Revista de Matemática: Teoría y Aplicaciones 2016 **23**(1) : 143–154
cimpa – ucr issn: 1409-2433 (Print), 2215-3373 (Online)

CONDENSACIÓN CONTROLADA EN K-NN Y SU APLICACIÓN PARA LA IDENTIFICACIÓN DEL COLOR EN TIEMPO REAL

CONTROLLED CONDENSATION IN K-NN AND ITS APPLICATION FOR REAL TIME COLOR IDENTIFICATION

Carmen Villar-Patiño* Carlos Cuevas-Covarrubias†

Received: 4 Jun 2014; Revised: 23 Sep 2015;
Accepted: 30 Sep 2015

*Facultad de Ingeniería, Universidad Anáhuac, México. E-Mail: maria.villar@anahuac.mx
†Facultad de Ciencias Actuariales, Universidad Anáhuac, México. E-Mail: ccuevas@anahuac.mx

Resumen

Los algoritmos de vecinos cercanos (k-NN) son métodos ampliamente empleados en la clasificación estadística. Los cuales destacan por ser precisos y por no depender de ningún supuesto distribucional. A pesar de estas ventajas tienen el inconveniente de implicar un alto costo computacional. Conseguir formas eficientes de implementarlos es un reto importante para el desarrollo del reconocimiento de patrones. En este trabajo se discute una versión mejorada del algoritmo k-NN Condensación Controlada y se analiza su potencial en la identificación de color en tiempo real. Se basa en la representación de datos de entrenamiento en función de un conjunto reducido de prototipos informativos. Incluye dos parámetros que controlan el balance entre rapidez y precisión. Esto permite definir el porcentaje de condensación sin sacrificar demasiado la precisión del algoritmo. Probamos nuestra propuesta en un problema de clasificación instantánea en imágenes de video. Logramos la identificación de color en tiempo real mediante el algoritmo k-NN Condensación Controlada ejecutado con técnicas de programación multihilos. Los resultados obtenidos hasta el momento son alentadores.

Palabras clave: clasificación supervisada; vecinos cercanos; programación multihilos; condensación; selección de prototipos.

Abstract

k-NN algorithms are frequently used in statistical classification. They are accurate and distribution free. Despite these advantages, k-NN algorithms imply a high computational cost. To find efficient ways to implement them is an important challenge in pattern recognition. In this article, an improved version of the k-NN Controlled Condensation algorithm is introduced. Its potential for instantaneous color identification in real time is also analyzed. This algorithm is based on the representation of data in terms of a reduced set of informative prototypes. It includes two parameters to control the balance between speed and precision. This gives us the opportunity to achieve a convenient percentage of condensation without incurring in an important loss of accuracy. We test our proposal in an instantaneous color identification exercise in video images. We achieve the real time identification by using k-NN Controlled Condensation executed through multi-threading programming methods. The results are encouraging.

Keywords: supervised classification; nearest neighbours; multi-threading; condensation; prototype selection.

Mathematics Subject Classification: 62H30, 68U10.

1 Introducción

1.1 El color

El sistema visual humano, formado por el cerebro y los ojos, es el sistema de procesamiento de imágenes más efectivo que existe [1]. La mayor parte de la información que recibimos del entorno llega a través de él. El color es una de las características más evidentes y comunes de los objetos, que se puede pensar es fácilmente reconocible. Sin embargo, no es así, el color es una sensación que involucra factores físicos (una radiación electromagnética con longitud de onda entre 400nm. y 700nm.), fisiológicos (tres tipos de células fotoreceptoras localizadas en la retina, llamadas conos) y psicológicos (el cerebro procesa la información y define la percepción). En Coren et al. [2] se menciona que en 1965 la Oficina Estadounidense de Normas, tenía reconocidos 7500 nombres de colores.

Con base al estudio de las propiedades psicofísicas, se han desarrollado diversos modelos que buscan representar en un sistema coordenado a un color como un sólo punto. Dentro de estos modelos se encuentran el RGB y el CIELAB. El modelo RGB es el más común orientado a hardware, usado principalmente en dispositivos digitales de entrada, como las cámaras de video y dispositivos de salida como los monitores. El modelo CIELAB, creado por la *Commission Internationale de l'Eclairage*, la organización más importante responsable de estandarizar las métricas y términos usados en la ciencia del color, se caracteriza por definir una escala estándar de comparación que es independiente del dispositivo, además de incorporar la teoría de colores oponentes [5].

Entre 1984 y 2009, se publicaron más de 1000 reportes de investigación sobre color y textura, de acuerdo al estudio de Ilea y Whelan [8], lo cual nos da una idea de su importancia.

1.2 Algoritmo k-NN

Dada su facilidad de implementación, la regla de los k vecinos cercanos (k-NN, k-Nearest Neighbors) es uno de los clasificadores no paramétricos más usados. Sus propiedades teóricas garantizan que su probabilidad de error, está acotada por el doble de la probabilidad del error Bayesiano [4]. Dado un elemento ω de la población bajo estudio y una métrica de distancia, dicha regla identifica los k vecinos más cercanos en una base de entrenamiento, y asigna ω en la categoría con el máximo número de ejemplares. Es posible definir un umbral de decisión t, el cual especifica el número de elementos a partir del cual se clasifica a ω como perteneciente a una clase. En su forma más simple es un método de aprendizaje basado en casos, que conserva todos los datos de entrenamiento para clasificar, por lo que se le describe como un método del tipo “*lazy learning*” [6]. Sus tres limitantes más importantes en la implementación son [4]:

1. Requiere espacio de almacenamiento grande para la base de entrenamiento a partir de la cual se crea la regla de decisión.

2. Bajo rendimiento en la ejecución de la regla de decisión por calcular la medida de similaridad constantemente entre las bases de entrenamiento y prueba.

3. Baja tolerancia al ruido, especialmente en la regla del vecino cercano (*i.e.* cuando $k = 1$), al considerar todos los datos como relevantes.

Una técnica que enfrenta estos tres retos es la Reducción de Datos, también conocida como métodos de Selección de Instancias o Selección de Prototipos. Su objetivo, es obtener un conjunto de entrenamiento representativo de tamaño mucho menor que el original y con capacidad de predicción para nuevas instancias. En otras palabras, se tiene un conjunto T, compuesto de $M + N$ instancias $x_i = (x_{i1}, x_{i2}, ..., x_{id})$, que se divide en dos conjuntos: uno de entrenamiento (BE), compuesto por M instancias y uno de prueba (BP), compuesto por N instancias, se aplica un algoritmo de Selección a la BE para generar un conjunto llamado base condensada (BC), donde $BC \subset BE$, para clasificar a los elementos x_j de BP usando la regla k-NN [3]. Una recopilación y propuesta de taxonomía de los diferentes métodos de Selección de Instancias, se encuentra en el trabajo de García et al. [4].

2 k-NN condensación controlada

Inspirados por el algoritmo original propuesto por Guo et al. [6], *k-NN Model Based Approach* (MBA), Jímenez & Cuevas [9] proponen un algoritmo llamado k-NN Condensación Controlada (CC). El cual genera un modelo o base condensada de representantes (BCR), que es un conjunto de vectores de la forma $< r_i, Cls(r_i), Dis(r_i) >$, dónde: r_i es el vector de características del centro de la i-ésima región cubierta por el representante, $Cls(r_i)$ es la clase del grupo de k elementos representado por r_i, $Dis(r_i)$ es la medida del radio desde r_i a su k-ésimo vecino cercano. Dicho modelo se diferencia de una base condensada, por contener no sólo el vector de características y la clase, si no también la distancia de la región que cubre ese representante, utilizado para clasificar.
Jímenez y Cuevas trabajan el problema de 2 clases. Para definir el mejor valor de k, utilizan el área bajo la curva ROC. La clase del grupo de representados

por el elemento r_i se asigna por medio de un umbral t de decisión. Con base en $Num_{\Omega_1}(r_i)$, el número de elementos representados por r_i que pertenecen a Ω_1:

$$Cls(r_i) = \begin{cases} \Omega_0 & \text{si } Num_{\Omega_1}(r_i) < t \\ \Omega_1 & \text{si } Num_{\Omega_1}(r_i) \geq t. \end{cases} \tag{1}$$

Este algoritmo permite controlar el tamaño de la base condensada de representantes, lo cual es una ventaja cuando el tiempo de ejecución es importante. También permite definir una combinación adecuada de especificidad y sensibilidad.

El algoritmo para crear la BCR es el siguiente [9]:

1. Construir una matriz de distancias de los datos en la base de entrenamiento.
2. Etiquetar a todos sus elementos como “no agrupado”.
3. Encontrar, para cada dato, la bola (vecindad) con centro en ese dato que cubra a sus k vecinos más cercanos.
4. Encontrar el dato r_i cuya vecindad sea de radio mínimo. Definir y guardar al representante $\langle r_i, Cls(r_i), Dis(r_i)\rangle$ y etiquetar a sus elementos como “agrupado”. La clase de r_i se asigna con base en la regla de decisión mostrada en 1.
5. Repetir los pasos 3 y 4 hasta que todos los elementos de la muestra estén agrupados.

Es importante notar que:

- Al seleccionar el conjunto con el radio mínimo exista un empate, en ese caso se elegirá el conjunto más puro, es decir, aquel que contenga más elementos de una de las clases. Si el empate persiste, se hará una elección aleatoria.
- El último conjunto contiene los elementos sobrantes de todas las clases que juntos no aportan información útil. Por lo tanto se desecha. Entonces, el algortimo se detiene cuando quedan menos de k+1 elementos no agrupados.

Los puntos de la nota anterior no estaban contemplados en la propuesta original, son aportaciones de este trabajo.

Un ejemplo de la aplicación de este algoritmo con $M = 48$, $k = 4$ y $t = 1$ se muestra en la Figura 1. El primer cuadro muestra la base de entrenamiento, el

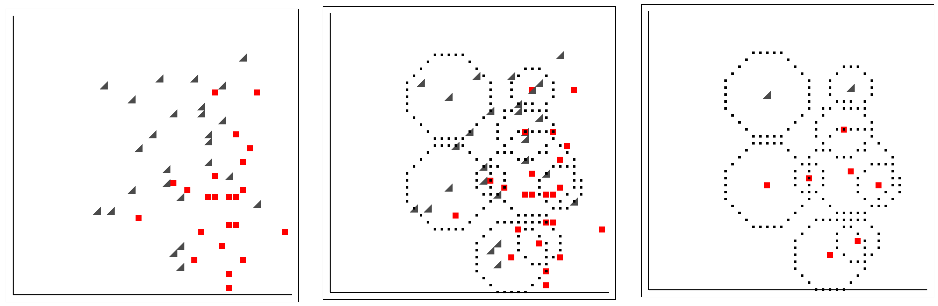

Figura 1: BE, su agrupación con el algoritmo k-NN CC y BCR obtenida.

segundo el algoritmo de condensación y el último la base condensada de representantes. El resultado son tan sólo 9 representantes.

El criterio de clasificación es el siguiente:

1. Para una nueva observación x por clasificar, calcular la distancia entre x y todos los r_i en la base condensada de representantes.

2. Si x está contenido unicamente en la región representada por r_j, i.e. la distancia de x a r_j es menor o igual a $Dis(r_j)$, clasificar x en $Cls(r_j)$.

3. Si x está contenido en al menos dos regiones de diferente categoría, clasificar x en la categoría del representante con el menor $Dis(r_j)$.

4. Si x no está contenido en alguna región, clasificar x en la categoría del representante con frontera más cercana.

Ejemplos de estos criterios se muestran en la Figura 2, dónde el individuo x se clasifica como «triángulo» por caer en una sola región, como «cuadro» por tener mayor cercanía a la frontera de un representante «cuadro» y como «triángulo» por tener esa región el radio más pequeño de entre las 2 regiones intersectadas.

Algunos criterios de clasificación permiten "poner en duda" la clase a la cual pertenece una observación [11]; generalmente, esta duda se resuelve con la aplicación de un segundo criterio. El algoritmo propuesto en este trabajo no contempla esta posibilidad, lo cual puede ser una debilidad en algunos contextos. Sin embargo, no es así en su aplicación para la identificación instantánea del color donde: por un lado, es necesario clasificar el 100% de los pixeles de cada imagen; y por el otro, emplear un segundo criterio nos alejaría del tiempo real. Si el algoritmo es usado en un contexto diferente, podría definirse la opción de duda siempre que una nueva observación caiga fuera de cualquier región en la base condensada de representantes; o bien, si la nueva observación está contenida en la intersección de dos regiones que sólo difieren por su categoría.

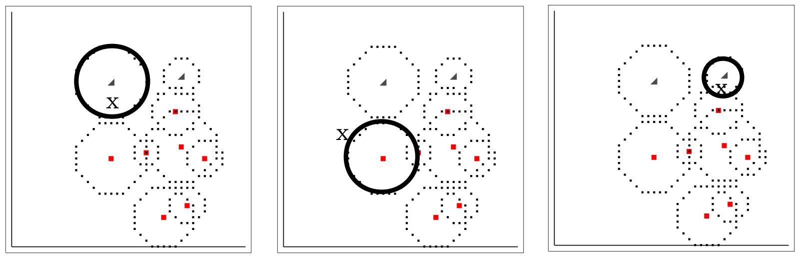

Figura 2: Posibilidades de clasificar a un individuo nuevo x con k-NN CC.

3 Experimentos

Los modelos de color usados fueron RGB, que requiere un vector de características con 3 elementos y el CIELAB que permite disminuir el efecto de la iluminación si se suprime la componente llamada L, que nos deja un vector de características con 2 elementos. Se tiene una base formada por colores de pixeles provenientes de imágenes en vídeo de garbanzos pintados y cuya distribución en el modelo CIELAB se aprecia en la Figura 3. Se tomó una muestra de 1000 pixeles de color Rojo, Verde y Azul y 500 pixeles, para cada uno de los colores restantes. Los colores para las pruebas de clasificación fueron elegidos de tal forma que fueran semejantes, para que se presentaran empalmes y complicar la prueba. Las combinaciones fueron las siguientes:

- Rojo.vs.No Rojo (Naranja, Rojo Ladrillo, Rosa, Rosa Mexicano y Fondo).
- Verde.vs.No Verde (Verde obscuro, Verde agua, Verde pasto y Fondo).
- Azul.vs.No Azul (Lila, Azul cielo y Fondo).

Para medir la precisión de los algoritmos se realizó una validación cruzada aleatoria con 50 repeticiones, con un 85% de los datos para la base de entrenamiento y el restante para prueba. Se compararon los algoritmos: k-NN tradicional, k-NN Model Based Approach y el k-NN Condensación Controlada con dos variantes: en la primera calibrado para alcanzar una condensación similar a la definida por el MBA, en la segunda calibrado para acercarse al tiempo real.

Las resultados obtenidos en cuanto al Error de clasificación, el área bajo la curva ROC [9], la sensibilidad y especificidad [10], así como el tamaño tanto en

Figura 3: Garbanzos pintados y su representación con CIELAB.

Tabla 1: Medidas rendimiento algoritmos con RGB.

kNN	Color	Error	A.C.ROC	Sen.	Esp.	BCR
Original	Rojo	7.00%	0.974	0.928	0.930	
	Verde	0.53%	0.999	0.994	0.995	
	Azul	4.61%	0.984	0.955	0.953	
MBA	Rojo	7.50%		0.804	0.949	121(95.2%)
	Verde	1.46%		0.964	0.995	18(99.2%)
	Azul	7.28%		0.861	0.949	71(95.8%)
CC=MBA	Rojo	8.15%	0.962	0.842	0.934	118(95.4%)
	Verde	4.14%	0.990	0.833	0.990	18(99.2%)
	Azul	8.21%	0.966	0.846	0.941	69(96.0%)
CC	Rojo	10.73%	0.945	0.804	0.910	29(98.9%)
	Verde	6.27%	0.981	0.937	0.937	14(99.4%)
	Azul	11.19%	0.943	0.801	0.917	26(98.5%)

número de elementos, como en porcentaje de condensación en la base condensada de representantes se muestran en la Tabla 1 para el modelo RGB y para el CIELAB en la Tabla 2.

Para medir la velocidad de la clasificación, se midieron los cuadros por segundo (fps, *frames per second*), cantidad directamente proporcional al tamaño de la ventana que captura el vídeo. En este caso fue una ventana de 640x480, lo cual implica clasificar en cada cuadro 307,200 pixeles. El objetivo es alcanzar al menos 15 fps para considerarlo tiempo real.

El algortimo de clasificación se programó en una computadora personal [1] explotando el procesamiento multihilos (*multi-threading*) [7], se dividió la imagen en secciones iguales, cada una enviada a un hilo de ejecución. Los resultados obtenidos, se muestran en la Figura 5.

[1]Intel core i7 860 a 2.8 Ghz., con 4 núcleos (8 procesadores lógicos), 2GB en RAM y sistema operativo Windows 7 de 32 bits.

Tabla 2: Medidas rendimiento algoritmos con CIELAB.

kNN	Color	Error	A.C.ROC	Sen.	Esp.	BCR
Original	Rojo	7.53%	0.969	0.924	0.925	
	Verde	1.51%	0.996	0.984	0.995	
	Azul	13.86%	0.927	0.866	0.863	
MBA	Rojo	7.90%		0.867	0.941	123(95.9%)
	Verde	1.46%		0.973	0.991	26(98.9%)
	Azul	15.01%		0.849	0.836	147(92.9%)
CC=MB	Rojo	7.56%	0.967	0.881	0.942	116(96.1%)
	Verde	2.12%	0.997	0.944	0.915	25(99.0%)
	Azul	14.76%	0.914	0.850	0.854	121(94.2%)
CC	Rojo	9.19%	0.959	0.836	0.936	27(99.1%)
	Verde	1.79%	0.997	0.970	0.988	14(99.4%)
	Azul	16.49%	0.907	0.840	0.832	25(98.8%)

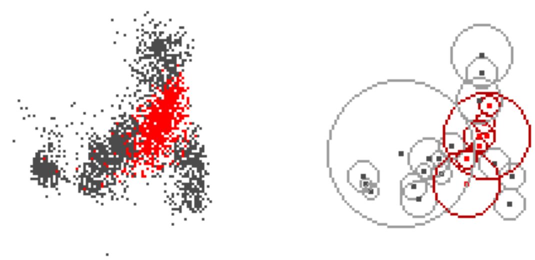

Figura 4: Problema Rojo .vs. No Rojo. Derecha Base de entrenamiento, Izquierda Base condensada de representantes.

La comparación gráfica del resultado de la clasificación, ocupando para el kNN CC la base condensada de representantes con la cual se obtuvo la mayor velocidad en el proceso, se muestra en las Figuras 4 y 6.

4 Conclusiones

Este trabajo ilustra con claridad cómo la identificación de color es una tarea compleja. Es posible que bajo ciertas condiciones de iluminación dos pixeles tengan el mismo vector de características pero diferente color. Sin embargo, en la mayoría de las imágenes se puede identificar a los garbanzos rojos, verdes y azules. Las distribuciones de los valores de los colores rojo y azul en el modelo CIELAB están empalmadas con las distribuciones de sus complementos. Los resultados coinciden con la evidencia presentada en [9]. En sus ejemplos con datos

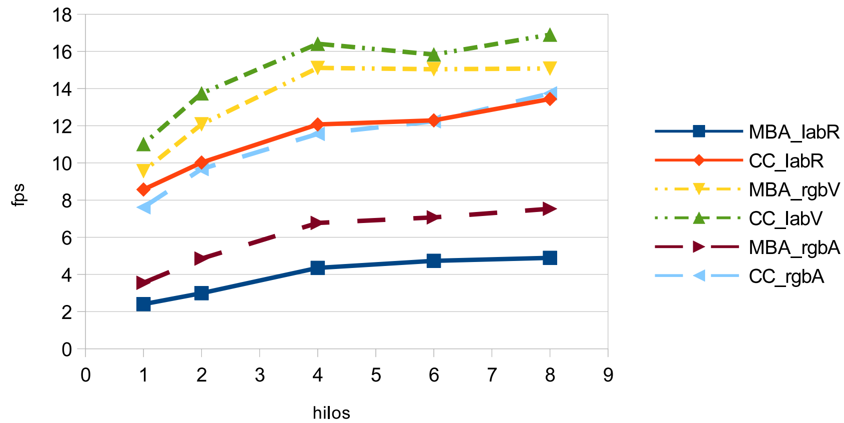

Figura 5: Rendimiento con programación multi-hilos.

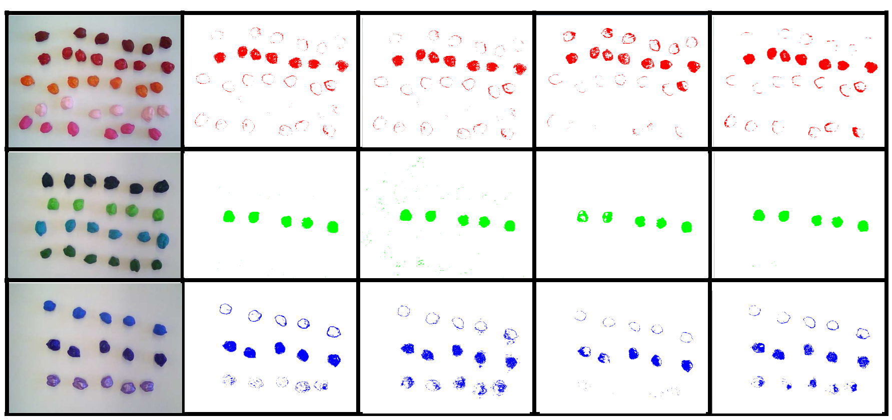

Figura 6: De izquierda a derecha: Video original, Clasificación con MBA y RGB, Clasificación con MBA y CIELAB, Clasificación con CC y RGB, Clasificación con CC y CIELAB.

simulados, el k-NN CC se desempeña mejor que el MBA cuando el empalme de las categorías es importante. Situación observada en nuestro experimento.

En los ejemplos discutidos en este trabajo, una mayor especificidad es deseable, el algoritmo k-NN CC, permite controlar esta característica para reducir la tasa de falsos positivos. La velocidad obtenida facilita la identificación de color en tiempo real, especialmente cuando se combina con la programación multihilos. Podemos decir que el k-NN controlado es competitivo en cuanto a su precisión y muy atractivo por su flexibilidad.

Se considera como trabajo futuro la programación del algoritmo en GPU (*Graphic Processing Units*), tanto en la clasificación como en el preprocesamiento requerido para determinar los valores de k y t. Otra meta es el analizar el problema de clasificación en 3 o más categorías. Para esto consideraremos un espacio de soluciones ampliado, que contemple opciones de duda y rechazo.

Referencias

[1] Artigas, J.M.; Capilla, P.; Felipe, A.; Pujol, J. (1995) *Óptica Fisiológica: Psicofísica de la Visión*. McGraw-Hill/Interamericana de España, Madrid.

[2] Coren, S.; Ward, L.M.; Enns, J.T. (2001) *Sensación y Percepción*. McGraw-Hill/Interamericana de México, México.

[3] García, S.; Derrac, J.; Cano, J. R. ; Herrera, F. (2010) "Prototype selection for nearest neighbor classification: Survey of methods", Technical Report T–4–2010–PS Methods, Soft Computing and Intelligent Information Systems Research Group, Universidad de Granada. Disponible en: `http://sci2s.ugr.es/sites/default/files/files/TematicWebSites/pr/T-4-2010-PSMethods.pdf`.

[4] Garcia, S.; Derrac, J.; Cano, J.R.; Herrera, F. (2012) "Prototype selection for nearest neighbor classification: Taxonomy and empirical study", *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34(3): 417–435.

[5] Gonzalez R. C.; Woods R. E. (2008) *Digital Image Processing*, 3rd Edition. Pearson Prentice Hall, Upper Saddle River NJ.

[6] Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. (2003) "KNN model-based approach in classification", in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Springer, Berlin: 986–996.

[7] Hughes, C.; Hughes T. (2008) *Professional Multicore Programming: Design and Implementation for C++ Developers*. Wrox, Indiana.

[8] Ilea, D.E.; Whelan, P.F. (2011) "Image segmentation based on the integration of colour-texture descriptors-A review", *Pattern Recognition* 44(10): 2479–2501.

[9] Jiménez, R.; Cuevas, C. (2011) "Curvas ROC y vecinos cercanos, propuesta de un nuevo algoritmo de condensación", *Revista de Matemática: Teoría y Aplicaciones* 18(1): 21–32.

[10] Lalkhen, A.G.; McCluskey, A. (2008) "Clinical tests: sensitivity and specificity", *Continuing Education in Anaesthesia, Critical Care & Pain* 8(6): 221–223.

[11] Ripley B.D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.