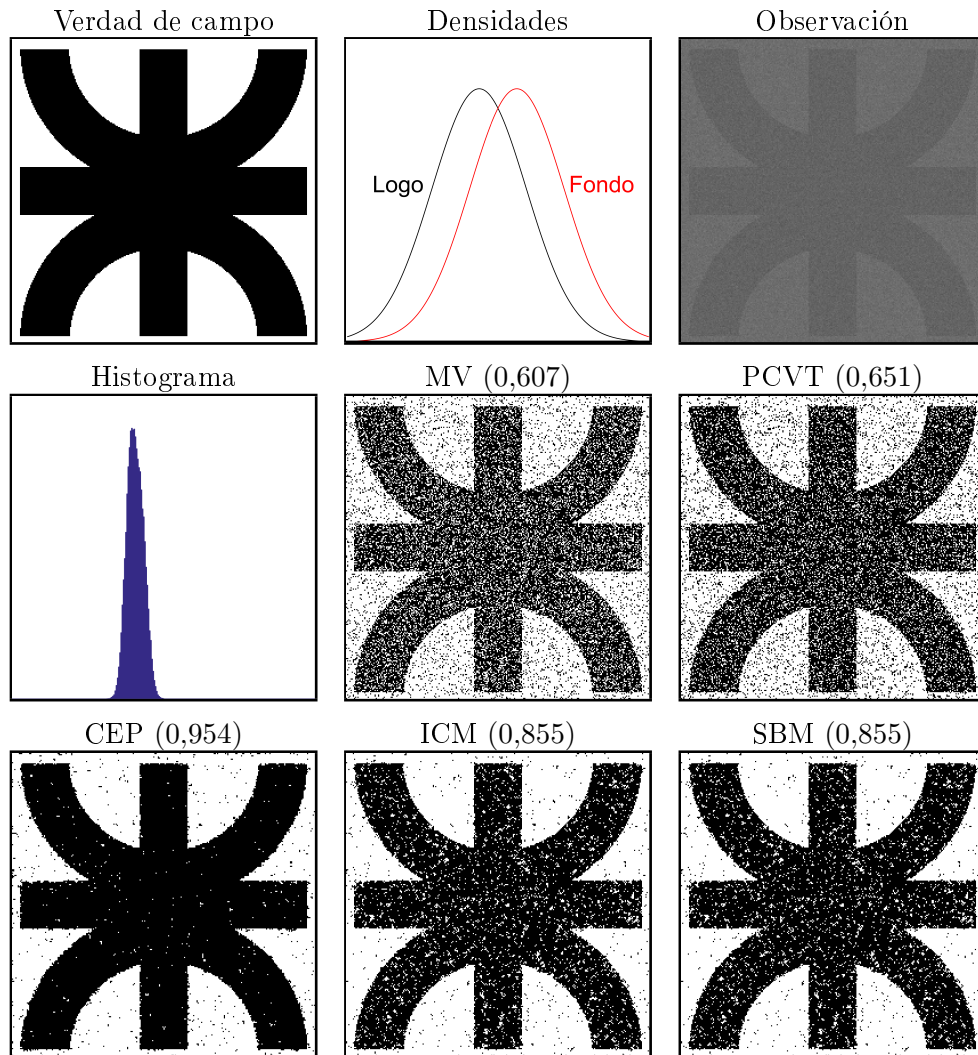


FIGURA 7.19: Se muestra el logotipo con fondo de la UTN que se toma como verdad de campo. Además se presentan los resultados en el caso $logotipo \sim N(100, 25)$ y $fondo \sim N(109, 5, 25)$.

7.3.3. Imágenes simuladas

En este experimento se aplica un procedimiento parecido al experimento de la UTN. Partiendo de una verdad de campo artificial, se simulan 200 imágenes de acuerdo al algoritmo 7 usando como probabilidades *Gamma*, *inverse Gaussian*, *Nakagami* y *Weibull*.

Luego se corren los algoritmos PCVT, CEP y SBM junto con los métodos de estimar la distribución óptima “online” de la sección 3.6. Al final, se comparan los resultados de los métodos *Bayes*, *KS* y *KL* con los algoritmos de referencia MV e ICM.

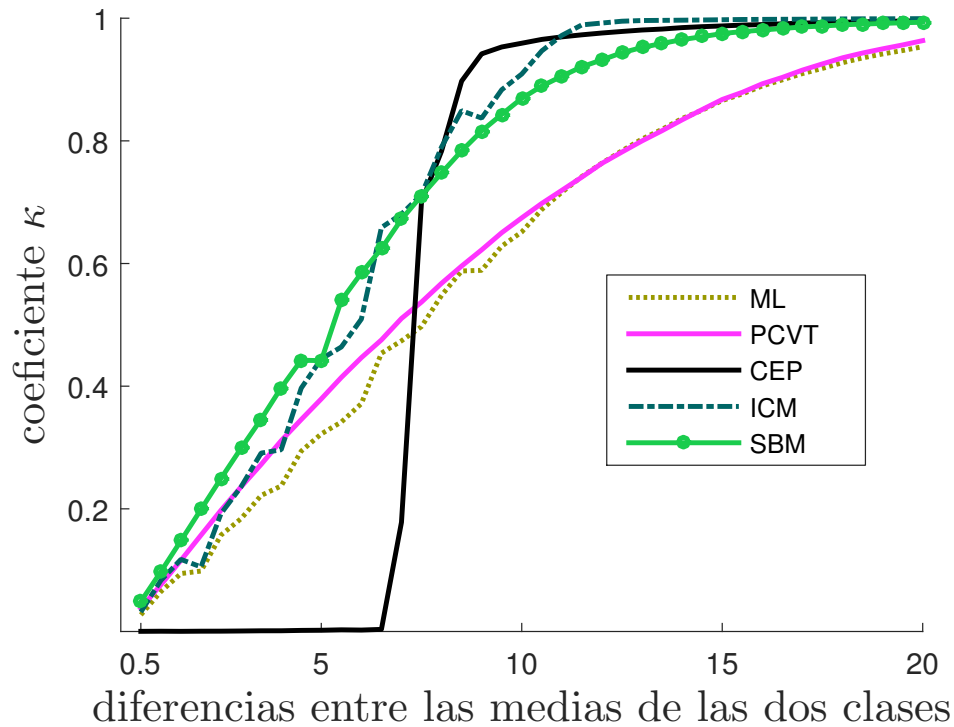


FIGURA 7.20: Evaluación del experimento UTN. Se observa que SBM tiene los valores κ más altos cuando las medias de las dos clases son muy parecidas. En cambio, para el algoritmo CEP hay un punto clave (alrededor de una diferencia de 7 entre las dos clases) a partir del cual logra separar las dos clases. Los resultados de PCVT son muy parecidos a MV.

Algoritmo 7: Imágenes simuladas

1. Definir la verdad de campo
 2. Sortear el orden de las clases
 3. Sortear una distribución para cada clase
 4. Fijar los parámetros de cada distribución al azar pero dentro de ciertos límites
 5. Observar la verdad de campo de acuerdo a las funciones de probabilidad
 6. Repetir los pasos 2), 3), 4) y 5) 200 veces en total
-

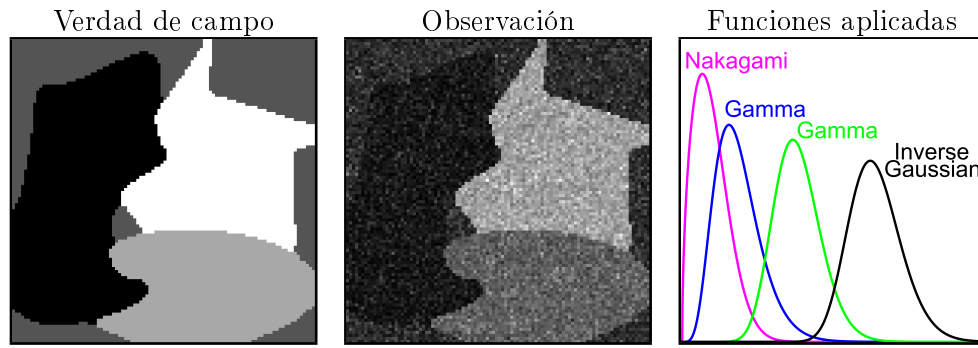


FIGURA 7.21: Ejemplo de una imagen simulada. Se muestra la verdad de campo, una observación de la verdad de campo y las correspondientes funciones de probabilidad.

CUADRO 7.2: Evaluación de los resultados de las imágenes simuladas. Por un lado, se muestran los promedios de κ para los algoritmos PCVT, CEP y SBM. Por otro lado, se comparan las probabilidades de encontrar las funciones de emisión usando los métodos Bayes, *KS* y *KL*.

	κ_{Normal}	κ_{Bayes}	κ_{KS}	κ_{KL}	Bayes	KS	KL
PCVT	0,689	0,684	0,741	0,813	61,8 %	60,6 %	55,9 %
CEP	0,742	0,689	0,744	0,810	54,9 %	55,1 %	54,5 %
SBM	0,810	0,694	0,690	0,820	64,0 %	64,8 %	64,6 %

En la figura 7.21 se muestra la verdad de campo junto con una imagen que se simuló con las distribuciones *Nakagami*, *Gamma*, *Gamma*, *inverse Gaussian*. Después se presentan los resultados de todos los algoritmos en combinación con los métodos *Bayes*, *KS* y *KL* en la figura 7.22.

Para evaluar los resultados de las 200 imágenes simuladas se usa la tabla 7.3.3. En esta tabla se muestran los promedios de κ y las probabilidades con las cuales los métodos *Bayes*, *KS* y *KL* identificaron bien las funciones de emisión. Se observa que SBM en conjunto con *KL* es el algoritmo que obtiene los valores de κ más altos. En la figura 7.23 se presentan los resultados de la tabla 7.3.3 de forma gráfica agregando los valores de κ después de la primera iteración.

7.3.4. Imagen estándar

La última imagen que se usa para comparar MV, PCVT, CEP, ICM y SBM es una imagen estándar de procesamiento de imágenes. La imagen denominado “Cameraman” forma parte de muchas bases de datos de imágenes, como por ejemplo [129,130] y muchos programas de procesamiento de imágenes como Matlab[®].

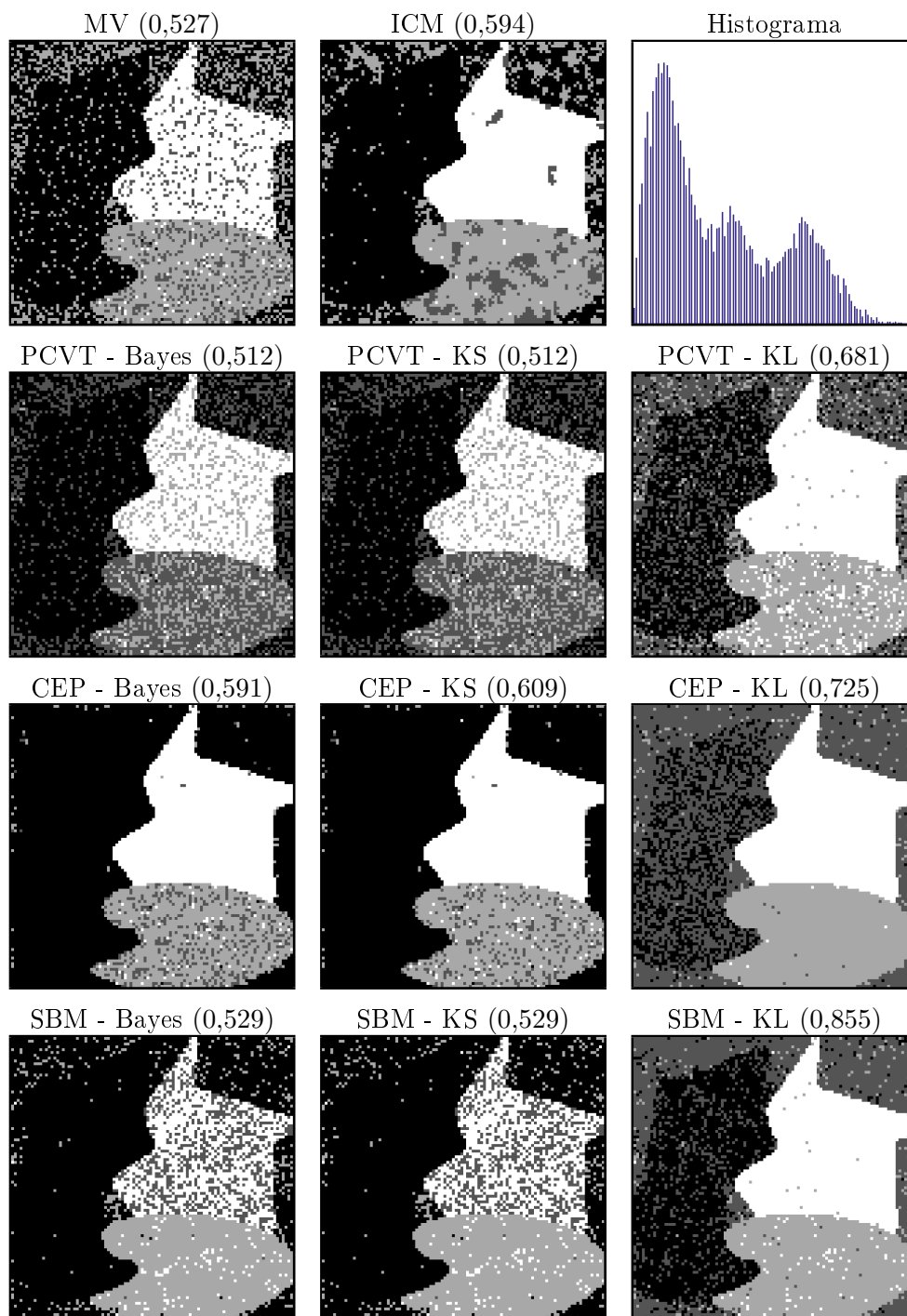


FIGURA 7.22: Segmentaciones de una imagen simulada. Se presentan los resultados de los distintos algoritmos juntos con su coeficiente de κ y el histograma de la observación. Note que el método *KL* consigue los valores más altos de κ .

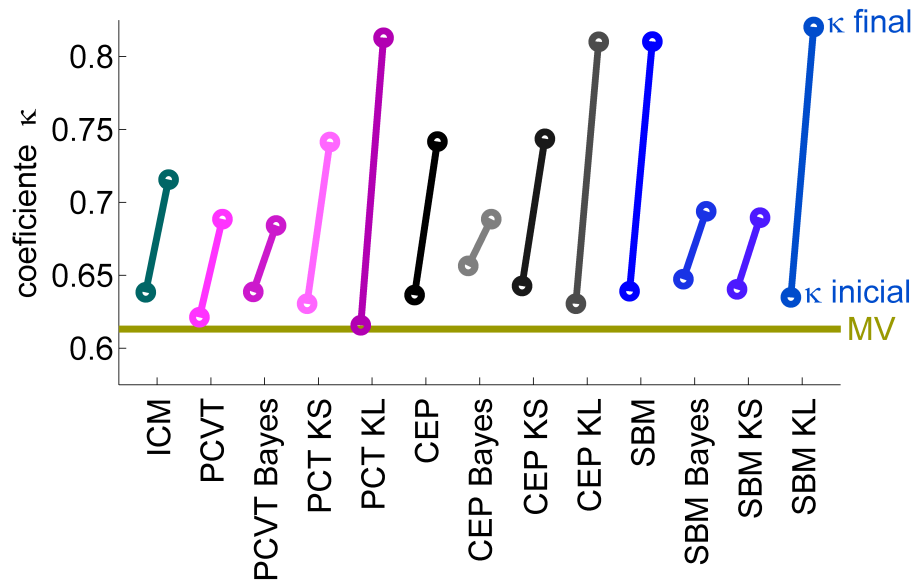


FIGURA 7.23: Evolución del coeficiente κ desde el mapa inicial (MV), luego la primera iteración (κ inicial) hasta convergencia (κ final) para las imágenes simuladas.

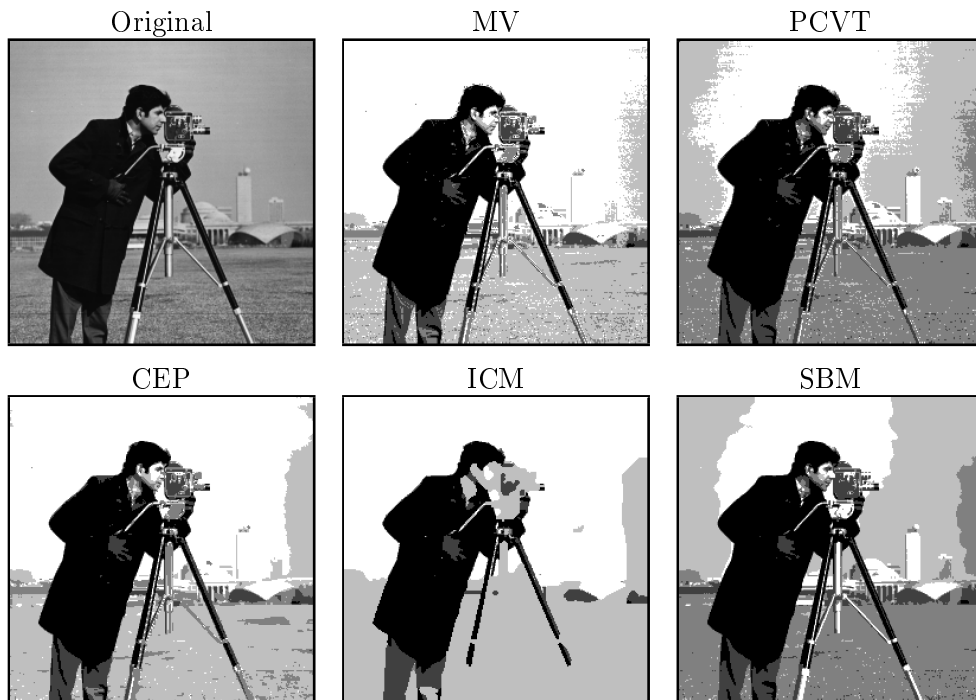


FIGURA 7.24: Segmentaciones de la imagen “Cameraman”. Se observa que ICM suaviza mucho la imagen mientras CEP y SBM logran buenos resultados.

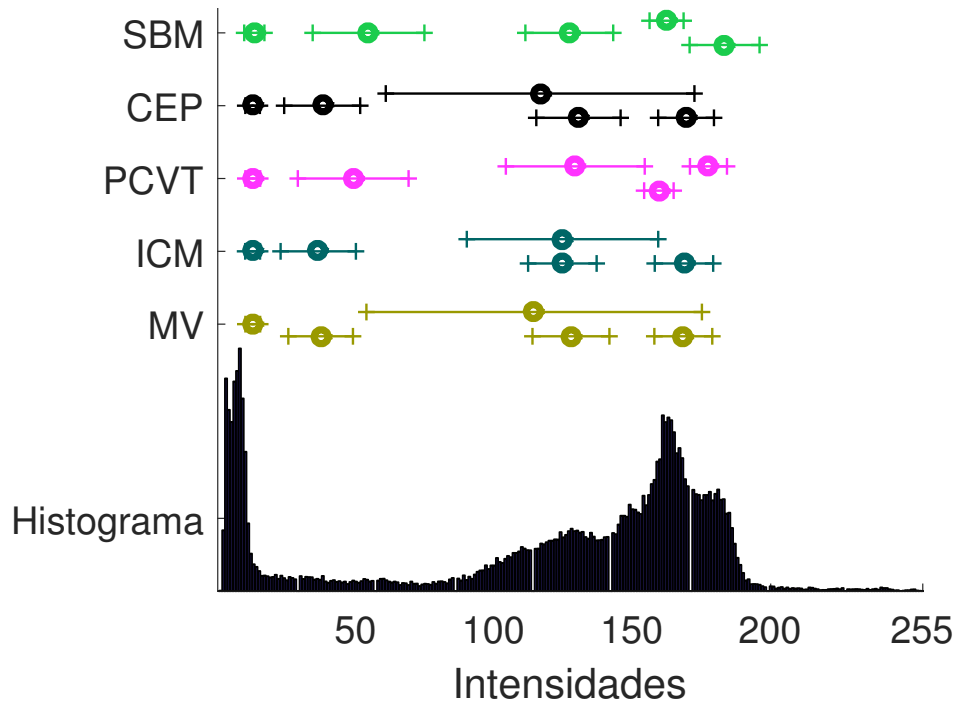


FIGURA 7.25: Arriba se muestran las medias y desviaciones estándar de los algoritmos MV, ICM, PCVT, CEP y SBM para la imagen “Cameraman”. Abajo está el histograma de la imagen.

La idea de este experimento es investigar las capacidades de los algoritmos de suavizar ciertas áreas de la imagen (como el pasto o el cielo) sin emborronar los detalles finos de la cara del hombre. En la figura 7.24 se presentan la imagen original y los resultados de los algoritmos. Se observa que ICM suaviza demasiado mientras PCVT no logra segmentar ni el pasto ni el cielo. Aparentemente CEP y SBM (e incluso MV) llevan a buenos resultados.

Como no hay una verdad de campo, no se pueden calcular los coeficientes de κ . Sin embargo, se pueden comparar los parámetros de las distintas clases. Para esto, se presenta el histograma y las correspondientes distribuciones de los cinco algoritmos en la figura 7.25.

7.4. Fusión de datos

En esta sección se evalúan los algoritmos de fusión de datos. Primero, se usa un escenario teórico con datos simulados. Después, se corren los algoritmos K-medias, MG, AJ y MDF para datos reales.

7.4.1. Datos simulados

Muchas veces es difícil obtener datos reales de distintos sensores para el mismo lote en el mismo instante de tiempo. Entonces, es una buena opción recurrir a datos simulados. Las ventajas de datos simulados son:

- Se adquieren datos sin costo
- Se puede generar un escenario con cualquier cantidad de sensores unidimensionales y multiespectrales
- Se conoce la verdad de campo
- Se conocen los parámetros de las funciones de probabilidad

La idea en esta sección es simular, por un lado, datos equidistantes y multiespectrales que representen los píxeles de una imagen satelital y por el otro, mediciones locales en sitios arbitrarios. En la figura 7.26 se presenta el escenario teórico de esta sección. Como verdad de campo se diseña un lote cuadrado con cuatro clases. Cabe destacar que cada una de las cuatro clases limita con las restantes tres clases. De esta forma, se verifica si los algoritmos son capaces de separar todas las clases.

En la figura 7.27 se muestran las bandas 1, 2 y 3 que representan los píxeles de una imagen satelital óptica, mientras las bandas 4 y 5 simulan los píxeles de una imagen de radar. Además, hay mediciones *in situ* de humedad de suelo (banda 6) y potencial de tallo (banda 7).

En la figura 7.28 se muestran los resultados de los algoritmos K-medias, MG, AJ y MDF. Además, se evalúan los resultados y los tiempos de computación en la figura 7.29.

7.4.2. Datos reales

En este experimento se evalúan los algoritmos de fusión de datos para observaciones reales. Las mediciones de humedad de suelo, fotosíntesis y potencial de tallo se llevaron a cabo en el sitio experimental de CONAE en Falda del Carmen, ruta C45 Km 8, donde se está cultivando trigo de 5 meses con una altura de aproximadamente 50 cm. Cabe destacar que un día después de las mediciones pasó el satélite Landsat 8 por la provincia

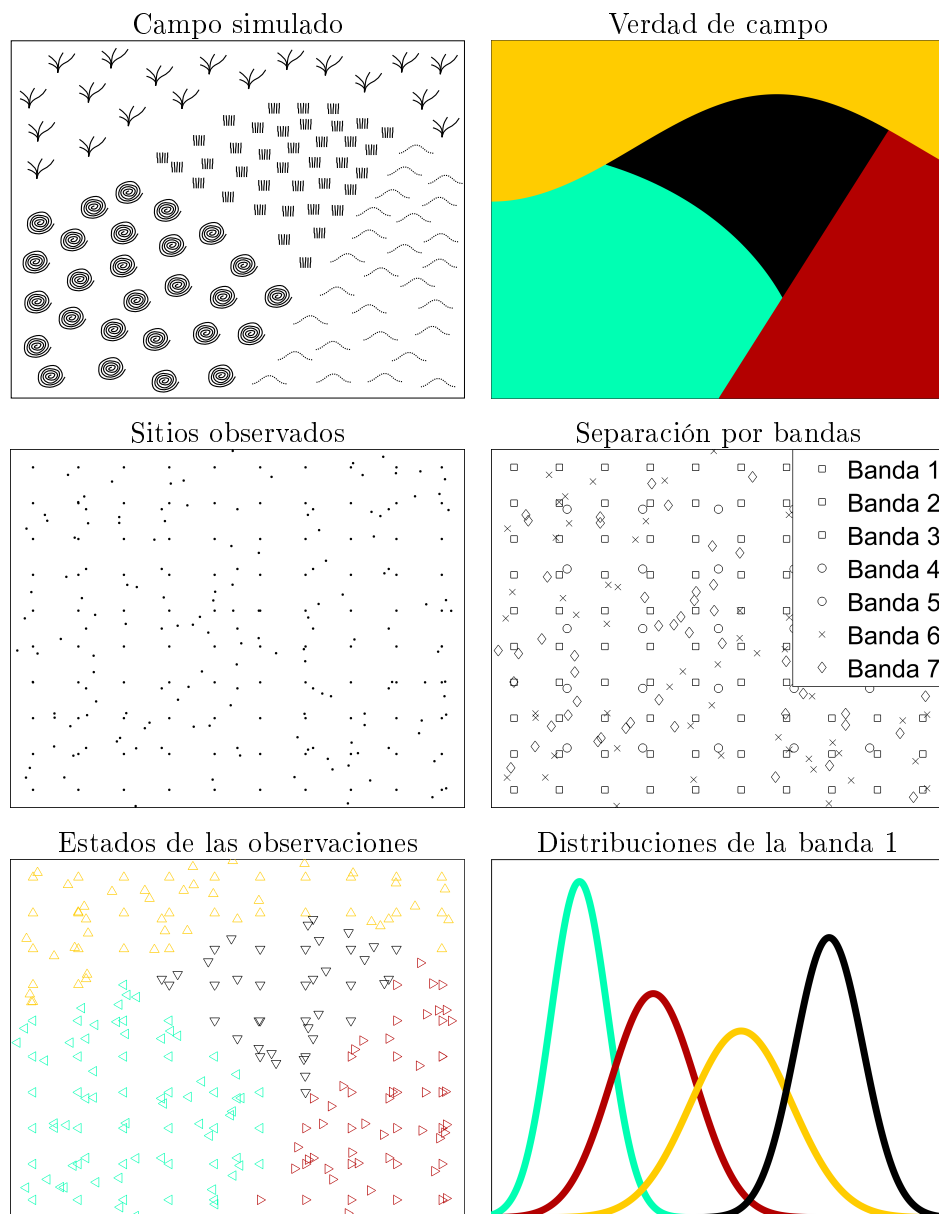


FIGURA 7.26: Configuración del experimento de fusión de datos con observaciones simuladas. Las bandas 1, 2 y 3 representan una imagen óptica multispectral. Por lo tanto, estas observaciones comparten los mismos sitios equidistantes. Asimismo, las bandas 4 y 5 son observaciones multispectrales de una imagen de radar con una resolución distinta a la de la imagen óptica.

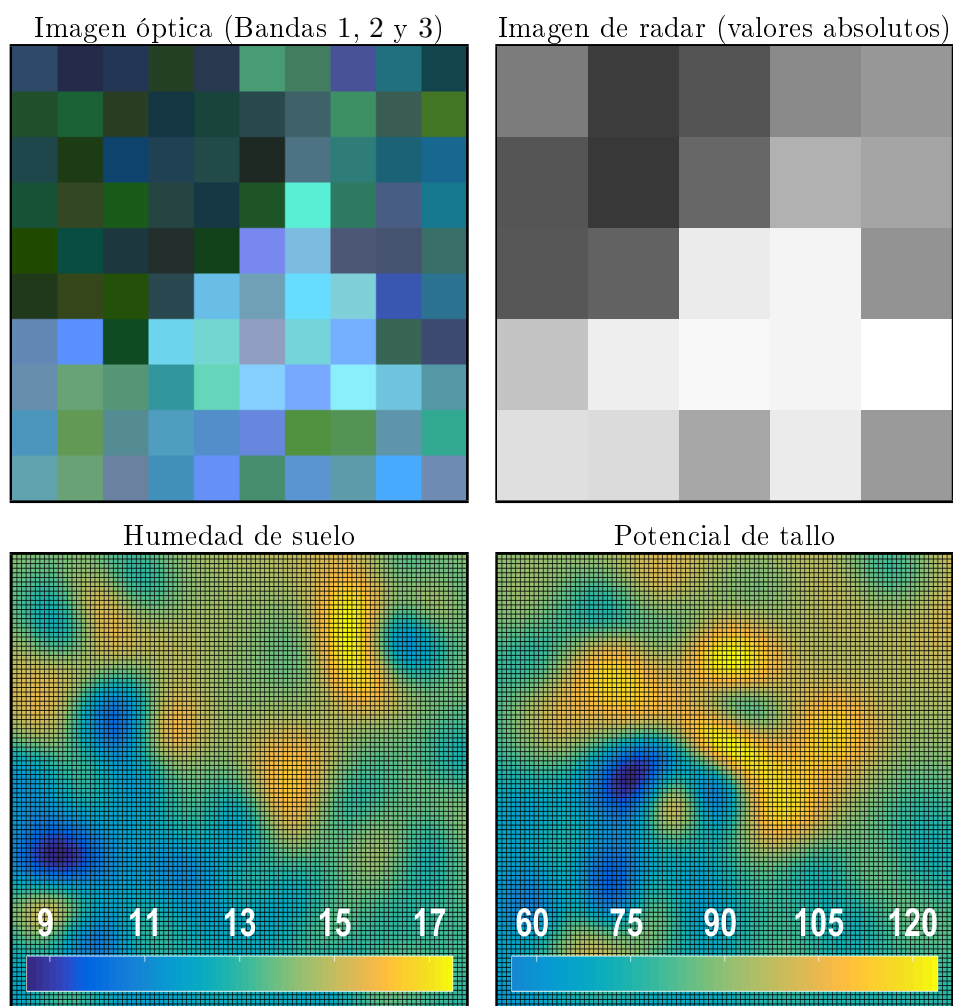


FIGURA 7.27: Datos simulados en el experimento de fusión de datos. En el caso de las mediciones locales (humedad de suelo y potencial de tallo) se ajustó una curva en dos dimensiones a los datos para visualizarlos mejor.

de Córdoba. El número de identificación de esta imagen es *Path 229, Row 082*. Si bien sería óptimo que el satélite pase en el mismo momento en que se realizan las mediciones hay que notar que por varias razones (días no hábiles, días nublados, disponibilidad de todos los sensores, permiso de acceder al lote, disponibilidad horaria) es difícil coordinar las mediciones locales con el satélite Landsat 8. Sin embargo, es razonable suponer que el lote se encontraba prácticamente en las mismas condiciones durante dos días siguientes.

La latitud del lote estudiado es $-31,524121$ y la longitud es $-64,453456$. El tamaño de la zona de interés es de aproximadamente $450\text{ m} \times 450\text{ m}$. En total se midió la humedad de suelo en 7 sitios, mientras se cuantificó la fotosíntesis de 5 plantas. Además, se midió el potencial de tallo de 5 plantas. Sumando a estas observaciones 256 píxeles de la imagen

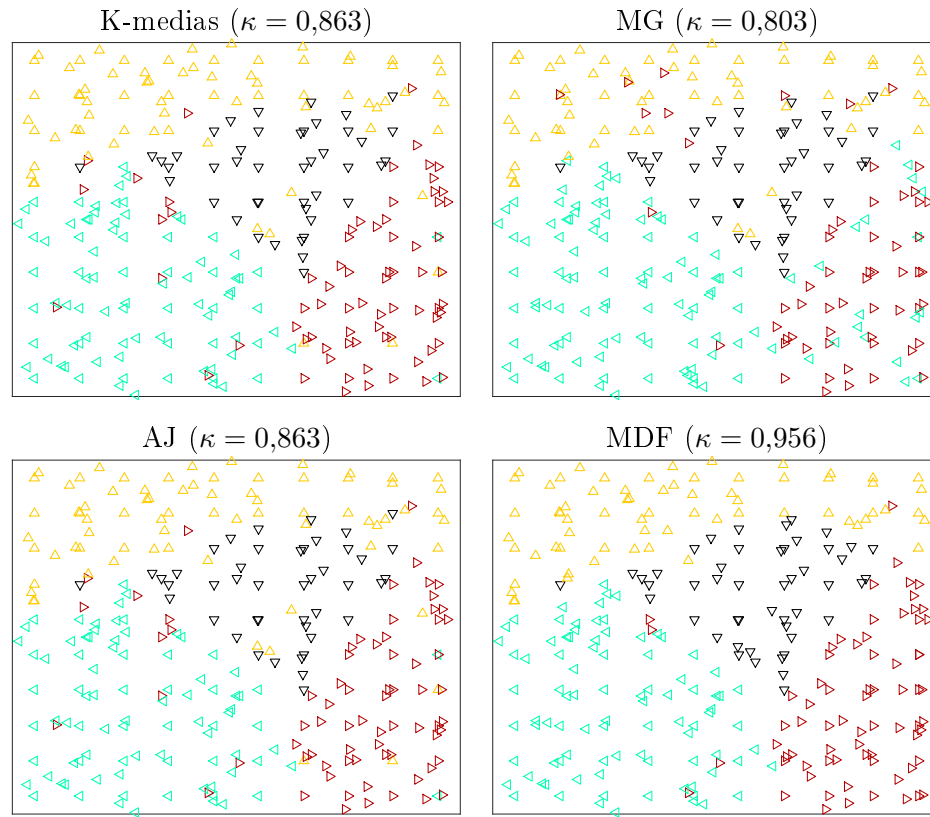


FIGURA 7.28: Resultados de fusión de datos con observaciones simuladas. En aproximadamente 40% de los casos K-medias y AJ convergen a la misma solución. El desempeño de MDF es claramente mejor que el de los competidores K-medias, MG y AJ.

Landsat 8, llegamos a un total de 273 observaciones. Hacemos hincapié en que todas las mediciones (igual que los píxeles de la imagen satelital) se repartieron por todo el lote de forma casi uniforme sin dejar zonas no observadas. En la figura 7.30 se presenta el esquema general y la verdad de campo. Además, se muestran los datos usados en este experimento en la figura 7.31.

En la figura 7.32 se muestran algunos resultados del medidor de potencial de tallo. Si bien los resultados de frecuencias variables son muy prometedores, se usan solo frecuencias constantes en este trabajo ya que es difícil coordinar la señal de entrada (generado por el celular) con el intervalo de medición. En cambio, para frecuencias estacionarias no influye en qué momento se empieza a grabar la salida. Entonces, se registran respuestas estacionarias para 100 Hz, 1000 Hz y 5000 Hz. Después se calcula la magnitud y la fase para las tres frecuencias. Por ende, se trabaja con 6 valores para cada planta.

Para evaluar los algoritmos de fusión de datos, se utiliza una verdad de campo hecho

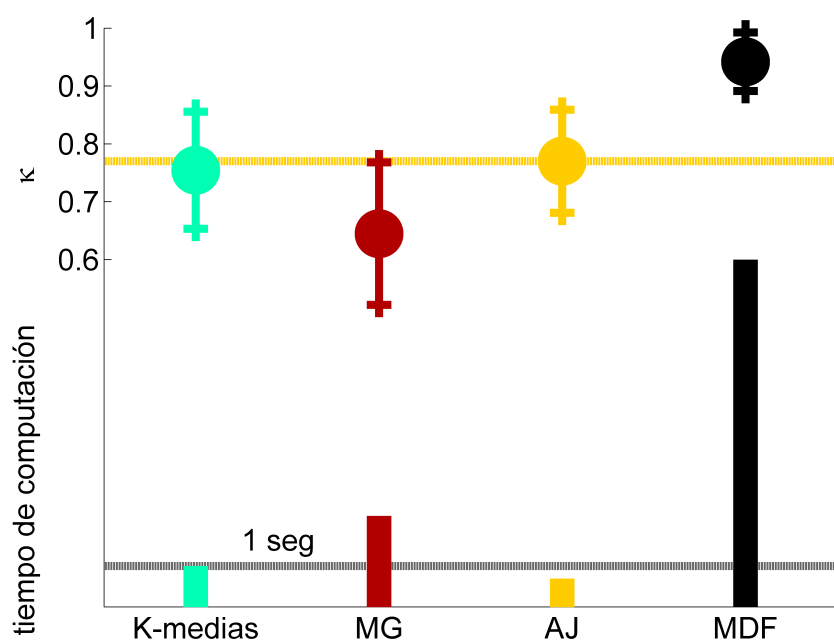


FIGURA 7.29: Evaluación del experimento de fusión de datos con observaciones simuladas. Se muestran los valores de κ promedios y las desviaciones estándar para 100 recorridos. En el caso de fusión de datos, se cuenta generalmente con pocos datos (278 sitios en este experimento) ya que cada medición *in situ* tiene un costo alto. Por lo tanto, los tiempos de computación son muy chicos con respecto al procesamiento de imágenes.

por un ingeniero agrónomo el mismo día que se realizaron las mediciones. Los resultados de los algoritmos de fusión de datos y los respectivos coeficientes de κ se presentan en la figura 7.33. Se observa que el coeficiente de κ de MDF es el más alto, seguido por K-medias.

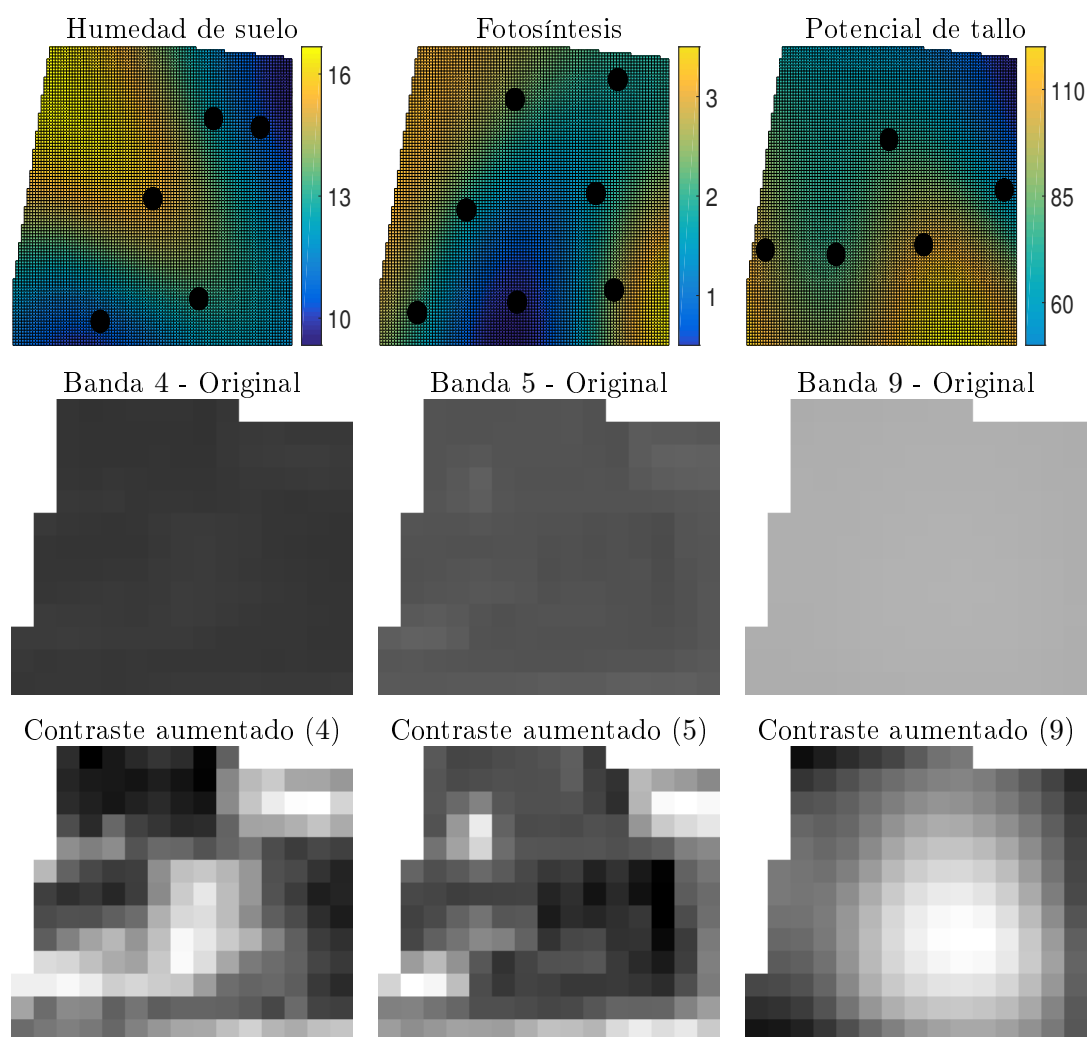


FIGURA 7.31: Datos reales para el experimento de fusión de datos. En el caso de las mediciones locales (humedad de suelo, fotosíntesis y potencial de tallo) se ajustó una curva en dos dimensiones a los datos para visualizarlos mejor. Cabe destacar que las bandas 1 a 8 del satélite Landsat 8 fueron tomadas con el sensor OLI, mientras las bandas 9 y 10 son producto del sensor TIRS que es un sensor de infrarrojo termal.

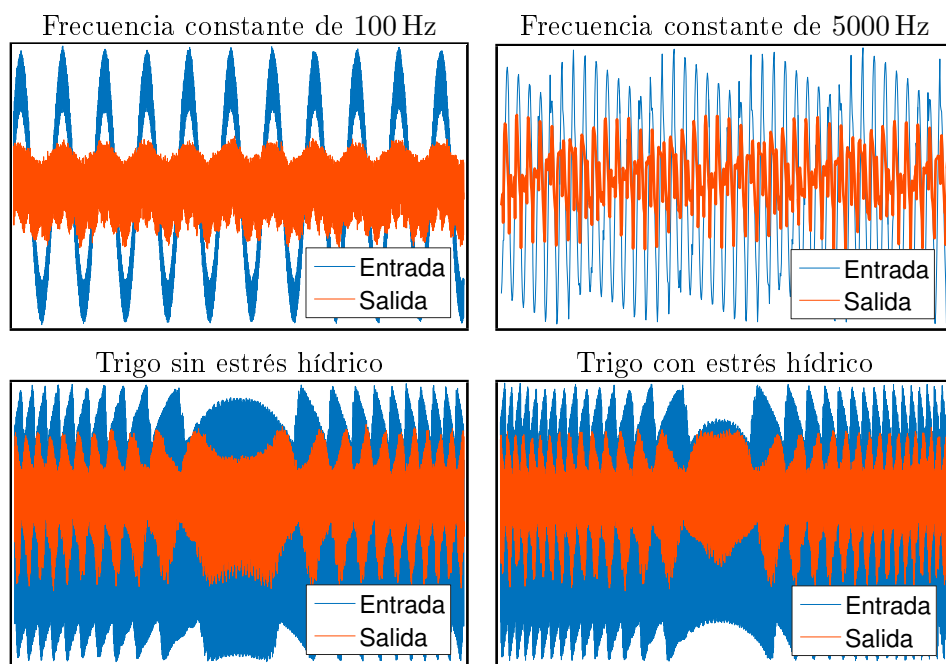


FIGURA 7.32: Resultados del medidor de potencial de tallo. Las dos respuestas de la segunda fila fueron obtenidas para frecuencias variables entre 20 Hz y 20 kHz.

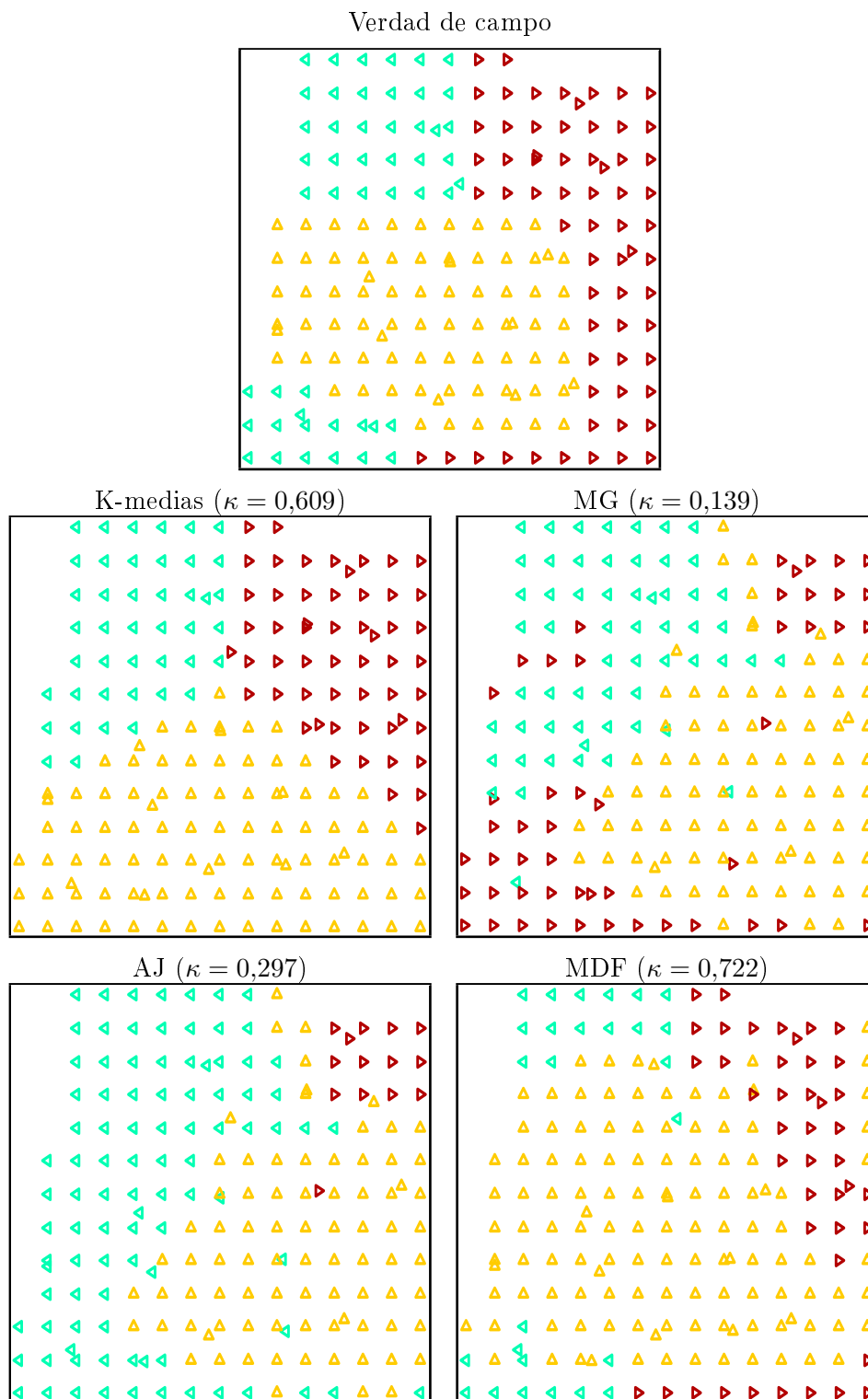


FIGURA 7.33: Resultados de fusión de datos con observaciones reales.

Capítulo 8

Evaluación final

8.1. Resumen de la tesis

En este trabajo se introdujeron, diseñaron y aplicaron modelos markovianos para reconocer patrones en lotes y zonas agrícolas partiendo de imágenes satelitales y mediciones *in situ*. En cuanto modelos ocultos de Markov de dos dimensiones, se utilizó el algoritmo PCVT como referencia y se diseñó el algoritmo CEP como competidor. En el caso de modelos isotrópicos de Potts se trabajó con ICM como algoritmo de referencia a partir del cual se desarrolló SBM. Como método no-contextual se usó MV. Además, se desarrolló un nuevo algoritmo de fusión de datos llamado MDF. Este algoritmo es una extensión de SBM de tal forma que se pueden procesar observaciones en sitios arbitrarios. Para evaluar MDF con datos reales se diseñó un novedoso medidor eléctrico denominado “potencia de tallo”. Por último, cabe destacar que en varias oportunidades se estudió la influencia de distintas funciones de probabilidad como funciones de emisión. Incluso, se derivaron tres métodos de estimar la función de densidad de cada clase “online” después cada iteración.

La motivación principal para el desarrollo de CEP fue que, PCVT descarta la gran mayoría de combinaciones de estados antes de correr el *Viterbi Training*. Para aliviar esta situación, se cambiaron algunas suposiciones de PCVT de tal forma que se obtuvo un conjunto de ecuaciones que permite hallar el mapa de estados de un modelo oculto de Markov de dos dimensiones sin tener que descartar ningún estado.

En el caso de los modelos de Potts, el algoritmo estándar desde hace muchos años es ICM. Sin embargo, ICM cuenta con algunas desventajas que se pretendieron resolver

al derivar el algoritmo SBM. En contrario a ICM, SBM es capaz de aplicar cualquier distribución de probabilidad a cualquier banda de una imagen multiespectral. Además, SBM puede trabajar con vecindades de distintos tamaños en cada banda.

Para poder procesar no solamente imágenes de teledetección sino también datos obtenidos por mediciones en sitios arbitrarios se extendió el concepto teórico del algoritmo SBM, obteniendo el algoritmo MDF. Este algoritmo claramente muestra un mejor desempeño que los competidores K-medias, MG y AJ para datos de distintos sensores.

Muchos sensores *in situ* aplicados en la agricultura de precisión tienen desventajas notables, como muestra la tabla 5.1. Por lo tanto, se desarrolló un medidor de bajo costo que mide directamente propiedades de la planta y no variables indirectas, como lo hace un medidor de humedad de suelo. El aparato llamado “Potencial de tallo” permite conocer el estado hídrico de una planta. Asimismo, se obtiene una estimación de la cantidad de nutrientes que tiene la planta en el momento de la medición.

8.2. Discusión de los resultados experimentales

Los algoritmos markovianos de este trabajo se han evaluado en distintos escenarios con datos reales y simulados. En esta sección, se indican los puntos donde los resultados fueron muy coherentes y se presentan los posibles motivos de las observaciones poco coherentes.

Antes que nada, es importante mencionar que el método MV – el único método no contextual – no logra un buen desempeño en ningún experimento. Esto refuerza la hipótesis de esta tesis, que se puede expresar de la siguiente manera: Se necesitan modelos markovianos (o sea modelos contextuales) para encontrar patrones en datos aleatorios provenientes de la agricultura de precisión.

A pesar de ser un método no contextual, hay de destacar que MV es el algoritmo con menor coste computacional. En segundo lugar, se ubica el algoritmo ICM cuya complejidad computacional está un poco encima de MV. En cambio, los algoritmos PCVT, CEP y SBM dependen fuertemente de ciertos parámetros, como la cantidad de clases, la cantidad de bandas, la cantidad de píxeles, el tamaño de la vecindad y la función de emisión. Como regla general, se puede constatar que la cantidad de píxeles junto con la

cantidad de clases es lo que más afecta a CEP, mientras SBM tiene un alto costo computacional al segmentar imágenes con muchas bandas. En el caso de PCVT, no existe una regla universal. Sin embargo, se observa que los tiempos de computación de PCVT son altos cuando el algoritmo necesita muchas iteraciones para converger. Esto ocurre sobre todo cuando la tarea de segmentación es difícil y los coeficientes de κ son bajos. Por último, cabe mencionar que los métodos de estimar la función de probabilidad “online” tienen un alto costo computacional, siendo KL el algoritmo que requiere notablemente más recursos que sus competidores.

Otro punto a discutir son los parámetros de los algoritmos. Por ejemplo, MV y CEP necesitan solamente la cantidad de clases y la distribución usada. En cambio, PCVT y ICM tienen un parámetro más para ajustar. En el caso de PCVT, se trata de la cantidad de secuencias probables para correr el Viterbi Training. Este parámetro se puede entender como un ajuste entre el tiempo de computación y el desempeño. En este trabajo se fija una cantidad de 100 secuencias por diagonal que previamente ha mostrado ser un buen compromiso. El parámetro libre de ICM es la vecindad aplicada. En algunos experimentos se investigó la influencia de la vecindad y no se pudo determinar una vecindad óptima. A veces una vecindad grande (7×7) resultó en mejores segmentaciones y otras veces vecindades medias (5×5) o chicas (3×3) obtuvieron los mejores resultados. Por ende, es necesario ajustar ese parámetro a cada tarea individualmente. Pero, en contrario a SBM, no hay un estimador que determina el tamaño de la vecindad automáticamente. Por otro lado, SBM tiene otro parámetro más para ajustar, que es el kernel del filtro bilateral. Cabe destacar que en este trabajo se propuso un estimador para el kernel de SBM.

Evaluando el desempeño general de los algoritmos, se observan varios puntos importantes. Primero, si bien ICM suaviza el mapa de estados de MV, se nota que la solución de ICM generalmente no difiere mucho de la solución propuesta por MV. La razón de esto, es que ICM trabaja con las mismas probabilidades marginales que MV durante todas las iteraciones. Solo se cambian los estados en función de los estados vecinos. En cambio, PCVT, CEP y SBM re-ajustan las funciones de emisión después de cada iteración. Esto permite obtener segmentaciones que se distinguen notablemente de la segmentación MV. Segundo, se observa que mientras más alto el coeficiente κ , mejor el desempeño de CEP en comparación con los otros algoritmos. O sea, CEP es el algoritmo óptimo para imágenes donde los segmentos están bien marcados. Tercero, PCVT se destaca en muy pocas situaciones y por ende es un algoritmo a mejorar. Cuarto, a pesar de que SBM no siempre

(pero muchas veces) logra los coeficientes κ más altos, se observa que este algoritmo encuentra patrones (y segmentos) muy diferentes a los demás algoritmos. Por lo tanto, SBM no solamente es una alternativa interesante a los algoritmos de referencia, sino también un muy buen complemento.

En cuanto la estimación “online” de las funciones de probabilidad, hay que destacar el algoritmo KL. Por un lado, este método tiene un costo computacional más alto que los algoritmos Bayes y KS. Por el otro, KL muestra el mejor desempeño de los tres métodos en todos los experimentos. La única incoherencia que tiene KL es que logra altos valores de κ a pesar de acertar la función de emisión con igual o menor precisión que Bayes y KS. La razón de esta incoherencia es probablemente que KL muchas veces encuentra funciones de probabilidad aptas para modelar los datos observados, aunque no sean las densidades correctas.

Los experimentos de fusión de datos muestran claramente que MDF es superior a los algoritmos de referencia al procesar mediciones de agricultura de precisión. Sin embargo, MDF requiere más recursos computacionales que los algoritmos de referencia K-medias, MG y AJ. Una de las grandes ventajas de MDF es que es capaz de ponderar los datos de sensores de buena calidad y al mismo tiempo bajar la prioridad a los datos de sensores de mala calidad.

En el experimento de fusión de datos con observaciones reales surgió un problema inesperado. Cada medición local demora aproximadamente 10 min, contando el tiempo para llegar al punto de interés (a pie), el uso del sensor y la determinación de la ubicación con el GPS. Por lo tanto, se realizaron solo pocas mediciones con cada sensor (entre 5 y 7 mediciones por sensor). En el caso de los algoritmos K-medias, MG y AJ la cantidad de mediciones por sensor no influye en el resultado final. Sin embargo, ocurre un problema con el algoritmo MDF. Recuerde que MDF trabaja con una distribución única por clase y por sensor. Entonces, si el objetivo es encontrar tres clases, se necesitarían por lo menos nueve datos por sensor para poder ajustar correctamente los parámetros de las tres distribuciones de probabilidad (mejor aún sería tener más que 12 datos por sensor). Si no está disponible esta cantidad de datos hay que ajustar las funciones de probabilidad de MDF con subconjuntos de datos, que claramente no es óptimo.

Por último, es importante mencionar que en este trabajo se presentaron los resultados de todos los experimentos, sin tener en cuenta que algoritmo mostró el mejor desempeño.

Además, se evitó completamente la aplicación de métodos de pre- y pos-procesamiento, como filtros no-lineales, que en muchos trabajos son utilizados para “pulir” los resultados finales. Por lo tanto, todos los experimentos del capítulo 7 son completamente reproducibles.

8.3. Conclusiones

La hipótesis principal de esta tesis es que los datos observados para la agricultura de precisión se pueden modelar como campos aleatorios de Markov. Usando el teorema de Hammersley-Clifford se puede demostrar que un campo aleatorio de Markov es equivalente a un campo aleatorio de Gibbs, donde las observaciones siguen la distribución de Gibbs. Entonces, para comprobar si es válido usar modelos markovianos en el ámbito de la agricultura de precisión, se evaluaron dos modelos ocultos de Markov y dos modelos gibbsianos para distintos tipos de datos.

La primera conclusión de esta tesis es que los modelos markovianos requieren varias suposiciones para ser tratables. Sin estas suposiciones algunos modelos (como SBM) tendrían un tiempo de computación muy elevado mientras otros modelos (como PCVT) directamente no se podrían computar dentro de miles de milenios. Por lo tanto, los resultados y conclusiones de este trabajo se aplican solamente a los modelos tratables que son derivados de verdaderos modelos markovianos. Sin embargo, todos los algoritmos contextuales (PCT, CEP, ICM y SBM) procesan la información de acuerdo al paradigma de Markov. Esto quiere decir que cada observación está influenciada solo por algunas observaciones que cumplan con alguna noción de cercanía o pasado.

La segunda conclusión es que los modelos markovianos tienen un desempeño mucho mejor que MV, que es un método no contextual. Esta observación es muy coherente en todos los experimentos, ya sean datos que están directamente relacionados con la agricultura de precisión (como imágenes satelitales o mediciones *in situ*), datos simulados o datos de referencia general (como la imagen “Cameraman”). Por consiguiente, se puede afirmar la hipótesis de esta tesis en el sentido que el modelado como campo aleatorio de Markov es más adecuado que el modelado de observaciones aisladas como lo hace MV. Sin embargo, falta comprobar en futuros trabajos si los modelos markovianos se adaptan mejor a las

observaciones que otras clases de modelos estadísticos, como por ejemplo modelos de textura.

Otro aspecto de este trabajo es que muchas veces se observaron grandes diferencias entre los cuatro modelos markovianos. Como tendencia general, se puede constatar que ICM es el método que más suaviza las observaciones mientras SBM es el algoritmo que más mantiene la estructura de las observaciones. En otras palabras, ICM pone mucho énfasis en los datos contextuales; mientras SBM prioriza a las probabilidades marginales. La razón de esto es que ICM solo cuenta los estados ocultos de los vecinos sin tener en cuenta si es un vecino cercano o lejano. En cambio, SBM asigna un peso particular a cada vecino de acuerdo a su distancia. De esta manera la influencia de los vecinos lejanos es muy limitada. En este contexto cabe destacar que ICM no actualiza las probabilidades marginales después de cada iteración y por ende obtiene resultados que una estructura parecida a MV (aunque mucho más suavizado).

Igual que los modelos de Potts (ICM y SBM), PCVT y CEP tienen sus características propias. Por ejemplo, PCVT parece ser muy sensible con respecto a la función de probabilidad. Esto se ve en varios experimentos, donde los valores de κ de PCVT cambian bruscamente al cambiar la distribución de probabilidad. Asimismo, una particularidad de CEP es que muestra una alta sensibilidad con respecto a la calidad de los datos. En los casos donde todos los algoritmos tienen altos coeficientes de κ , CEP es el método que mejor anda. En cambio, en experimentos con mala relación señal-ruido, CEP es uno de los algoritmos con el peor desempeño.

En cuanto funciones de probabilidad, se puede concluir que la distribución aplicada tiene una influencia notable en los resultados finales. Especialmente las densidades *Logistic* y *Weibull* mostraron su capacidad de describir datos aleatorios de distintos experimentos. Aparte de evaluar densidades individualmente, se propusieron tres algoritmos para encontrar la función de probabilidad óptima. La evaluación de estos tres métodos dio un claro ganador: Los algoritmos llamados Bayes y KS tiene un mal desempeño mientras el método KL logra buenos resultados. Por otro lado, el tiempo de computación de KL es notablemente más grande que el de Bayes y KS. En futuros trabajos es necesario estudiar si el desempeño de los modelos markovianos realmente mejora al usar KL en vez de una distribución común como es la distribución Normal o Weibull.

Evaluando los algoritmos de fusión de datos, la conclusión es muy sencilla. El método propuesto en este trabajo (MDF) claramente logra fusionar los datos de distintos sensores mejor que sus competidores, K-medias, MG y AJ. Tanto en los datos simulados, como en los datos reales se muestra una ventaja importante de MDF. Por otro lado, hay que reconocer que el algoritmo MDF necesita una cierta cantidad de observaciones por sensor. Considerando el tiempo de adquisición de datos “a mano”, es posible que MDF sea más adecuado para lotes donde haya sensores instalados fijos. De esta manera se obtienen muchos datos en poco tiempo, y, por ende, se evitan problemas a la hora de estimar las densidades de MDF.

En resumen, los resultados experimentales demuestran que realmente hay una gran variabilidad espacial en casi cada lote agrícola. Recuerde que esta característica es fundamental para la agricultura de precisión. Entonces, vale la pena pensar en las posibles oportunidades y limitaciones de aplicar los modelos de este trabajo a la realidad. El primer obstáculo de usar los modelos markovianos en la práctica es que se necesita un usuario que puede leer a los datos crudos en el formato correcto, luego pone a correr los algoritmos y finalmente saber interpretar los resultados. Además, hay problemas técnicos, como por ejemplo imágenes satelitales inútiles en caso de que el cielo este nublado, o el manejo correcto del medidor de fotosíntesis que requiere un ajuste complicado antes de funcionar correctamente. Otro desafío es como aprovechar los resultados de los modelos markovianos. Lo ideal sería transformar el mapa de estados a coordenadas de GPS y luego cargar esas coordenadas junto con indicaciones claras a las máquinas agrícolas. De esta forma se lograría un manejo óptimo de cada zona (segmento). Por otro lado, no sería difícil usar los algoritmos de esta tesis (junto con todos los datos disponibles) para obtener una idea general de las distintas zonas que se encuentran en un lote de interés. Este camino sencillo podría ser un buen inicio para empezar a incorporar modelos abstractos en el proceso de la agricultura de precisión.

8.4. Futuros trabajos

Como se ha demostrado en esta tesis, los modelos markovianos tienen una alta capacidad de encontrar patrones en datos ruidosos provenientes de la agricultura de precisión. Sin embargo, hay varios puntos abiertos para ser estudiados en futuros trabajos.

Una desventaja de los algoritmos desarrollados en esta tesis es que requieren muchos recursos computacionales. Para disminuir los tiempos de computación hay dos posibles caminos. Primero, cambiar las suposiciones teóricas de tal forma que se simplifiquen las cuentas necesarias. Una idea sería no actualizar las funciones de emisión de PCVT, CEP y SBM después de cada iteración siguiendo el ejemplo de ICM. La segunda opción para bajar el tiempo de computación, es aplicar simplificaciones en la implementación de los algoritmos. Por ejemplo, se podría establecer un umbral para considerar nulo las probabilidades que estén debajo de ese umbral.

Otra posible línea de investigación es mejorar los algoritmos presentados en este trabajo. En el caso de los modelos causales (PCVT y CEP) esto podría significar generalizar la notación del “pasado”. Por ejemplo, se podrían realizar cuatro recorridas sucesivas que empiecen en los cuatro rincones de una imagen. O se podrían agregar más píxeles al “pasado”. En cuanto ICM, se podría diseñar una versión que no requiera de la distribución normal como función de emisión y que estime el tamaño de la vecindad “online” en cada iteración, como lo hace SBM.

Si bien los modelos markovianos lograron buenos resultados (suponiendo que los datos se pueden modelar como un campo aleatorio de Markov) se podría estudiar la posibilidad de extender los algoritmos de tal forma que encuentren segmentos homogéneos y también texturas. Como motivación, uno puede pensar en la marca de las ruedas de los tractores como patrón de textura.

Otro aspecto interesante para futuros trabajos es extender los modelos existentes a tres dimensiones, donde la tercera dimensión representa el tiempo. De esta manera, se podrían relacionar datos que fueron tomados en distintos instantes de tiempo. Entonces, sería posible analizar y controlar el crecimiento de un cultivo desde la siembra hasta la cosecha.

Por último, hay que destacar que los resultados del medidor de potencial de tallo son muy prometedores. Sin embargo, es necesario mejorar este sistema para poder grabar datos de forma automática en un escenario con frecuencias variables. Además, es importante cambiar el diseño del medidor de tal forma que sea robusto y fácil de manejar para achicar los tiempos de medición.

En resumen, hay varios desafíos para abarcar en futuros trabajos. Sobre todo, el avance de los penúltimos dos puntos va a ser crucial para que modelos markovianos sigan jugando

un rol importante en el modelado y el reconocimiento de patrones para la agricultura de precisión.

Apéndice A

Teorema de Hammersley-Clifford

El teorema de Hammersley-Clifford dice que las siguientes dos definiciones son equivalentes.

Definición A.1. Un grafo no dirigido \mathcal{G} se llama campo aleatorio de Markov si dos nodos son condicionalmente independiente siempre cuando hay un nodo de evidencia que los separe. En otras palabras, para cualquier nodo X_i del grafo vale la siguiente propiedad condicional:

$$P(X_i | X_{\mathcal{G} \setminus i}) = P(X_i | X_{N_i}), \quad (\text{A.1})$$

donde $X_{\mathcal{G} \setminus i}$ representa todos los nodos excepto X_i , y X_{N_i} son todos los nodos de la vecindad de X_i .

Definición A.2. Una función de probabilidad $P(X)$ definida sobre un grafo no dirigido \mathcal{G} se llama distribución de Gibbs si $P(X)$ puede ser factorizada en funciones positivas que son definidas para cliques que cubren todos los nodos y bordes de \mathcal{G} . Esto se puede expresar como:

$$P(X) = \frac{1}{Z} \prod_{c \in C_{\mathcal{G}}} \phi_c(X_c), \quad (\text{A.2})$$

donde $Z = \sum_x \prod_{c \in C_{\mathcal{G}}} \phi_c(X_c)$ es una constante de normalización y $C_{\mathcal{G}}$ es un conjunto de todos los cliques en \mathcal{G} .

Existen varias demostraciones del teorema de Hammersley-Clifford [57]. Por ejemplo, Besag presentó una demostración en [56]. Sin embargo, en este apéndice seguimos el camino de Cheung [131].

Primero, demostramos que un campo aleatorio de Markov se puede describir como la distribución de Gibbs. Esta demostración es constructiva en el sentido que vamos a mostrar como se puede expresar $\phi_c(X_c)$ a partir de la probabilidad conjunta y la condición de Markov. Para cualquier conjunto $s \subset \mathcal{G}$ definimos una posible función de la siguiente manera:

$$f_s(X_s = x_s) = \prod_{z \subset s} P(X_z, X_{\mathcal{G} \setminus z} = 0)^{-1^{|s|-|z|}} \quad (\text{A.3})$$

Note en la ecuación (A.3) que se calcula un producto de todos los conjuntos de s . Entonces, dado un conjunto z de s , $P(X_z, X_{\mathcal{G} \setminus z} = 0)$ significa que los nodos que contiene z coinciden con la correspondiente parte de X_s mientras lo que queda del grafo se toma como configuración estándar “0”. Ahora, fíjese en la ecuación (A.3) que la diferencia de tamaño entre s y z determina si la potencia de $P(X_z, X_{\mathcal{G} \setminus z} = 0)$ es $+1$ o -1 . Y como f_s es obviamente positivo lo único que hay que demostrar es:

1. $\prod_{s \subset \mathcal{G}} f_s(X_s) = P(X)$
2. $f_s(X_s) = 1$ si s no es un clique

Para demostrar el punto 1 empezamos con una identidad simple:

$$0 = (1 - 1)^K = C_0^K - C_1^K + C_2^K - \dots + (-1)^K C_K^K, \quad (\text{A.4})$$

donde C_N^K representa la cantidad de combinaciones en caso de elegir N elementos de un conjunto con un total de K elementos. Lo que se busca mostrar es que todos los factores de $\prod_{s \subset \mathcal{G}} f_s(X_s)$ se cancelan, excepto $P(X)$.

En lo siguiente, consideramos cualquier conjunto $z \subset \mathcal{G}$ y el correspondiente factor $\Delta = P(X_z, X_{\mathcal{G} \setminus z} = 0)$ y nos damos cuenta que este factor aparece en $f_z(X_z)$ como $\Delta^{-1^0} = \Delta$. El mismo factor también aparece en las $C_1^{|\mathcal{G}|-|z|}$ funciones de los conjuntos que contienen z y, además, en un elemento adicional que es responsable por el factor inverso $\Delta^{-1^1} = \Delta^{-1}$. Siguiendo con las $C_2^{|\mathcal{G}|-|z|}$ funciones de los conjuntos que contienen z y los dos elementos adicionales se genera un factor $\Delta^{-1^2} = \Delta$. Si continuamos con este proceso, la ecuación

(A.4) ayuda a demostrar que todos los términos se cancelan, salvo el caso cuando $z = \mathcal{G}$. En este caso hay un solo factor que es $P(x)$.

Para demostrar el punto 2 se necesita la condición de Markov. Como s no es un clique, tenemos que encontrar dos nodos a y b que no son conectados. Por lo tanto, expresamos $f_s(X_s)$ de la siguiente manera:

$$\begin{aligned}
 f_s(X_s = x_s) &= \prod_{z \subset s} P(X_z = x_z, X_{\mathcal{G} \setminus z} = 0)^{-1^{|s|-|z|}} \\
 &= \prod_{w \subset s \setminus \{a,b\}} \left[\frac{P(X_w, X_{\mathcal{G} \setminus w} = 0) P(X_{w \cup \{a,b\}}, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0)}{P(X_{w \cup \{a\}}, X_{\mathcal{G} \setminus w \cup \{a\}} = 0) P(X_{w \cup \{b\}}, X_{\mathcal{G} \setminus w \cup \{b\}} = 0)} \right]^{-1^*}.
 \end{aligned} \tag{A.5}$$

Lo que hace la ecuación (A.5) es clasificar cada conjunto $z \in s$ en una de cuatro clases: $z = w$, $z = w \cup \{a\}$, $z = w \cup \{b\}$ y $z = w \cup \{a, b\}$. Cabe destacar que no importa si la potencia del producto en la ecuación (A.5) es $+1$ o -1 y por ende la expresamos como -1^* .

Ahora, demostraremos que los cuatro términos de la ecuación (A.5) se cancelan. Usando la regla de Bayes obtenemos:

$$\frac{P(X_w, X_{\mathcal{G} \setminus w} = 0)}{P(X_{w \cup \{a\}}, X_{\mathcal{G} \setminus w \cup \{a\}} = 0)} \tag{A.6}$$

$$= \frac{P(X_a = 0 | X_b = 0, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0) P(X_b = 0, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0)}{P(X_a | X_b = 0, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0) P(X_b = 0, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0)} \tag{A.7}$$

$$= \frac{P(X_a = 0 | X_b, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0) P(X_b, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0)}{P(X_a | X_b, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0) P(X_b, X_w, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0)} \tag{A.8}$$

$$= \frac{P(X_{w \cup \{b\}}, X_{\mathcal{G} \setminus w \cup \{b\}} = 0)}{P(X_{w \cup \{a,b\}}, X_{\mathcal{G} \setminus w \cup \{a,b\}} = 0)} \tag{A.9}$$

Note que, en la ecuación (A.7) intencionalmente no se cancelan los segundos factores (el a priori) para cambiar el valor estándar de $X_b = 0$ al valor actual X_b en la ecuación (A.8). Esta operación es válida porque X_a y X_b son condicionalmente independiente dado que el resto del grafo. Por lo tanto, se puede reemplazar la condición X_b con cualquier valor.

En lo siguiente, demostraremos que una distribución de Gibbs es equivalente a un campo aleatorio de Markov. Primero, definimos que $D_i = N_i \cup \{X_i\}$ es el conjunto de nodos que contienen el nodo X_i y también su vecindad. Empezamos con la definición de un campo

aleatorio de Markov:

$$P(X_i|X_{N_i}) = \frac{P(X_i|X_{N_i})}{P(X_{N_i})} \quad (\text{A.10})$$

$$= \frac{\sum_{\mathcal{G} \setminus D_i} \prod_{c \in \mathcal{C}_{\mathcal{G}}} \phi_c(X_c)}{\sum_{x_i} \sum_{\mathcal{G} \setminus D_i} \prod_{c \in \mathcal{C}_{\mathcal{G}}} \phi_c(X_c)} \quad (\text{A.11})$$

Ahora, si partimos el conjunto máximo de cliques $\mathcal{C}_{\mathcal{G}}$ en dos partes donde una parte contiene X_i y la otra parte no. Obtenemos $C_i = \{c \in \mathcal{C}_{\mathcal{G}} : X_i \in c\}$ y $R_i = \{c \in \mathcal{C}_{\mathcal{G}} : X_i \notin c\}$. Con esta notación podemos separar la ecuación (A.11) en dos productos de C_i y R_i .

$$P(X_i|X_{N_i}) = \frac{\sum_{\mathcal{G} \setminus D_i} \prod_{c \in C_i} \phi_c(X_c) \prod_{c \in R_i} \phi_c(X_c)}{\sum_{x_i} \sum_{\mathcal{G} \setminus D_i} \prod_{c \in C_i} \phi_c(X_c) \prod_{c \in R_i} \phi_c(X_c)} \quad (\text{A.12})$$

$$= \frac{\prod_{c \in C_i} \phi_c(X_c) \sum_{\mathcal{G} \setminus D_i} \prod_{c \in R_i} \phi_c(X_c)}{\sum_{x_i} \prod_{c \in C_i} \phi_c(X_c) \sum_{\mathcal{G} \setminus D_i} \prod_{c \in R_i} \phi_c(X_c)} \quad (\text{A.13})$$

Note en la ecuación (A.13) que la suma sobre $\mathcal{G} \setminus D_i$ solo aplica a los cliques de R_i ya que C_i viene dado por D_i . Otro aspecto importante es que el termino $\sum_{\mathcal{G} \setminus D_i} \prod_{c \in R_i} \phi_c(X_c)$ no involucra X_i . Por lo tanto, este termino puede ser separado de la suma y luego se cancela con su par en el numerador. Entonces tenemos

$$P(X_i|X_{N_i}) = \frac{\prod_{c \in C_i} \phi_c(X_c)}{\sum_{x_i} \prod_{c \in C_i} \phi_c(X_c)} \quad (\text{A.14})$$

$$= \frac{\prod_{c \in C_i} \phi_c(X_c)}{\sum_{x_i} \prod_{c \in C_i} \phi_c(X_c)} \cdot \frac{\prod_{c \in R_i} \phi_c(X_c)}{\prod_{c \in R_i} \phi_c(X_c)} \quad (\text{A.15})$$

$$= \frac{\prod_{c \in \mathcal{C}_{\mathcal{G}}} \phi_c(X_c)}{\sum_{x_i} \prod_{c \in \mathcal{C}_{\mathcal{G}}} \phi_c(X_c)} \quad (\text{A.16})$$

$$= \frac{P(X)}{P(X_{\mathcal{G} \setminus \{i\}})} \quad (\text{A.17})$$

$$= P(X_i|X_{\mathcal{G} \setminus \{i\}}) \quad (\text{A.18})$$

Después de eliminar la suma de $\mathcal{G} \setminus D_i$ podemos multiplicar el numerador y el denominador con el mismo factor en la ecuación (A.15) para tener la funciones de potencial del resto de grafo. De esta manera, podemos volver a los términos $P(X)$ y finalmente $P(X_i|X_{\mathcal{G} \setminus \{i\}})$. Una vez que tenemos $P(X_i|X_{\mathcal{G} \setminus \{i\}})$ terminamos de demostrar que una distribución de Gibbs realmente es equivalente a un campo aleatorio de Markov. ■

Apéndice B

Distribuciones de probabilidad

Definición B.1. La distribución **Gamma** viene dada por:

$$f(x|a, b) = \frac{1}{b^a \Gamma(a)} x^{a-1} \exp \left\{ - \left(\frac{x}{b} \right)^b \right\}, \quad (\text{B.1})$$

donde Γ es la función Gamma y los dos parámetros son a y b . La distribución Gamma esta solo definida para $x \geq 0$.

Definición B.2. La distribución **Generalized Extreme Value** viene dada por:

$$f(x|k, \mu, \sigma) = \frac{1}{\sigma} \exp \left(- \left(1 + k \frac{x - \mu}{\sigma} \right)^{-\frac{1}{k}} \right) \left(1 + k \frac{x - \mu}{\sigma} \right)^{-1 - \frac{1}{k}}, \quad (\text{B.2})$$

con los tres parámetros k , μ y σ . La distribución Generalized Extreme Value esta definida para cualquier valor de x , o sea $-\infty < x < \infty$.

Definición B.3. La distribución **Inverse Gaussian** viene dada por:

$$f(x|\mu, \lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp \left\{ - \frac{\lambda}{2\mu^2 x} (x - \mu)^2 \right\}, \quad (\text{B.3})$$

con los dos parámetros μ y λ . Esta distribución, también conocida como distribución de Wald, esta solo definida para $x > 0$.

Definición B.4. La distribución **Logistic** viene dada por:

$$f(x|\mu, \sigma) = \frac{\exp\left\{\frac{x-\mu}{\sigma}\right\}}{\sigma \left(1 + \exp\left\{\frac{x-\mu}{\sigma}\right\}\right)^2}, \quad (\text{B.4})$$

con los dos parámetros μ y σ . La distribución Logistic esta definida para cualquier valor de x , o sea $-\infty < x < \infty$.

Definición B.5. La distribución **Nakagami** viene dada por:

$$f(x|\mu, \sigma) = 2 \left(\frac{\mu}{\omega}\right)^\mu \frac{1}{\Gamma(\mu)} x^{(2\mu-1)} \exp\left\{-\frac{\mu}{\omega} x^2\right\}, \quad (\text{B.5})$$

con los dos parámetros μ y σ . La distribución Nakagami esta definida solo para $x \geq 0$.

Definición B.6. La distribución **Normal** viene dada por:

$$f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad (\text{B.6})$$

con los dos parámetros μ y σ . La distribución Normal esta definida cualquier valor de x , o sea $-\infty < x < \infty$.

Definición B.7. La distribución **Weibull** viene dada por:

$$f(x|a, b) = \frac{b}{a} \left(\frac{x}{a}\right)^{b-1} \exp\left\{-\left(\frac{x}{a}\right)^b\right\}, \quad (\text{B.7})$$

con los dos parámetros a y b . La distribución Weibull esta definida solo para $x \geq 0$.

Bibliografía

- [1] J. Sheeran. The challenge of hunger. *The Lancet*, 371(9608):180–181, 2008.
- [2] P. Gerland, A. E. Raftery, H. Ševčíková, N. Li, D. Gu, T. Spoorenberg, L. Alkema, B. K. Fosdick, J. Chunn, N. Lalic, G. Bay, T. Buettner, G. K. Heilig, and J. Wilmoth. World population stabilization unlikely this century. *Science*, 346(6206):234–237, 2014.
- [3] F. Pearce. *When the Rivers Run Dry: Journeys Into the Heart of the World’s Water Crisis*. Key Porter, 2006.
- [4] J. A. Krautkraemer. Population growth, soil fertility, and agricultural intensification. *Journal of Development Economics*, 44(2):403–428, 1994.
- [5] H. C. J. Godfray, J. R. Beddington, I. R. Crute, L. Haddad, D. Lawrence, J. F. Muir, J. Pretty, S. Robinson, S. M. Thomas, and C. Toulmin. Food security: The challenge of feeding 9 billion people. *Science*, 327(5967):812–818, 2010.
- [6] P. Erdkamp. *The Grain Market In The Roman Empire: A Social, Political And Economic Study*. Cambridge University Press, 2005.
- [7] M. Overton. *Agricultural Revolution in England: The transformation of the agrarian economy 1500-1850*. Cambridge University Press, 1996.
- [8] B. L. Gardner. *American Agriculture in the Twentieth Century: How It Flourished and What It Cost*. Harvard University Press, 2002.
- [9] E. Chartuni, F. de Assis de Carvalho, D. Marçal, and E. Ruz. Precision agriculture: New tools to improve technology management in agricultural enterprises. *COMUNICA*, 1(2):24–31, 2007.

-
- [10] S. Cox. Information technology: the global key to precision agriculture and sustainability. *Computers and Electronics in Agriculture*, 36(2-3):93–111, 2002.
- [11] B. M. Whelan and A. B. McBratney. The “null hypothesis” of precision agriculture management. *Precision Agriculture*, 2(3):265–279, 2000.
- [12] H. Lin, X. Liu, and X. Xue. Correlation analysis between biochemical contents of camphor and in situ hyperspectral measurements. In *MMIT '08 Proceedings of the 2008 International Conference on MultiMedia and Information Technology*, pages 756–760, 2008.
- [13] X. Xu, C. Zhao, J. Wang, J. Zhang, and X. Song. Using optimal combination method and in situ hyperspectral measurements to estimate leaf nitrogen concentration in barley. *Precision Agriculture*, 15(2):227–240, 2014.
- [14] G. S. Nearing, H. V. Gupta, and W. T. Crow. Information loss in approximately bayesian estimation techniques: A comparison of generative and discriminative approaches to estimating agricultural productivity. *Journal of Hydrology*, 507:163–173, 2013.
- [15] A. McBratney, B. Whelan, T. Ancey, and J. Bouma. Future directions of precision agriculture. *Precision Agriculture*, 6(1):7–23, 2005.
- [16] R. Dutta, C. Li, D. Smith, A. Das, and J. Aryal. Big data architecture for environmental analytics. *IFIP Advances in Information and Communication Technology*, 448:578–588, 2015.
- [17] V. Rossi, F. Salinari, S. Poni, T. Caffi, and T. Bettati. Addressing the implementation problem in agricultural decision support systems: The example of vite.net©. *Computers and Electronics in Agriculture*, 100:88–99, 2014.
- [18] J. He. Advances in data mining: History and future. In *Third International Symposium on Intelligent Information Technology Application*, volume 1, pages 634–636, Nov 2009.
- [19] F. Coenen. Data mining: past, present and future. *The Knowledge Engineering Review*, 26(1):25–29, 2011.
- [20] O. Maimon and L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer Science+Business Media, 2 edition, 2010.

- [21] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3 edition, 2011.
- [22] J. H. Meng, T. Dong, M. Zhang, X. You, and B. Wu. *Predicting optimal soybean harvesting dates with satellite data*, pages 209–215. Wageningen Academic Publishers, precision agriculture '13 edition, 2013.
- [23] C. Gomez, R. A. Viscarra Rossel, and A. B. McBratney. Soil organic carbon prediction by hyperspectral remote sensing and field vis-nir spectroscopy: An australian case study. *Geoderma*, 146(3):403–411, 2008.
- [24] C. Römer, M. Wahabzada, A. Ballvora, F. Pinto, M. Rossini, C. Panigada, J. Behmann, J. Léon, C. Thureau, C. Bauckhage, K. Kersting, U. Rascher, and L. Plümer. Early drought stress detection in cereals: simplex volume maximisation for hyperspectral image analysis. *Functional Plant Biology*, 39(11):878–890, 2012.
- [25] T. Rumpf, A. K. Mahlein, U. Steiner, E. C. Oerke, H.-W. Dehne, and L. Plümer. Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Computers and Electronics in Agriculture*, 74(1):91–99, 2010.
- [26] X. Meng, A. Dodson, J. Zhang, Y. Cai, C. Liu, and K. Geary. Geospatial data fusion for precision agriculture. In *2011 International Symposium on Image and Data Fusion (ISIDF)*, pages 1–4, 2011.
- [27] G. Ruß and A. Brenning. Data mining in precision agriculture: Management of spatial information. *Computational Intelligence for Knowledge-Based Systems Design*, 6178:350–359, 2010.
- [28] W. Xie, Y. Xue, L. Zhai, and H. Sang. Data fusion technology of multi-platform earth observation on agriculture. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:189–192, aug 2013.
- [29] V. I. Adamchuk, R. A. Viscarra Rossel, K. A. Sudduth, and P. Schulze Lammers. *Sensor Fusion for Precision Agriculture*, pages 27–40. InTech, sensor fusion - foundation and applications edition, 2011.
- [30] N. N. Taleb. *The Black Swan*. Random House, 2007.
- [31] Y. A. Rozanov. *Markov Random Fields*. Springer New York, 1982.

- [32] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.
- [33] J. E. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.
- [34] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [35] J. Baumgartner, C. Rodríguez Rivero, and J. Pucheta. A genetic algorithm based design approach for the properties of a gaussian process for time series forecasting. In *XXII Congreso Argentino de Control Automático - AADECA*, Buenos Aires, Argentina, 2010.
- [36] J. Baumgartner, C. Rodríguez Rivero, and J. Pucheta. Pronóstico de lluvia en un punto desde diversos puntos geográficos de observación mediante procesos gaussianos. In *XXIII Congreso Nacional del Agua - CONAGUA*, Resistencia, Chaco, Argentina, 2011.
- [37] J. Baumgartner, C. Rodríguez Rivero, and J. Pucheta. Identificación no-supervisada de parcelas agrícolas en imágenes satelitales multiespectrales basado en la semejanza de píxeles homólogos en las distintas bandas. In *III Congreso Argentino de AgroInformática - CAI*, Córdoba, Argentina, 2011.
- [38] J. Pucheta, M. Herrera, C. Rodríguez Rivero, and J. Baumgartner. Filtro predictor basado en redes neuronales para pronóstico de series temporales de lluvia acumulada empleando submuestreo. In *Jornadas de Informática - JAIIO 40, 3º Congreso Argentino de AgroInformática - CAI*, Córdoba, Argentina, 2011.
- [39] J. Pucheta, C. Rodríguez Rivero, M. Herrera, V. Sauchelli, and J. Baumgartner. Time series forecasting using kernel and feed-forward neural. In *XIV Reunión de Trabajo en Procesamiento de la Información y Control RPIC*, Oro Verde, Entre Ríos, Argentina, 2011.
- [40] C. Rodríguez Rivero, J. Pucheta, J. Baumgartner, M. Herrera, C. Salas, and V. Sauchelli. A nn-based autoregressive model that considers the energy associated of time series for forecasting. In *III Congreso de Mathematica Aplicada, Computacional e Industrial - III MACI 2011*, Bahía Blanca, Argentina, 2011.

- [41] C. Rodríguez Rivero, J. Pucheta, J. Baumgartner, M. Herrera, V. Sauchelli, D. Patiño, and S. Laboret. Neural networks based algorithms for forecasting short time series with uncertainties on their data. In *Proceedings of XXIII^o Argentine Association of Automation and Control*, Buenos Aires, Argentina, 2012.
- [42] J. Baumgartner, J. Gimenez, J. Pucheta, and A. G. Flesia. Classification of agricultural fields in satellite images using two-dimensional hidden markov models. In *V Congreso Argentino de AgroInformática - CAI*, Córdoba, Argentina, September 2013.
- [43] A. G. Flesia, J. Gimenez, and J. Baumgartner. On segmentation with markovian models. In *XIV Argentine Symposium on Artificial Intelligence - ASAI*, Córdoba, Argentina, September 2013.
- [44] J. Baumgartner, M. Scavuzzo, C. Rodríguez Rivero, and J. Pucheta. A new approach to segmentation of remote sensing images with hidden markov models. In *Biennial Congress of Argentina (ARGENCON), 2014 IEEE*, pages 130–135, 2014.
- [45] C. Rodríguez Rivero, J. Pucheta, J. Baumgartner, H. D. Patiño, and B. Kuchen. An approach for time series forecasting by simulating stochastic processes through time lagged feed-forward neural network. In *Proceedings of the 2010 International Conference on Data Mining*, volume 6, pages 287–293. CSREA Press, 2010.
- [46] C. Rodríguez Rivero, J. Pucheta, J. Baumgartner, M. Herrera, H. D. Patiño, and B. Kuchen. A nn-based model for time series forecasting in function of energy associated of series. In *Proc. of the International Conference on Applied, Numerical and Computational Mathematics - ICANCM'11*, ISBN 978-1-61804-030-5, pages 80–86, Barcelona, España, 2011.
- [47] J. Baumgartner, A. G. Flesia, J. Gimenez, and Julian Pucheta. A new approach to image segmentation with two-dimensional hidden markov models. In *1st BRICS Countries Congress on Computational Intelligence BRICS-CCI*, Recife, Brasil, September 2013.
- [48] C. Rodríguez Rivero, J. Pucheta, H. Patino, J. Baumgartner, S. Laboret, and V. Sauchelli. Analysis of a gaussian process and feed-forward neural networks based filter for forecasting short rainfall time series. In *The 2013 International Joint Conference on Neural Networks - IJCNN*, pages 1–6, 2013.

- [49] C. Rodríguez Rivero, M. Herrera, J. Pucheta, J. Baumgartner, H. D. Patiño, and V. Sauchelli. High roughness time series forecasting based on energy associated of series. *Journal of Communication and Computer*, 9(5):576–586, 2012.
- [50] J. Baumgartner, J. Gimenez, M. Scavuzzo, and J. Pucheta. A new approach to segmentation of multispectral remote sensing images based on MRF. *Geoscience and Remote Sensing Letters, IEEE*, 12(8):1720–1724, 2015.
- [51] J. Baumgartner, A. G. Flesia, J. Gimenez, and J. Pucheta. A new image segmentation framework based on two-dimensional hidden markov models. *Journal of Integrated Computer Aided Engineering*, 23:1–13, 2015. IOS Press.
- [52] J. Baumgartner, D. Aguilera, M. Scavuzzo, and J. Pucheta. A new approach to multisensor data fusion for precision agriculture based on Markov Random Fields. *Send to Computers and Electronics in Agriculture*, Elsevier, Octubre 2015.
- [53] A. G. Flesia, J. Baumgartner, J. Gimenez, and J. Martinez. Accuracy of MAP segmentation with hidden Potts and Markov mesh prior models via Path Constrained Viterbi Training, Iterated Conditional Modes and Graph Cut based algorithms. *Send to Pattern Recognition*, Elsevier, Julio 2015.
- [54] R. L. Dobrushin. The description of a random field by means of conditional probabilities and conditions of its regularity. *Theory of Probability and its Applications*, 13:197–224, 1968.
- [55] K. Abend, T. J. Harley, and L. N. Kanal. Classification of binary random patterns. *IEEE Transactions on Information Theory*, 11(4):538–544, 1965.
- [56] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, B 36(2):192–236, 1974.
- [57] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. <http://www.statslab.cam.ac.uk/~grg/books/hammfest/hamm-cliff.pdf>. Accedido: 19-10-2015, 1971.
- [58] L. A. Christopher and E. J. Delp. Comparing three-dimensional bayesian segmentations for images with low signal-to-noise ratio ($\text{snr} < 1$) and strong attenuation. *Journal of Electronic Imaging*, 23(4), 2014. Article number 043018.

- [59] O. Yousif and Y. Ban. Improving sar-based urban change detection by combining map-mrf classifier and nonlocal means similarity weights. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(10):4288–4300, 2014.
- [60] J. A. Gimenez. *Estimación de parámetros de modelos a priori para segmentación contextual de imágenes*. PhD thesis, Universidad Nacional de Córdoba; Facultad de Matemática, Astronomía y Física, 2014.
- [61] M. S. Kaiser, P. C. Caragea, and K. Furukawa. Centered parameterizations and dependence limitations in markov random field models. *Journal of Statistical Planning and Inference*, 142(7):1855–1863, 2012.
- [62] A. Poudyal. *Super resolution mapping for precision agriculture using VHR satellite imagery*. PhD thesis, University of Twente, 2013.
- [63] M. Rabiner and S. Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, 2007.
- [64] W.-B. Chen and C. Zhang. A hybrid framework for protein sequence clustering and classification using signature motif information. *Integrated Computer-Aided Engineering*, 16(4):353–365, 2009.
- [65] L. Tan and S. Xu. A model checking based approach to risk analysis in supply chain consolidations. *Integrated Computer-Aided Engineering*, 16(3):243–257, 2009.
- [66] J. Li, A. Najmi, and R. M. Gray. Image classification by a two dimensional hidden markov model. *IEEE Transactions on Signal Processing*, 48(2):517–533, 2000.
- [67] D. Joshi, J. Li, and J. Z. Wang. A computationally efficient approach to the estimation of two- and three-dimensional hidden markov models. *IEEE Transactions on Image Processing*, 15(7):1871–1886, 2006.
- [68] X. Ma, D. Schonfeld, and A. Khokhar. Image segmentation and classification based on a 2d distributed hidden markov model. In *Proc. SPIE 6822, Visual Communications and Image Processing*, 2008.
- [69] S. Yousefi, N. Kehtarnavaz, and Y. Cao. Computationally tractable stochastic image modelling based on symmetric markov mesh random fields. *IEEE Transactions on Image Processing*, 22(6):2192–2206, 2013.

- [70] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. In *Annals of Mathematical Statistics 1*, pages 164–171, 1970.
- [71] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [72] J. Li and R. Gray. *Image Segmentation and Compression using Hidden Markov Models*. Kluwer Academic Publishers, 2000.
- [73] C. Forbes, M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. Wiley, fourth edition, 2010.
- [74] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.
- [75] S. Bandyopadhyay and S. Saha. *Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications*. Springer, 2013.
- [76] H.C. Li, W. Hong, Y.R. Wu, and P.Z. Fan. On the empirical-statistical modeling of sar images with generalized gamma distribution. *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, 5(3):386–397, 2011.
- [77] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.
- [78] G. Marsaglia, W. Tsang, and J. Wang. Evaluating kolmogorov’s distribution. *Journal of Statistical Software*, 8(18):1–4, 2003.
- [79] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [80] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.
- [81] R. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48(1):106–109, 1952.

- [82] L. Sun, Y. F. Chang, and X. Cai. A discrete simulation of tumor growth concerning nutrient concentration. *International Journal of Modern Physics B: Condensed Matter Physics; Statistical Physics; Applied Physics*, 18(17-19):2651–2657, 2004.
- [83] S. Sanyal and J. A. Glazier. Viscous instabilities in flowing foams: A cellular potts model approach. *Journal of Statistical Mechanics*, 10, 2006.
- [84] Y. Yu, Z. Cao, and J. Feng. Continuous potts model based sar image segmentation by using dictionary-based mixture model. In B. Zhang, J. Mu, W. Wang, Q. Liang, and Y. Pi, editors, *The Proceedings of the Second International Conference on Communications, Signal Processing, and Systems*, volume 246 of *Lecture Notes in Electrical Engineering*, pages 577–585. Springer International Publishing, 2014.
- [85] M. Storath, A. Weinmann, J. Friel, and M. Unser. Joint image reconstruction and segmentation using the potts model. *Inverse Problems*, 31(2), 2015. paper no. 025003.
- [86] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [87] A. C. Frery, A. H. Correia, and C. C. Freitas. Classifying multifrequency fully polarimetric imagery with multiple sources of statistical evidence and contextual information. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3098–3109, 2007.
- [88] A. L. M. Levada, N. D. A. Mascarenhas, and A. Tannús. Pseudolikelihood equations for the Potts MRF model parameters estimation on higher order neighborhood systems. *IEEE Geoscience and Remote Sensing Letters*, 5(3):522–526, 2008.
- [89] J. Gimenez, A. C. Frery, and A. G. Flesia. Inference strategies for the smoothness parameter in the potts model. *Proc. of the IEEE International Geoscience and Remote Sensing Symposium*, 10(1):1–17, 2013.
- [90] J. A. Gimenez, A. C. Frery, and A. G. Flesia. When data do not bring information: A case study in markov random fields estimation. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 2015. In press.

- [91] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [92] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 839–846, 1998.
- [93] H. Peng, R. Rao, and S. A. Dianat. Multispectral image denoising with optimized vector bilateral filter. *IEEE Transactions on Image Processing*, 23(1):264–273, 2014.
- [94] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Science & Business Media, 2009.
- [95] M. S. Mahmoud and Y. Xia. *Networked Filtering and Fusion in Wireless Sensor Networks*. CRC Press, 2014.
- [96] F. E. White, Jr. Data Fusion Lexicon. Technical Panel, Technical Panel for C³, Data Fusion Sub-Panel, Joint Directors of Laboratories, San Diego, 1987. Naval Ocean Systems Center.
- [97] IEEE-GRSS Data Fusion Contest. <http://www.grss-ieee.org/community/technical-committees/data-fusion/data-fusion-contest>. Accedido: 08-08-2015.
- [98] B. V. Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1997.
- [99] F. Castanedo. A review of data fusion techniques. *The Scientific World Journal*, 2013, 2013. Article ID 704504, 19 pages.
- [100] J. Esteban, A. Starr, R. Willetts, P. Hannah, and P. Bryanston-Cross. A review of data fusion models and architectures: towards engineering guidelines. *Neural Computing & Applications*, 14(4):273–281, 2005.
- [101] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

- [102] G. Ligorio and A. M. Sabatini. A novel kalman filter for human motion tracking with an inertial-based dynamic inclinometer. *IEEE Transactions on Biomedical Engineering*, 62(8):2033–2043, 2015. Article number 7056449.
- [103] D. E. George and A. Unnikrishnan. On the divergence of information filter for multi sensors fusion. *Information Fusion*, 27(13):76–84, 2016.
- [104] H. Wang and C. Glennie. Fusion of waveform lidar data and hyperspectral imagery for land cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:1–11, 2015.
- [105] W. Li, H. Li, and H. Gu. Study of remote sensing imagery parallel segmentation based on block and processing chain strategy. In *2011 International Symposium on Image and Data Fusion (ISIDF)*, pages 1–4, 2011.
- [106] T. Idol, B. Haack, and R. Mahabir. Radar and optical remote sensing data evaluation and fusion; a case study for Washington, DC, USA. *International Journal of Image and Data Fusion*, 6(2):138–154, 2015.
- [107] L. Li, M. She, and H. Luo. Comparison on fusion algorithms of zy-3 panchromatic and multi-spectral images. *Transactions of the Chinese Society of Agricultural Engineering*, 30(16):157–165, 2014.
- [108] H. S. Mahmood. *Proximal soil sensors and data fusion for precision agriculture*. PhD thesis, Wageningen University, 2013.
- [109] F. A. Rodrigues, R. G. V. Bramley, and D. L. Gobbett. Proximal soil sensing for precision agriculture: Simultaneous use of electromagnetic induction and gamma radiometrics in contrasting soils. *Geoderma*, 243-244:183–195, 2015. Article number 7056449.
- [110] C. Cammalleri, M. C. Anderson, F. Gao, C. R. Hain, and W. P. Kustas. Mapping daily evapotranspiration at field scales over rainfed and irrigated agricultural areas using remote sensing data fusion. *Agricultural and Forest Meteorology*, 186:1–11, 2014.
- [111] C.-F. Chen, M. C. Valdez, N.-B. Chang, L.-Y. Chang, and P.-Y. Yuan. Monitoring spatiotemporal surface soil moisture variations during dry seasons in central america with multisensor cascade data fusion. *IEEE Journal of Selected Topics in*

- Applied Earth Observations and Remote Sensing*, 7(11):4340–4355, 2014. Article number 6899629.
- [112] C. H. Antink, C. Brüser, and S. Leonhardt. Detection of heart beats in multimodal data: A robust beat-to-beat interval estimation approach. *Physiological Measurement*, 36(8):1679–1690, 2015.
- [113] S. Grunwald, G. M. Vasques, and R. G. Rivero. Fusion of soil and remote sensing data to model soil properties. *Advances in Agronomy*, 131:1–109, 2015.
- [114] R. Casa, F. Castaldi, S. Pascucci, B. Basso, and S. Pignatti. Geophysical and hyperspectral data fusion techniques for in-field estimation of soil properties. *Vadose Zone Journal*, 12(4), 2013.
- [115] J. Shang, J. Liu, B. Ma, T. Zhao, X. Jiao, X. Geng, T. Huffman, J. M. Kovacs, and D. Walters. Mapping spatial variability of crop growth conditions using rapideye data in northern ontario, canada. *Remote Sensing of Environment*, 168:113–125, 2015.
- [116] A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231, 1959.
- [117] A. Srinivasan. *Handbook of Precision Agriculture: Principles and Applications*. CRC Press, 2006.
- [118] El gran desarrollo de los sensores de planta. <http://www.redagricola.com/reportajes/riego/el-gran-desarrollo-de-los-sensores-de-planta>. Accedido: 26-08-2015.
- [119] Humedad del suelo. http://agro.infoclima.com/?page_id=506. Accedido: 26-08-2015.
- [120] D. Tomkiewicz and T. Piskier. A plant based sensing method for nutrition stress monitoring. *Precision Agriculture*, 13(3):370–383, 2012.
- [121] R.F. Muñoz-Huerta, A. Ortiz-Melendez, R. G. Guevara-Gonzalez, I. Torres-Pacheco, G. Herrera-Ruiz, L. M. Contreras-Medina, J. Prado-Olivarez, and R.V. Ocampo-Velazquez. An analysis of electrical impedance measurements applied for plant n status estimation in lettuce (*lactuca sativa*). *Sensors*, 14(7):11492–11503, 2014.

- [122] K. Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.
- [123] C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72–101, 1904.
- [124] B. Johnson and Z. Xie. Classifying a high resolution image of an urban area using super-object information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83:40–49, 2013.
- [125] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei. Supervised spectral-spatial hyperspectral image classification with weighted markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 53(3):1490–1503, 2015.
- [126] V. Vijaya and G. J. Niveditha. Classification of COSMO SkyMed SAR Data Based on Coherence and Backscattering Coefficient. *International Journal of Computer Science and Information Technology*, 1(4):60–63, 2012.
- [127] M. Migliaccio, L. Mascolo, F. Nunziata, M. Sarti, and G. Mazzarella. Cosmoskymed hh/vv pingpong mode sar data to discriminate among sea, urban, and vegetated areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(7):2880–2894, 2014.
- [128] J. G. Flesia and A. G. Flesia. The influence of processing in the accuracy of measurements in indirect digitalized intra-oral radiographic imaging for forensic applications. *The Forensic Oral Pathology Journal*, 2(4):20–24, 2011.
- [129] Image databases. http://www.imageprocessingplace.com/root_files_V3/image_databases.htm. Accedido: 17-09-2015.
- [130] Public-domain test images for homeworks and projects. <http://homepages.cae.wisc.edu/~ece533/images/>. Accedido: 17-09-2015.
- [131] S. Cheung. Proof of hammersley-clifford theorem. http://www.vis.uky.edu/~cheung/courses/ee639/Hammersley-Clifford_Theorem.pdf. Accedido: 19-10-2015, 2008.