

Cómputo Paralelo

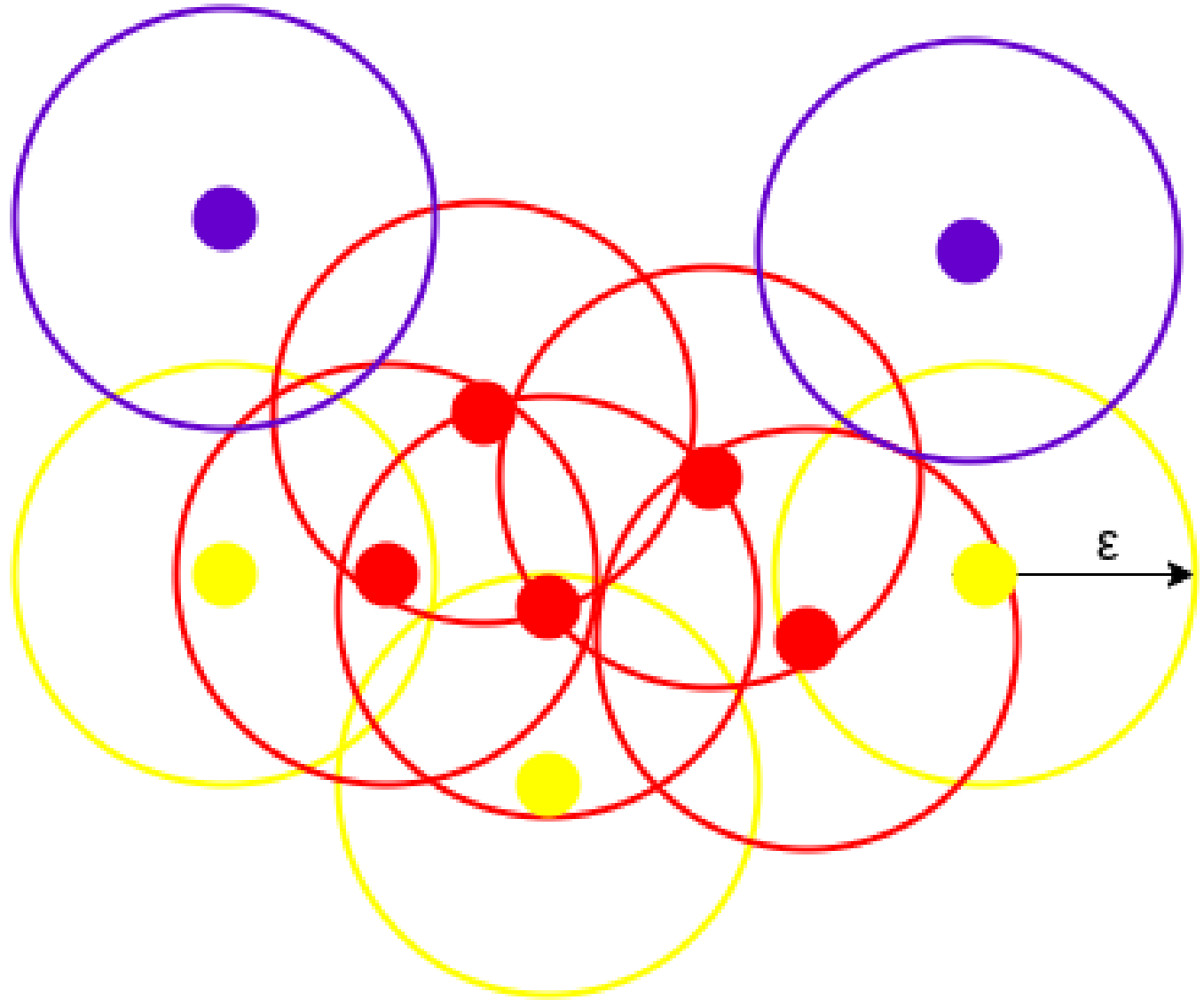
y en la Nube

Proyecto

Apertura

*DBSCAN paralelo para la
detección de ruido o
outliers con **OpenMP***

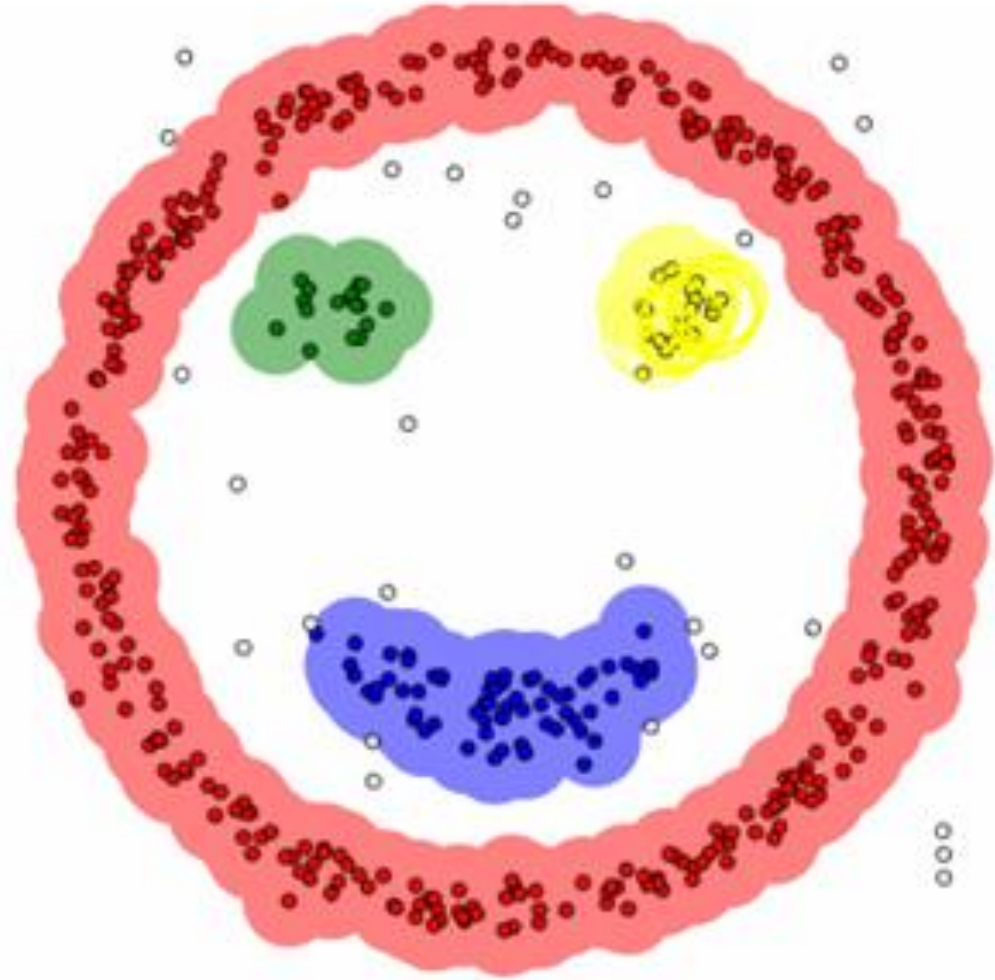
Density- Based Spatial Clustering of Applications with Noise



Density- Based Spatial Clustering of Applications with Noise

Clusters

Outliers



DBSCAN - parámetros

epsilon: Distancia

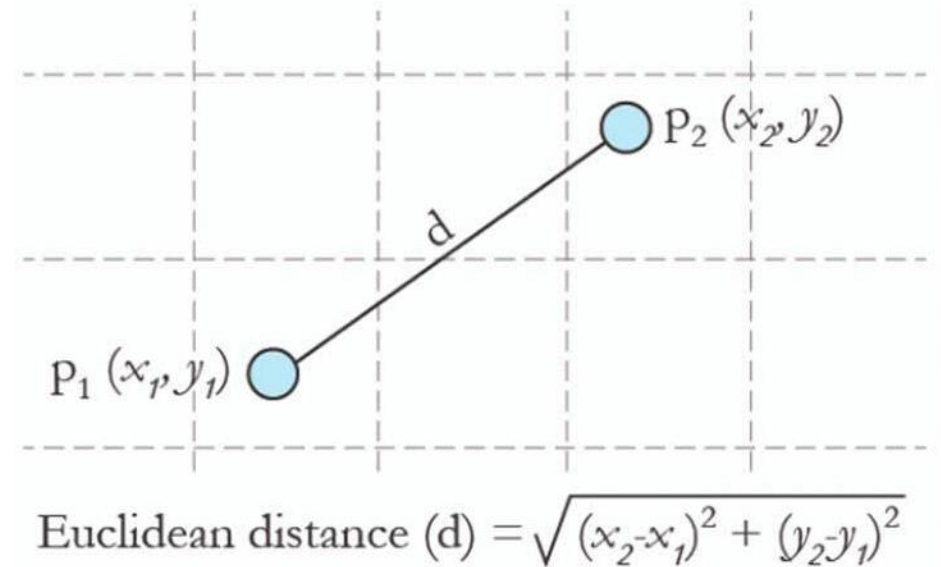
min_samples: Núm mínimo de puntos a distancia épsilon para ser denso

Tres tipos de nodos:

- Nodo **core** (de *primer* grado)
- Nodo **ruido** ó **outlier**
- Nodo **core** (de *segundo* grado)

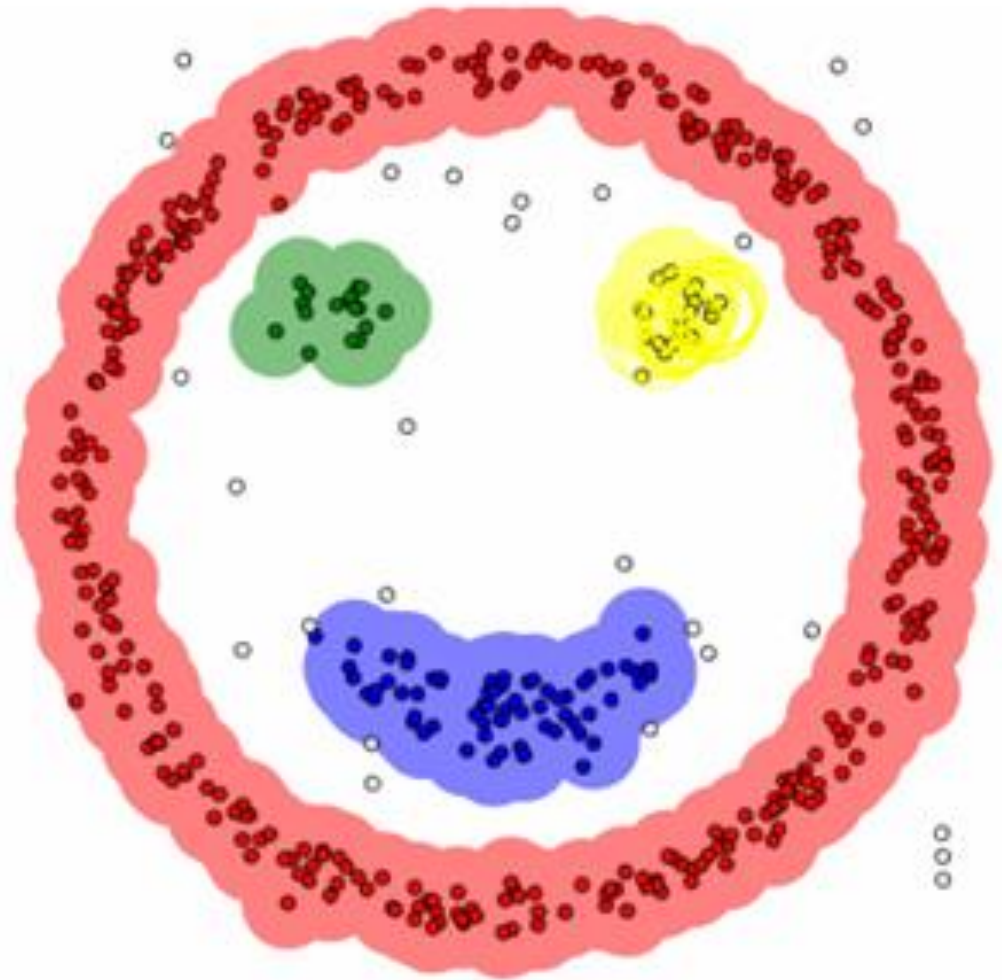
DBSCAN – algoritmo ingenuo para detectar ruido

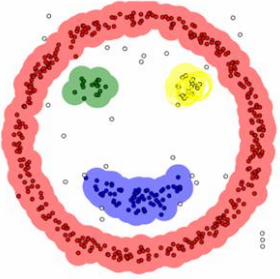
- **Paso 1:** Determina la categoría de cada punto en función de la distancia épsilon y del número mínimo de puntos a distancia épsilon para ser una zona densa.
 - Si un punto está en medio de una zona densa, es punto core (de primer orden)
 - Si no, es un punto ruido o outlier



DBSCAN – algoritmo ingenuo para detectar ruido

- **Paso 2:** Determina si un punto ruido (o outlier) es épsilon-alcanzable desde un punto core, si es el caso, re-etiquétalo como punto core (de segundo orden).





DBSCAN paralelo para la detección de ruido o outliers

- ¿Qué debo entregar?
 - Todo el código fuente
 - Versión serial.
 - Versión paralelizada.
 - Escrito con una descripción del código y de la estrategia de paralelización
 - Escrito con una descripción de la evaluación experimental de desempeño:
 - Explicación detallada de la definición del experimento.
 - Descripción del equipo donde se ejecutaron los experimentos en términos de hardware y software.
 - Interpretación y análisis de resultados.
 - Incluir gráficas
 - Archivo con los datos de los experimentos

Experimento para evaluar Rendimiento

Parametrizar el programa

- Número de puntos
{20000, 40000, 80000, 120000, 140000, 160000, 180000, 200000}
- Número de hilos
{1, (número de cores virtuales)/2, número de cores virtuales, número de cores virtuales*2}
- Promediar diez iteraciones para cada configuración.
- *Entrada del programa: PUNTOS ALEATORIOS. El programa debe recibir (como entrada) un archivo csv que se puede generar usando la libreta **DBSCAN_noise.ipynb** que se encuentra en el repositorio del curso.*
- *Salida del programa: PUNTOS CORE Y RUIDO. El programa debe generar (como resultado) un archivo csv que se pueda visualizar usando la libreta **DBSCAN_noise.ipynb** que se encuentra en el repositorio del curso.*
- *Comparar contra version serial*
- *Obtener gráfica de Speed Ups*

Criterios de evaluación

- Equipo de 1 o 2 personas.
- Peso total del proyecto: 2 puntos de su calificación final.
- Ejecución del proyecto con todos los requerimientos indicados en su descripción (1.2 puntos)
- Calidad y presentación de los documentos de descripción de código y evaluación experimental de desempeño (0.8 puntos)
- Fecha de entrega de código y documentación: **Jueves 17 de Octubre de 2024** en clase.
- **NOTA 1:** El programa paralelo debe alcanzar al menos un speedup de **1.5**. Caso contrario, habrá una penalización de 50%.
- **NOTA 2:** Si es una sola persona puede 1) hacer la mitad de los experimentos, 2) no entregar el escrito con la descripción del código y de la estrategia de paralelización (aunque Sí tiene que entregar el escrito con la descripción de la evaluación experimental de desempeño).
- **NOTA 3:** 20% menos por cada día natural de retraso.
- **NOTA 4:** Si se entrega después de la hora de entrega, en automático aplica un día menos.