DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Clustering basado en densidad

© Fernando Berzal, berzal@acm.org

Clustering basado en densidad

- Métodos basados en densidad
- DBSCAN
- OPTICS
- EnDBSCAN
- DENCLUE
- SNN Clustering

Métodos de agrupamiento

Familias de algoritmos de clustering:

- **Agrupamiento por particiones**
  k-Means, PAM/CLARA/CLARANS, BFR
- **Métodos basados en densidad**
  DBSCAN, Optics, DenClue
- **Clustering jerárquico**
  Diana/Agnes, BIRCH, CURE, Chameleon, ROCK

...

Métodos basados en densidad

- Un cluster en una región densa de puntos, separada por regiones poco densas de otras regiones densas.
- Útiles cuando los clusters tienen formas irregulares, están entrelazados o hay ruido/outliers en los datos.

Métodos basados en densidad

Criterio de agrupamiento local:
Densidad de puntos

Regiones densas de puntos separadas
de otras regiones densas por regiones poco densas.

Características

- Identifican clusters de formas arbitrarias.
- Robustos ante la presencia de ruido.
- Escalables: Un único recorrido del conjunto de datos

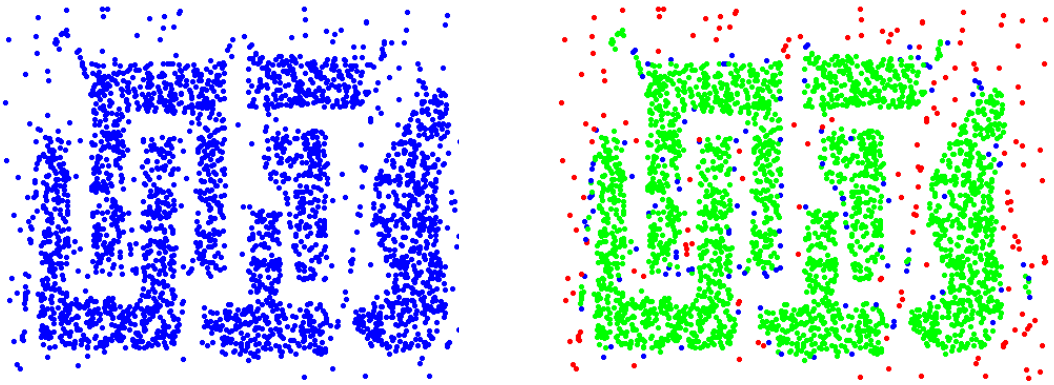
Métodos basados en densidad

Algoritmos basados en densidad

- **DBSCAN**: Density Based Spatial Clustering of Applications with Noise (Ester et al., KDD'1996)
- **OPTICS**: Ordering Points To Identify the Clustering Structure (Ankerst et al. SIGMOD'1999)
- **DENCLUE**: DENsity-based CLUstEring (Hinneburg & Keim, KDD'1998)
- **CLIQUE**: Clustering in QUEst (Agrawal et al., SIGMOD'1998)
- **SNN** (Shared Nearest Neighbor) density-based clustering (Ertöz, Steinbach & Kumar, SDM'2003)

DBSCAN

Detecta regiones densas de puntos separadas de otras regiones densas por regiones poco densas:



Parámetros: Epsilon = 10, MinPts = 4

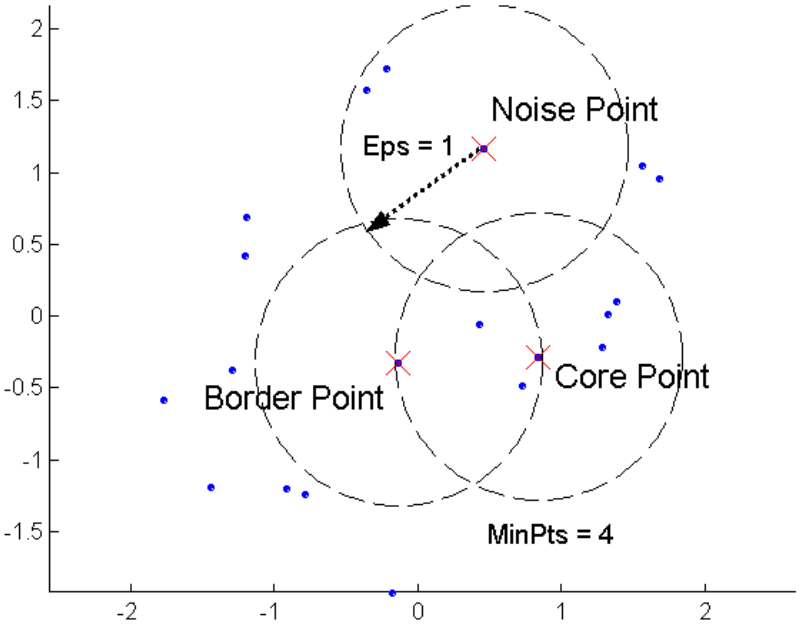
Puntos: **core** (cluster), **border** (frontera) y **noise** (ruido)

Eficiencia: **O(n log n)**

DBSCAN

- Densidad: Número de puntos en un radio específico (parámetro Epsilon)
- Puntos "**core**": Puntos interiores de un cluster (cuando tienen, al menos, un número mínimo de puntos MinPts en su vecindario de radio Epsilon)
- Puntos "**border**" (frontera): Tienen menos de MinPts puntos en su vecindario de radio Epsilon, estando en el vecindario de algún punto "core".
- Ruido (**noise**): Cualquier punto que no forma parte de un cluster ("core") ni está en su frontera ("border").

DBSCAN



DBSCAN

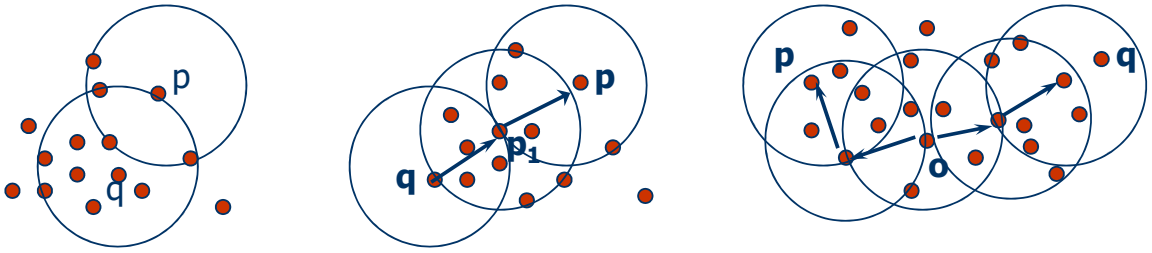
Algoritmo

- Se elimina el ruido del conjunto de datos
- Se agrupan los datos restantes:

current_cluster_label $\leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    current_cluster_label $\leftarrow$ *current_cluster_label* $+ 1$
    Label the current core point with cluster label *current_cluster_label*
  end if
  for all points in the *Eps*-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label *current_cluster_label*
    end if
  end for
end for

DBSCAN

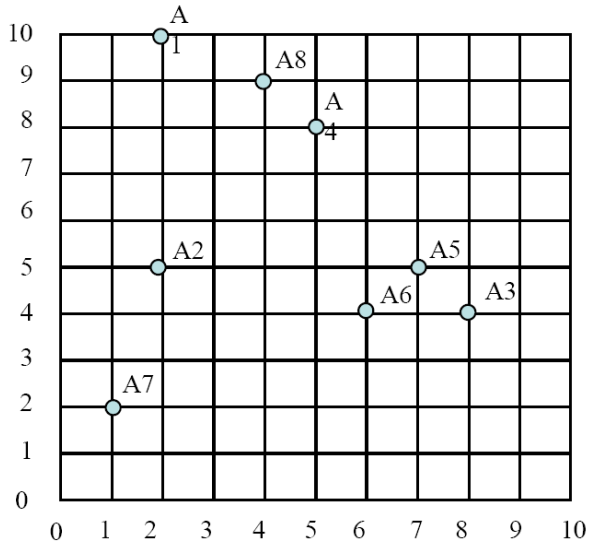
¿Cómo se detectan clusters de formas arbitrarias?



DBSCAN

Ejercicio

Agrupar los 8 puntos de la figura utilizando el algoritmo DBSCAN.



Número mínimo de puntos en el "vecindario":

MinPts = 2

Radio del "vecindario":

Epsilon $\sqrt{2} \triangleright \sqrt{10}$

DBSCAN

Ejercicio resuelto

Distancia euclídea

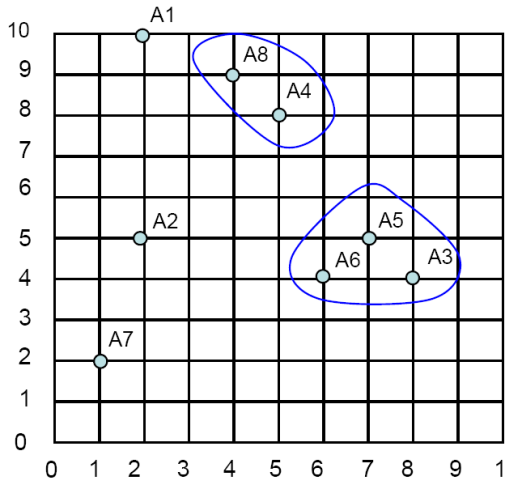
	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

DBSCAN

Ejercicio resuelto

Epsilon = $\sqrt{2}$

A1, A2 y A7 no tienen vecinos en su vecindario,
por lo que se consideran "outliers" (no están en zonas densas):

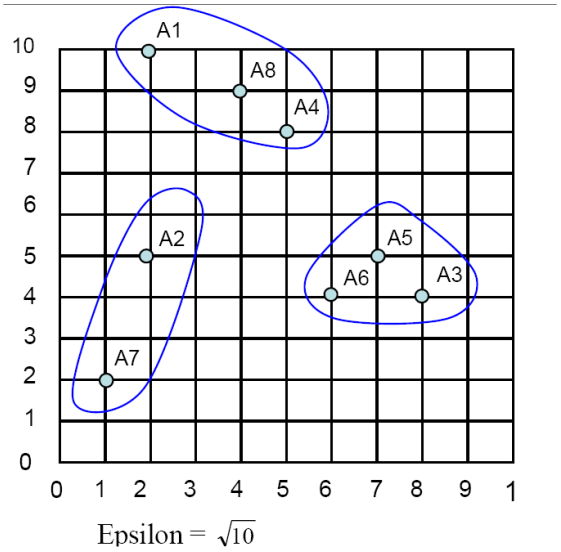


DBSCAN

Ejercicio resuelto

Epsilon = $\sqrt{10}$

Al aumentar el valor del parámetro Epsilon,
el vecindario de los puntos aumenta y todos quedan agrupados:

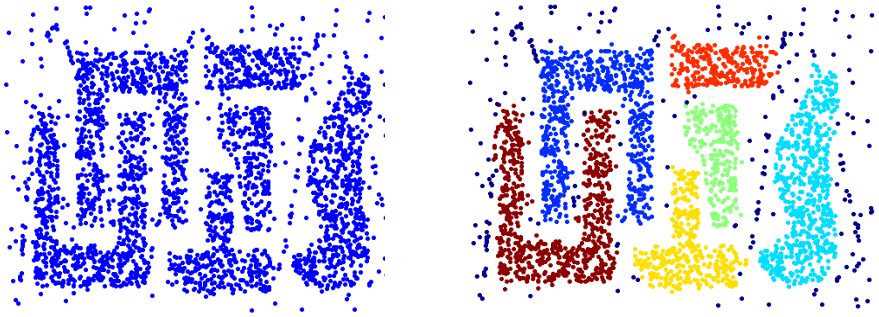


DBSCAN

DEMO

http://webdocs.cs.ualberta.ca/~yaling/Cluster/Applet/Code/Cluster.html

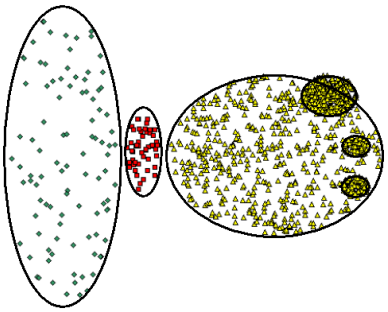
DBSCAN



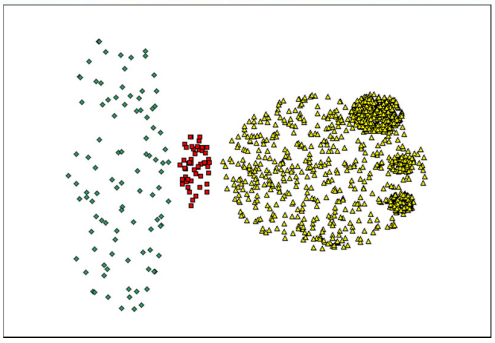
Clusters

DBSCAN… cuando funciona bien :-)

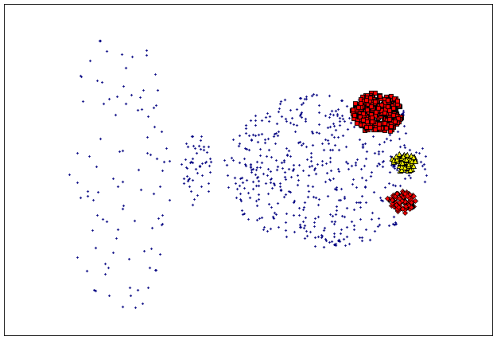
DBSCAN



Datos originales



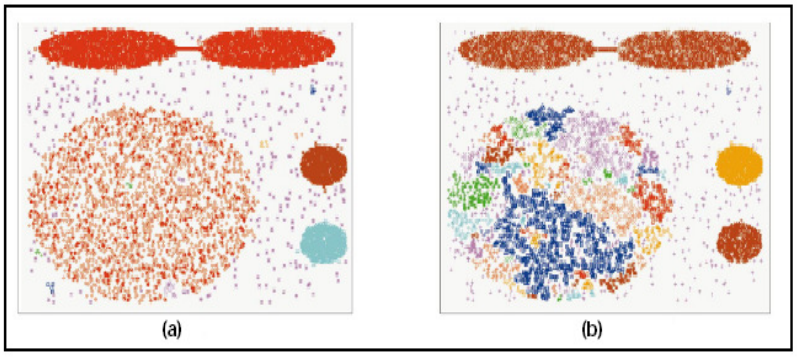
MinPts=4
Epsilon=9.75

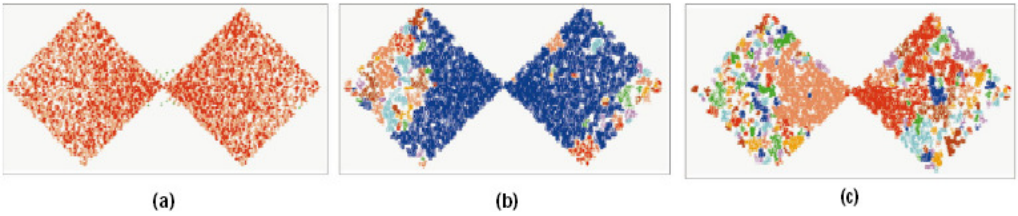


MinPts=4
Epsilon=9.92

DBSCAN… cuando no funciona :-(

DBSCAN





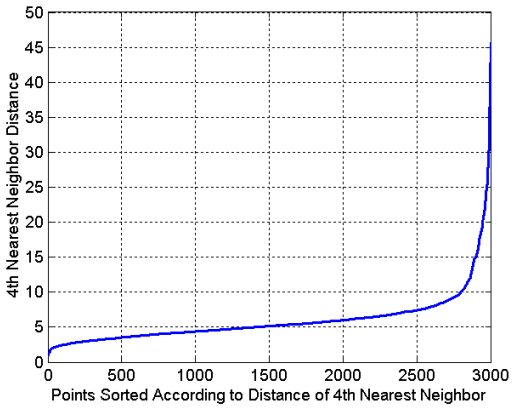
DBSCAN… cuando no funciona :-(

DBSCAN

Determinación de los parámetros (Epsilon & MinPts)

- En los puntos de un mismo cluster, su k-ésimo vecino debería estar más o menos a la misma distancia.
- En los puntos de ruido, su k-ésimo vecino debería estar más lejos.

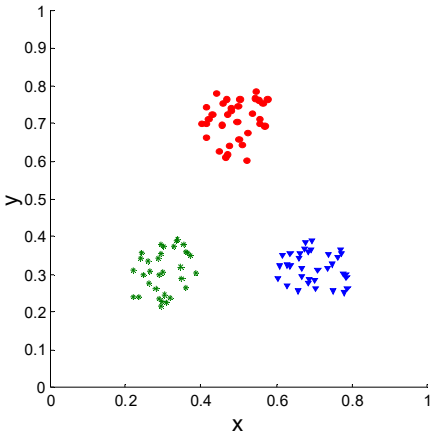
Posible solución:
Ordenar la distancia de todos los puntos a su k-ésimo vecino

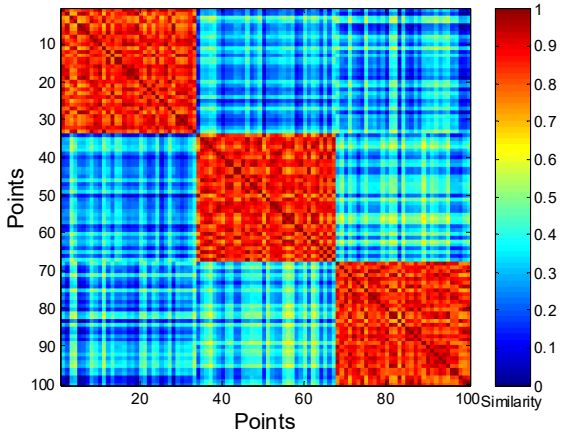


DBSCAN

Validación de resultados

Matriz de similitud

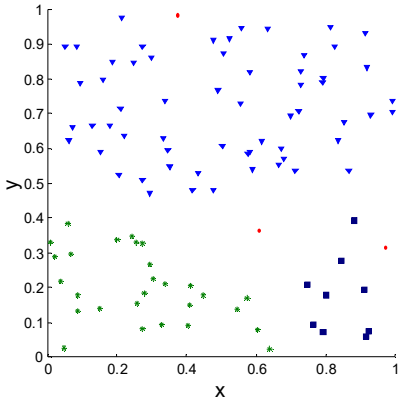


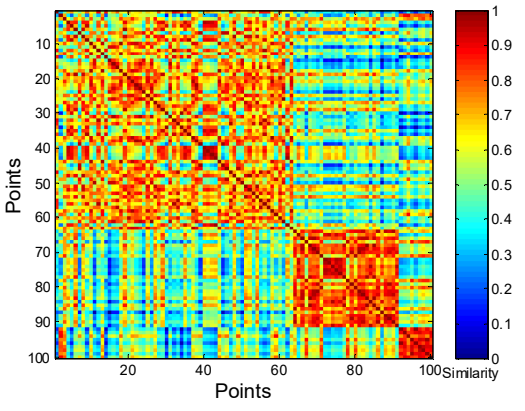


DBSCAN

Validación de resultados

Matriz de similitud

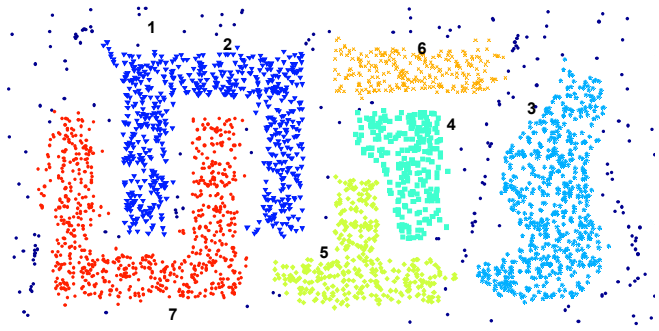


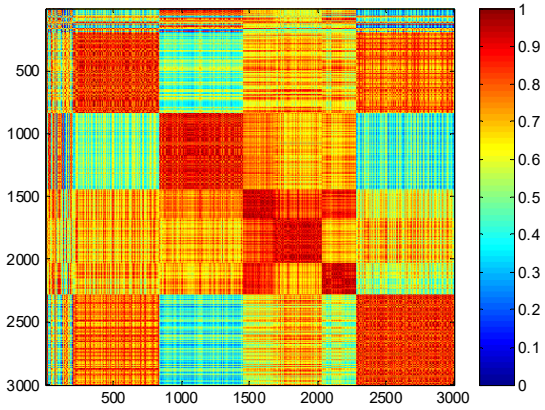


DBSCAN

Validación de resultados

Matriz de similitud





OPTICS

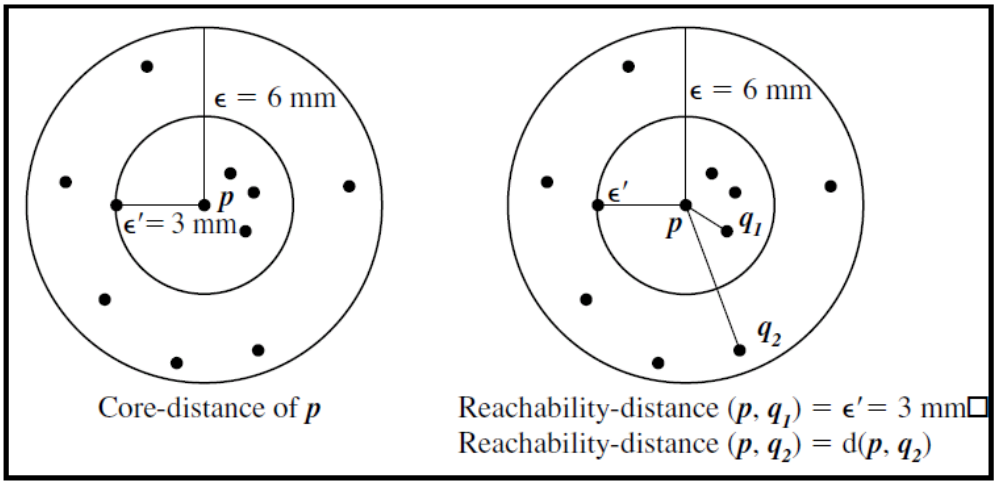
Ordering Points To Identify the Clustering Structure

Ankerst, Breunig, Kriegel & Sander (SIGMOD'99)

- Ordena los datos de acuerdo a la estructura de sus clusters: puntos cercanos espacialmente acaban siendo vecinos en la ordenación.
- Esta ordenación es equivalente a los clusters correspondientes a un amplio abanico de parámetros de un algoritmo de clustering basado en densidad.
- Se puede representar gráficamente (análisis de clusters tanto automático como interactivo).

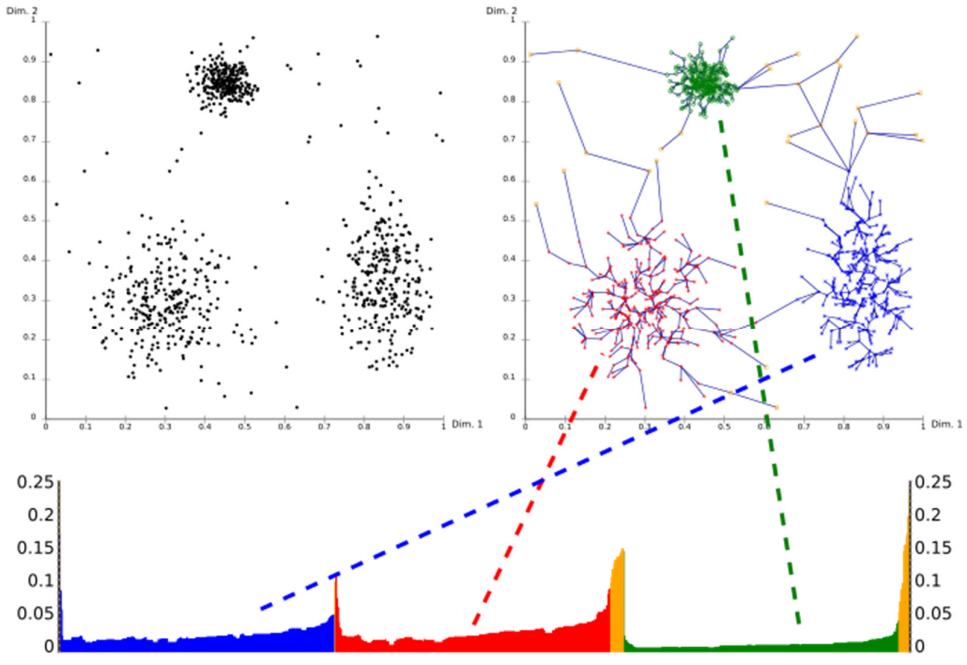
OPTICS

Eficiencia: **O(n log n)**



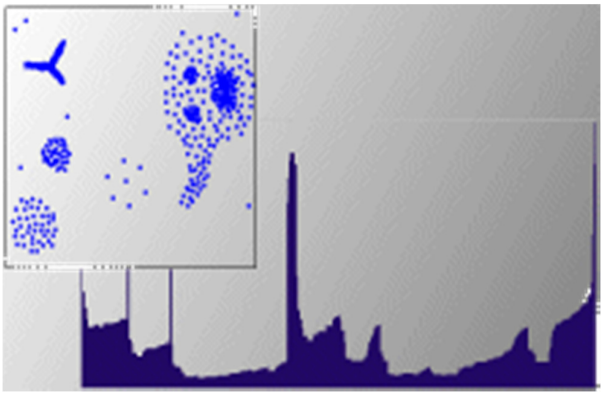
Core distance & Reachability distance

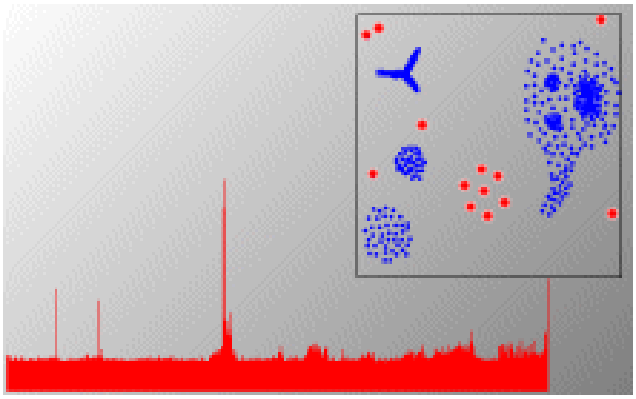
OPTICS



Reachability plot (~ dendrograma)

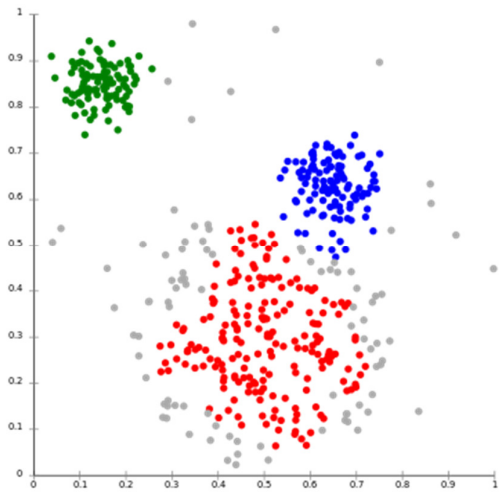
OPTICS

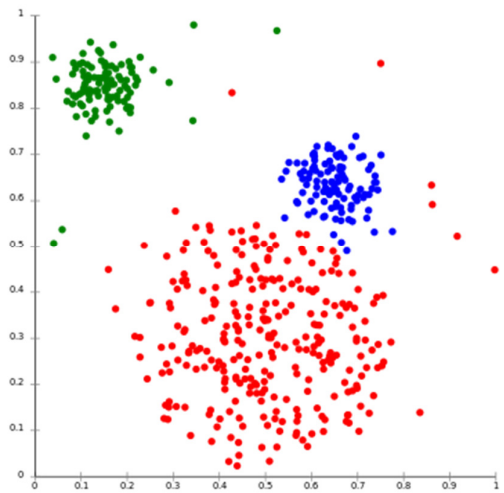




http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/OPTICS/Demo/

OPTICS



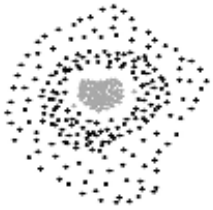


DBSCAN **OPTICS**

Mejor que DBSCAN con clusters de distinta densidad

EnDBSCAN

Extensión de DBSCAN capaz de detectar clusters anidados en espacios de densidad variable.

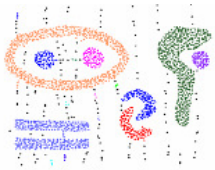
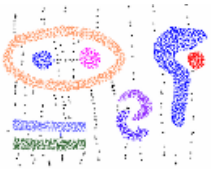
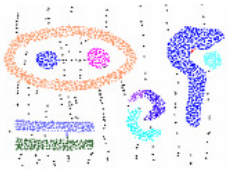
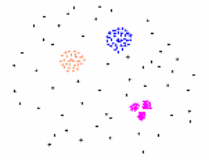
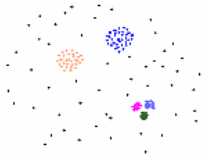
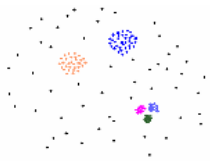
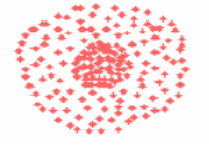
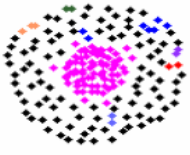
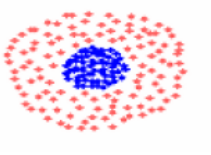


IDEA

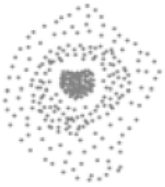
Los puntos vecinos dentro del núcleo de un cluster deben formar una región uniformemente densa.

Aplicaciones: MRI [Magnetic Resonance Imaging]

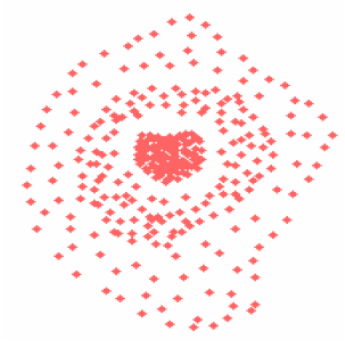
EnDBSCAN

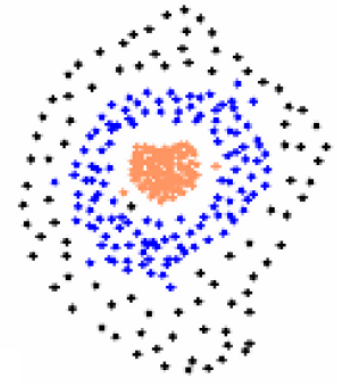
DBSCAN	**OPTICS**	**EnDBSCAN**	
			OK
			DBSCAN falla
			DBSCAN & OPTICS fallan

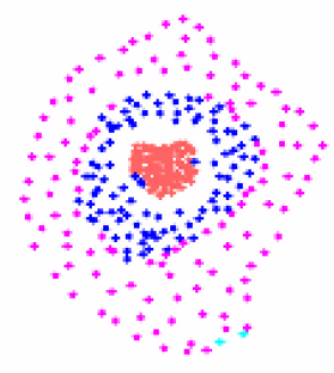
EnDBSCAN



DBSCAN **OPTICS** **EnDBSCAN**







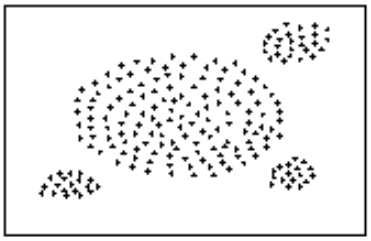
DENCLUE

DENsity-based CLUstEring

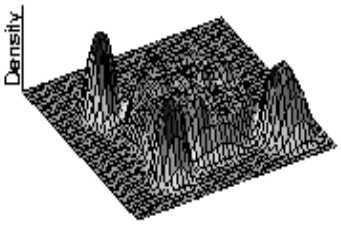
Hinneburg & Keim (KDD'98)

- Base matemática sólida.
- Funciona bien en conjuntos de datos con ruido.
- Permite una descripción compacta de clusters de formas arbitrarias en conjuntos de datos con muchas dimensiones.
- Más rápido que otros algoritmos (p.ej. DBSCAN)
- … pero necesita un número elevado de parámetros.

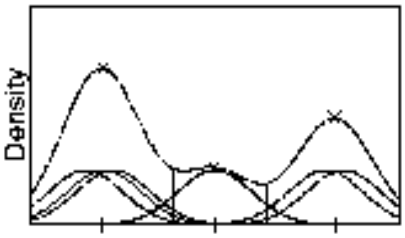
DENCLUE



(a) Data Set



(c) Gaussian

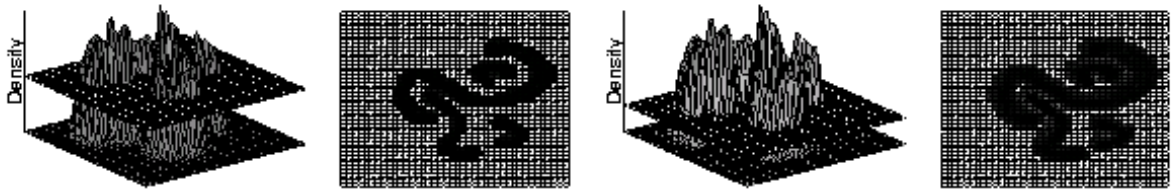


Density attractors (local maxima of the density function)

DENCLUE

(a) $\sigma = 0.2$ (b) $\sigma = 0.6$ (d) $\sigma = 1.5$

Figure 3: Example of Center-Defined Clusters for different σ



(a) $\xi = 2$ (b) $\xi = 2$ (c) $\xi = 1$ (d) $\xi = 1$

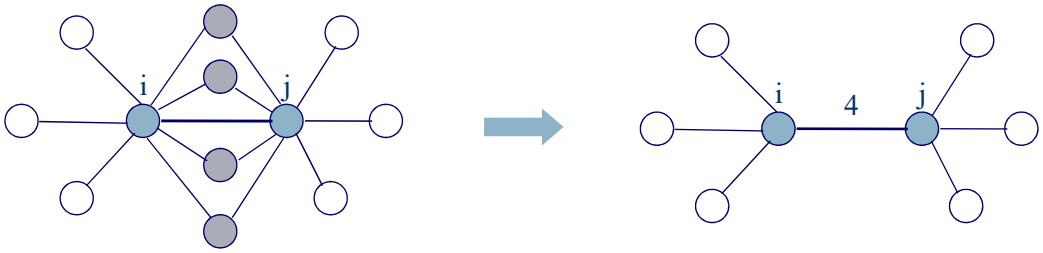
Figure 4: Example of Arbitray-Shape Clusters for different ξ

Clusters de formas arbitrarias combinando atractores de densidad conectados por caminos densos.

SNN Clustering

Número de vecinos compartidos

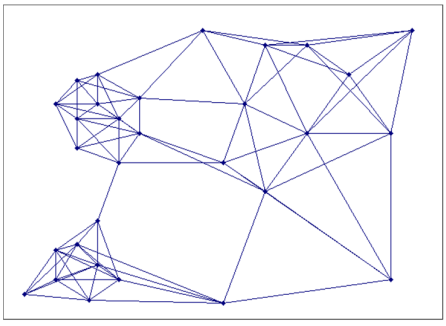
Una medida de similitud más:



SNN Clustering

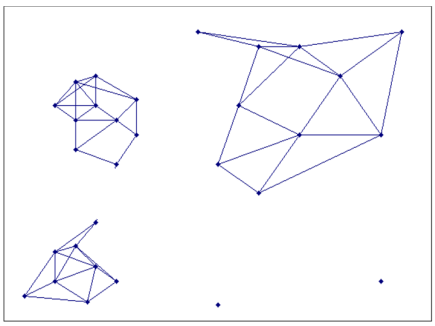
SNN Graph

Grafo de vecinos compartidos



Grafo ponderado

(similitudes entre puntos vecinos)



Grafo SNN

(peso de las aristas = número de vecinos compartidos)

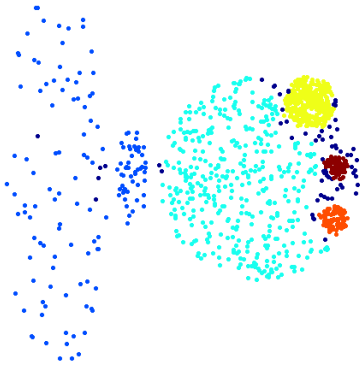
Jarvis-Patrick algorithm

- Se encuentran los k vecinos más cercanos de cada punto (esto es, se eliminan todos menos los k enlaces más fuertes de cada punto).
- Un par de puntos está en el mismo cluster si comparten más de T vecinos y cada punto están en la lista de k vecinos del otro punto.

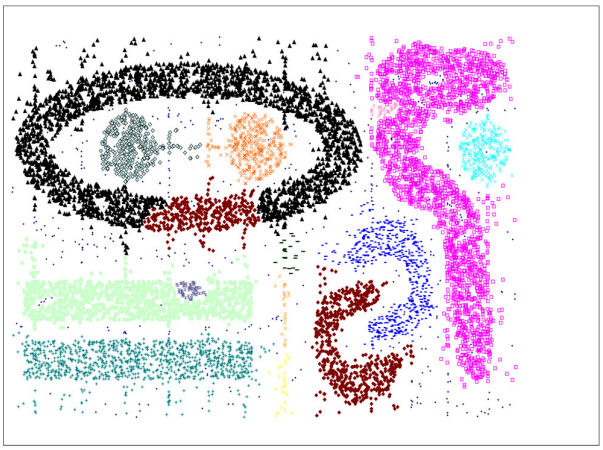
Algoritmo frágil

6 vecinos compartidos de 20

OK!

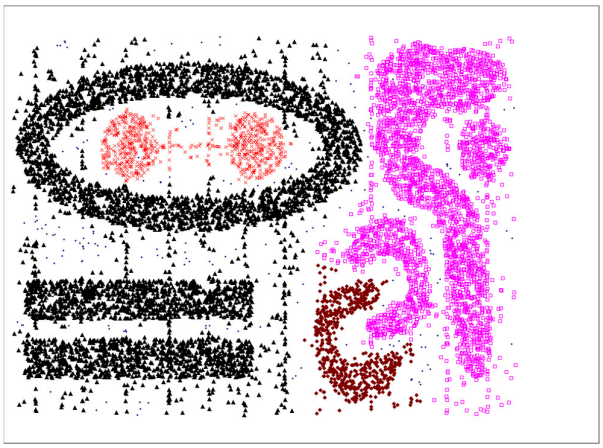


Jarvis-Patrick algorithm



Umbral T más bajo

que no mezcla clusters



Umbral T-1

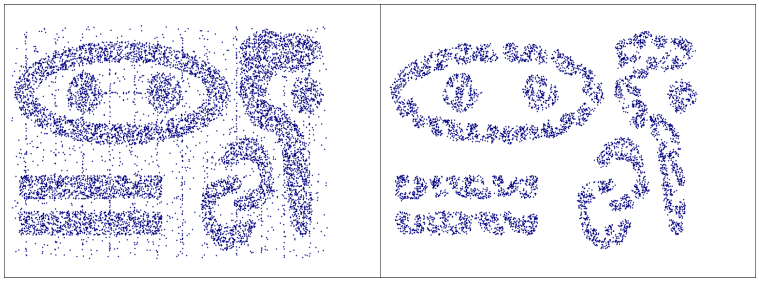
SNN Clustering

- Se calcula la matriz de similitud y nos quedamos sólo con los k vecinos más similares, a partir de los cuales construimos el grafo de vecinos compartidos SNN.

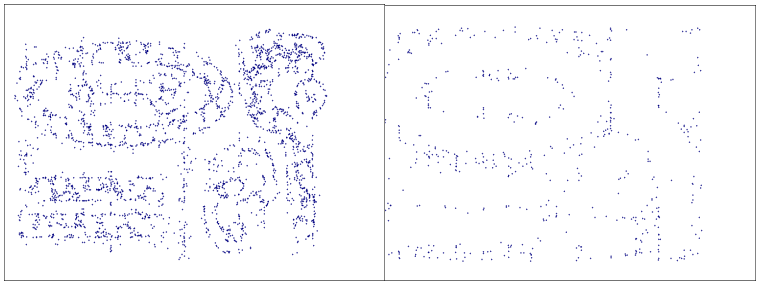
- Calculamos la **densidad SNN** de cada punto (usando un parámetro Eps dado por el usuario, el número de puntos que tienen una similitud igual o mayor que Eps para cada punto).

SNN Clustering

Densidad SNN



Puntos originales Densidad alta



Densidad media Densidad baja

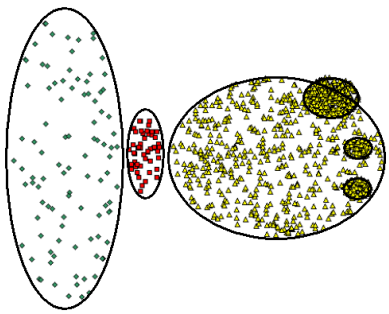
SNN Clustering

- Encontramos los puntos "core" (todos los puntos con densidad SNN mayor que MinPts, otro parámetro dado por el usuario).
- Agrupamos los puntos "core" (dos puntos a distancia Eps o menor se colocan en el mismo cluster).
- Descartamos todos los puntos no-"core" que estén a distancia mayor que Eps de un punto "core" (los consideramos ruido).
- Asignamos los puntos restantes al cluster del "core" más cercano.

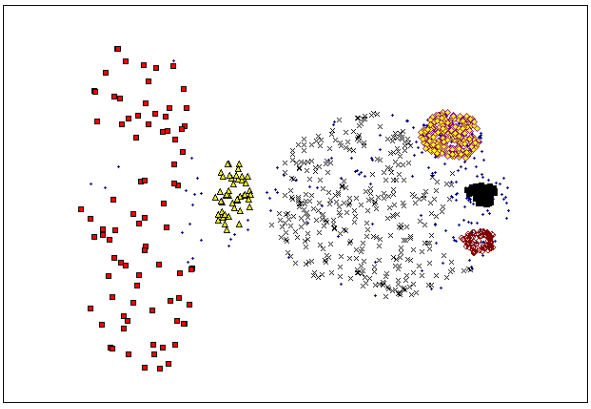
NOTA: Son los mismos pasos del algoritmo DBSCAN…

SNN Clustering

Funciona con clusters de distinta densidad



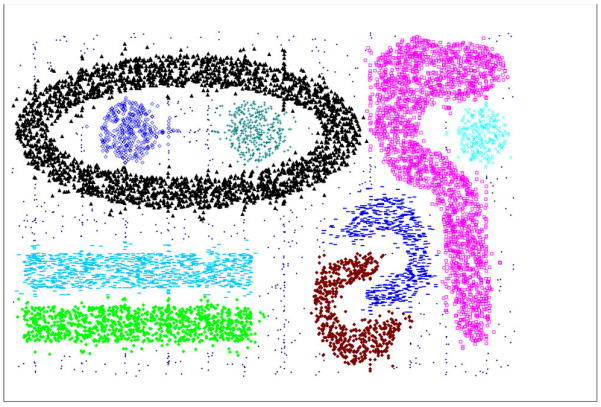
Puntos originales

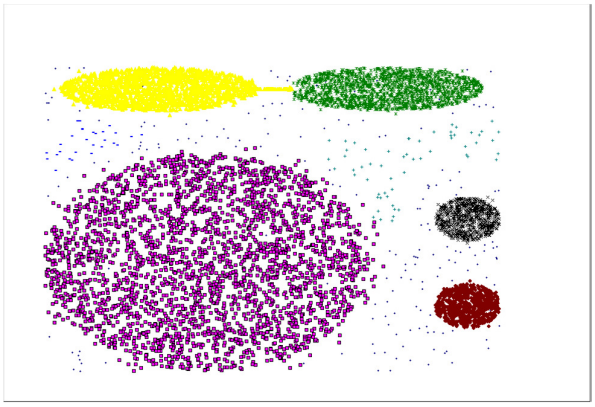


SNN Clustering

SNN Clustering

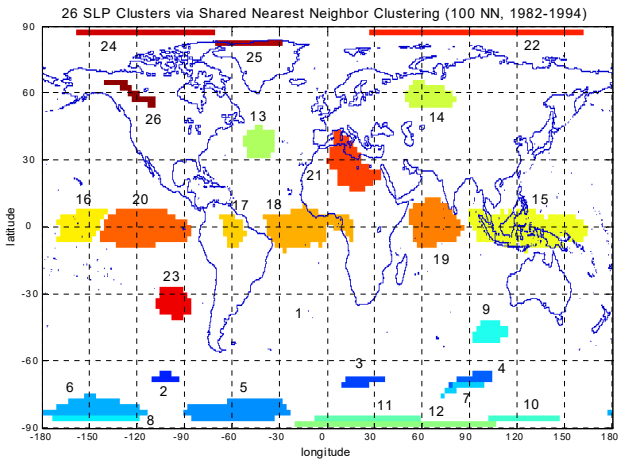
Detecta clusters de formas complejas

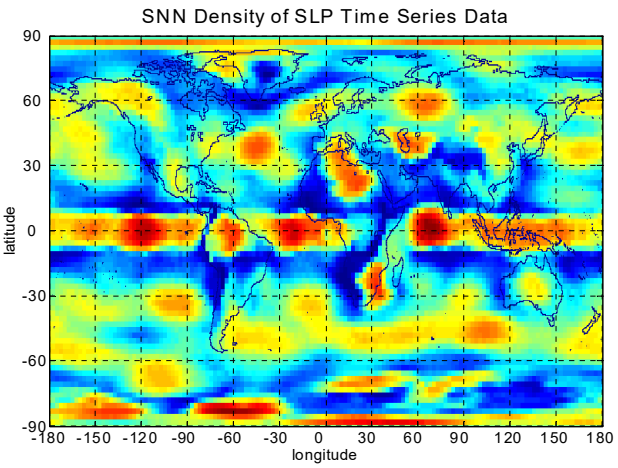




Aplicaciones

Datos espacio-temporales

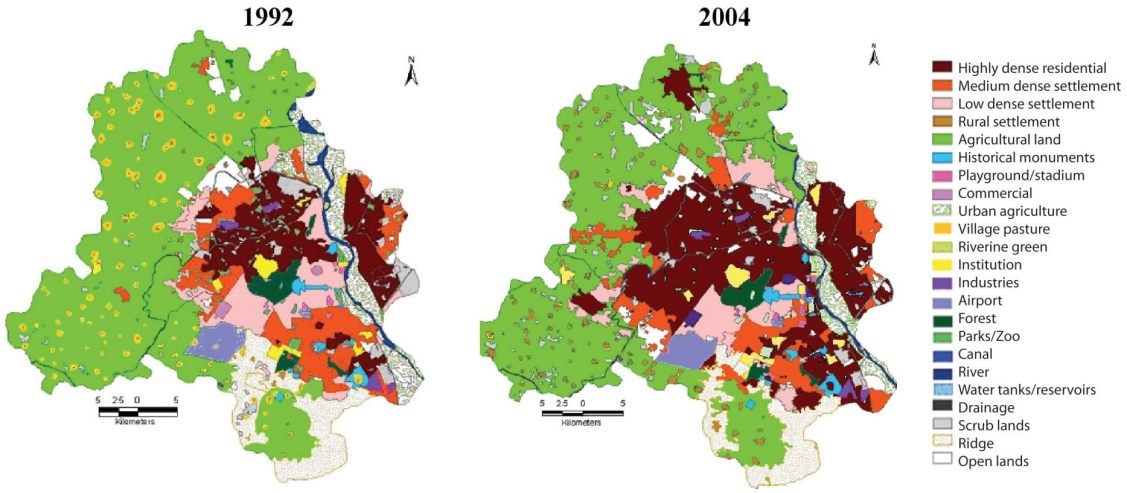




SNN Clustering

Aplicaciones

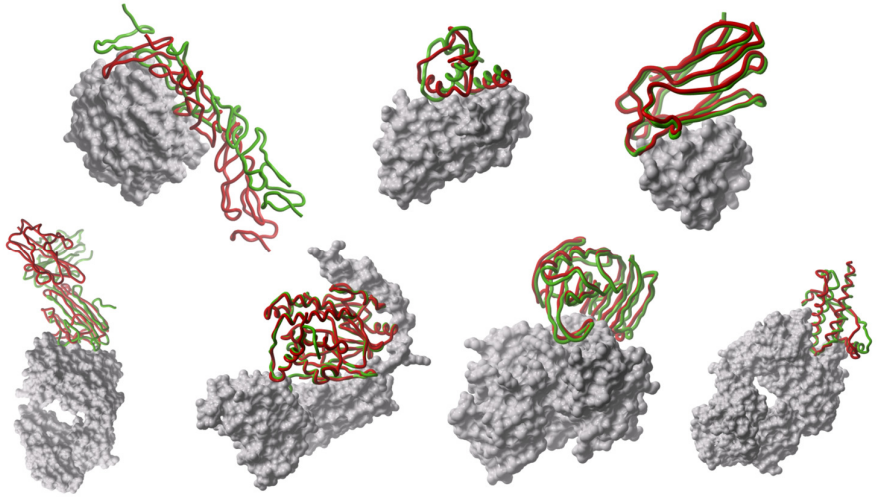
Detección del uso del terreno



DBSCAN

Aplicaciones

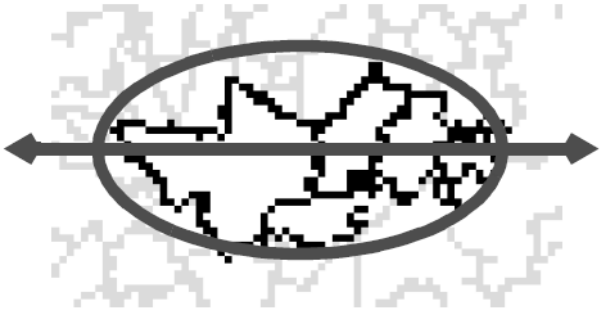
Potential protein-docking sites

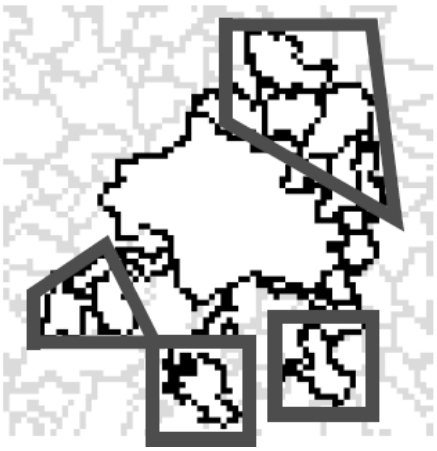


GDBSCAN

Aplicaciones

Influence regions in GIS

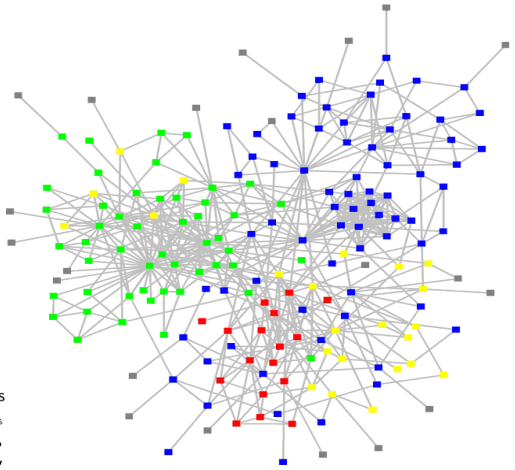




GDBSCAN

Aplicaciones

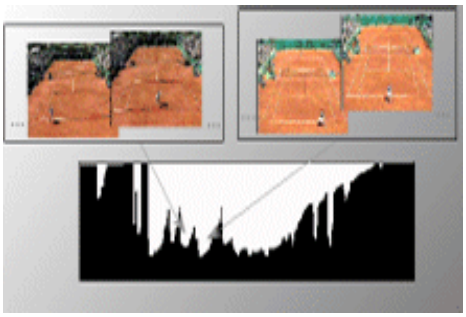
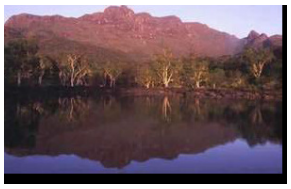
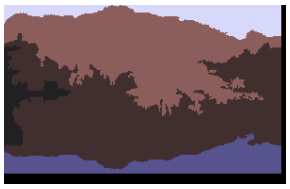
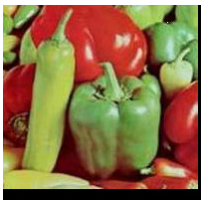
Web user profiling
(clustering web log sessions)

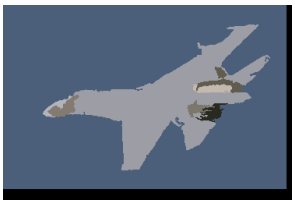


Aplicaciones

Clustering de imágenes
(p.ej. histogramas de color)



Eficiencia

O (n * tiempo necesario para encontrar los vecinos a distancia Epsilon)

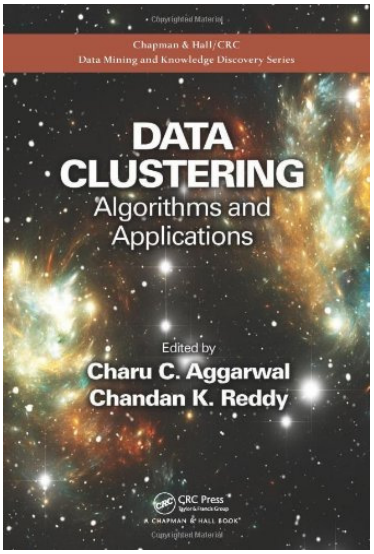
- En el peor caso, **O (n^2)**
- Si no hay demasiadas dimensiones, se pueden emplear estructuras de datos para encontrar los vecinos más cercanos, p.ej. R*trees, k-d trees…

O (n log n)

- También se pueden paralelizar…

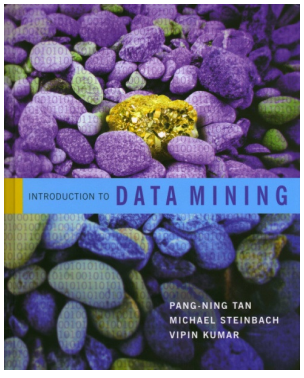
Bibliografía

- Charu C. Aggarwal & Chandan K. Reddy (editors): **Data Clustering: Algorithms and Applications**. Chapman & Hall / CRC Press, 2014. ISBN 1466558210.

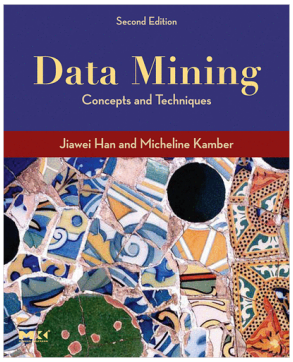


Bibliografía

- Pang-Ning Tan, Michael Steinbach & Vipin Kumar: **Introduction to Data Mining** Addison-Wesley, 2006. ISBN 0321321367 [capítulos 8&9]



- Jiawei Han & Micheline Kamber: **Data Mining: Concepts and Techniques** Morgan Kaufmann, 2006. ISBN 1558609016 [capítulo 7]



Bibliografía - Algoritmos

DBSCAN

- Martin Ester, Hans-Peter Kriegel, Jörg Sander & Xiaowei Xu: **A density-based algorithm for discovering clusters in large spatial databases with noise**. Proceedings KDD-96. AAAI Press, pp. 226–231. ISBN 1-57735-004-9. CiteSeerX: 10.1.1.71.1980

OPTICS

- Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriege & Jörg Sander: **OPTICS: ordering points to identify the clustering structure**. Proceedings ACM SIGMOD'99, pp. 49-60. DOI 10.1145/304182.304187

EnDBSCAN

- S. Roy & D. K. Bhattacharyya: **An Approach to Find Embedded Clusters Using Density Based Techniques**, ICDCIT'2005, pp. 523–535, 2005. DOI 10.1007/11604655_59

DENCLUE

- Alexander Hinneburg & Daniel A. Keim: **An efficient approach to clustering in large multimedia databases with noise**. Proceedings KDD'98, pp. 58-65. AAAI Press (1998).
- Alexander Hinneburg & Daniel A. Keim: **A General Approach to Clustering in Large Databases with Noise**. Knowledge and Information Systems, Volume 5, Issue 4, pp. 387-415, November 2003. DOI 10.1007/s10115-003-0086-9

Bibliografía - Algoritmos

SNN Clustering

- Levent Ertöz, Michael Steinbach & Vipin Kumar. **Finding clusters of different sizes, shapes, and densities in noisy, high-dimensional data**, SIAM International Conference on Data Mining (SDM'2003), pages 47-58.

Parallel density-based algorithms

- Stefan Brecheisen, Hans-Peter Kriegel & Martin Pfeifle: **Parallel density-based clustering of complex objects**. Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06), pp. 179-188. DOI 10.1007/11731139_22
- Yaobin He, Haoyu Tan, Wuman Luo, Huajian Mao, Di Ma, Shengzhong Feng & Jianping Fan: **MR-DBSCAN: An Efficient Parallel Density-Based Clustering Algorithm Using MapReduce**. ICPADS'2011, pp. 473-480. DOI 10.1109/ICPADS.2011.83
- Benjamin Welton, Evan Samanas, and Barton P. Miller. 2013. **Mr. Scan: extreme scale density-based clustering using a tree-based network of GPGPU nodes**. SC'2013, article 84. DOI 10.1145/2503210.2503262
- Mostofa Ali Patwary, Nadathur Satish, Narayanan Sundaram, Fredrik Manne, Salman Habib & Pradeep Dubey. **Pardicle: parallel approximate density-based clustering**. SC'2014, pages 560-571. DOI 10.1109/SC.2014.51
- Younghoon Kim, Kyuseok Shim, Min-Soeng Kim & June Sup Lee: **DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce**. Information Systems 42:15-35, June 2014. DOI 10.1016/j.is.2013.11.002

Apéndice: Notación O

El impacto de la eficiencia de un algoritmo…

n	10	100	1000	10000	100000
O(n)	10ms	0.1s	1s	10s	100s
O(n·log$_2$ n)	33ms	0.7s	10s	2 min	28 min
O(n^2)	100ms	10s	17 min	28 horas	115 días
O(n^3)	1s	17min	12 días	31 años	32 milenios