Neo4j: Detección de Comunidades

Big Data Aplicado

Julio Alberto López Gómez
JulioAlberto.Lopez@uclm.es

Big_Data Aplicado

Curso Especialización Inteligencia_Artificial y Big_Data

Detección de Comunidades:

Concepto:

- En grafos es inevitable que aparezcan comunidades
  - Grupos de nodos con más relaciones entre sí que con otros nodos fuera de la comunidad
  - Componentes conexas
- Permite identificar comportamientos emergentes y de rebaño
  - Grupos/Clústers de nodos
  - Grupos aislados
  - …

Detección de Comunidades:

Aplicaciones:

- Permite detectar hábitos y preferencias en los usuarios
- Permite inferir el comportamiento de un actor en una red
- Muy útiles para la visualización e inspección de un grafo
- …

Detección de Comunidades:

Conteo de triángulos:

- Número de triángulos que pasan por cada nodo del grafo
  - Triángulo: Conjunto de tres nodos que tienen relaciones entre sí
- Permite estudiar o inspeccionar de forma global el grafo
- Permite estudiar o inspeccionar una componente conexa del grafo

Detección de Comunidades:

Coeficiente de clustering:

- Proporciona una medida cuantitativa del grado de agrupación de un grupo
  - Para ello, se compara con el grado máximo de agrupación que podría tener
  - Se utiliza el conteo de triángulos
- Coeficiente local:
- $CC(u) = \frac{2R_u}{k_u(k_u-1)}$
  - Ru son los triángulos de u
  - ku es el grado de u

Detección de Comunidades:

Ejemplo:

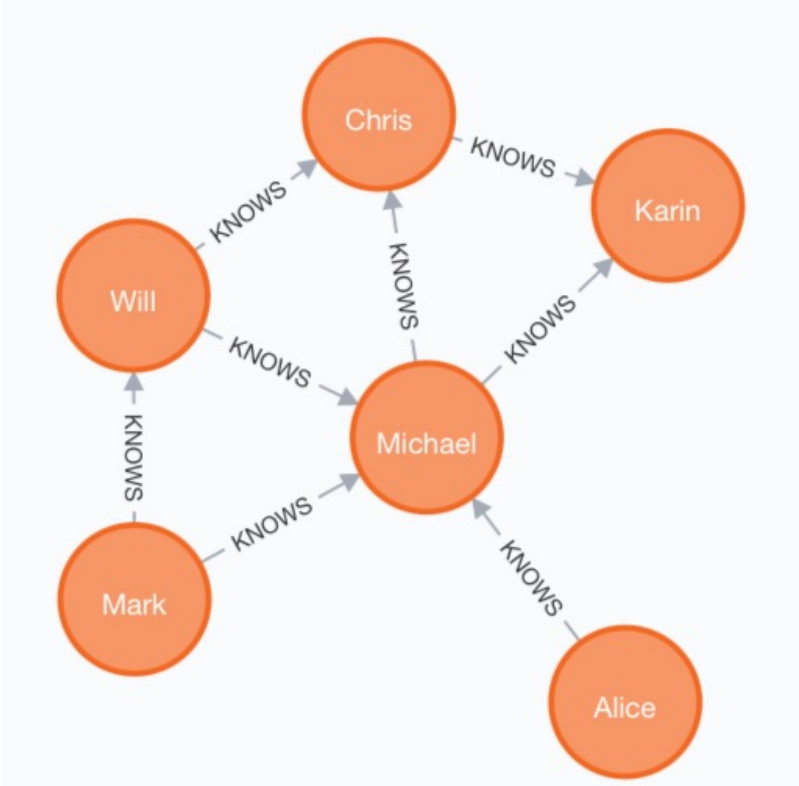
- Creación de grafo
  - Creación de los nodos y enlaces

```
CREATE
  (alice:Person {name: 'Alice'}),
  (michael:Person {name: 'Michael'}),
  (karin:Person {name: 'Karin'}),
  (chris:Person {name: 'Chris'}),
  (will:Person {name: 'Will'}),
  (mark:Person {name: 'Mark'}),

  (michael)-[:KNOWS]->(karin),
  (michael)-[:KNOWS]->(chris),
  (will)-[:KNOWS]->(michael),
  (mark)-[:KNOWS]->(michael),
  (mark)-[:KNOWS]->(will),
  (alice)-[:KNOWS]->(michael),
  (will)-[:KNOWS]->(chris),
  (chris)-[:KNOWS]->(karin)
```

Detección de Comunidades:

- Creación de grafo
  - Creación de los nodos y enlaces

Ejemplo:



Detección de Comunidades:

Conteo de triángulos:

- Almacenamiento del grafo

```
CALL gds.graph.create(
  'myGraph',
  'Person',
  {
    KNOWS: {
      orientation: 'UNDIRECTED'
    }
  }
)
```

Detección de Comunidades:

Conteo de triángulos:

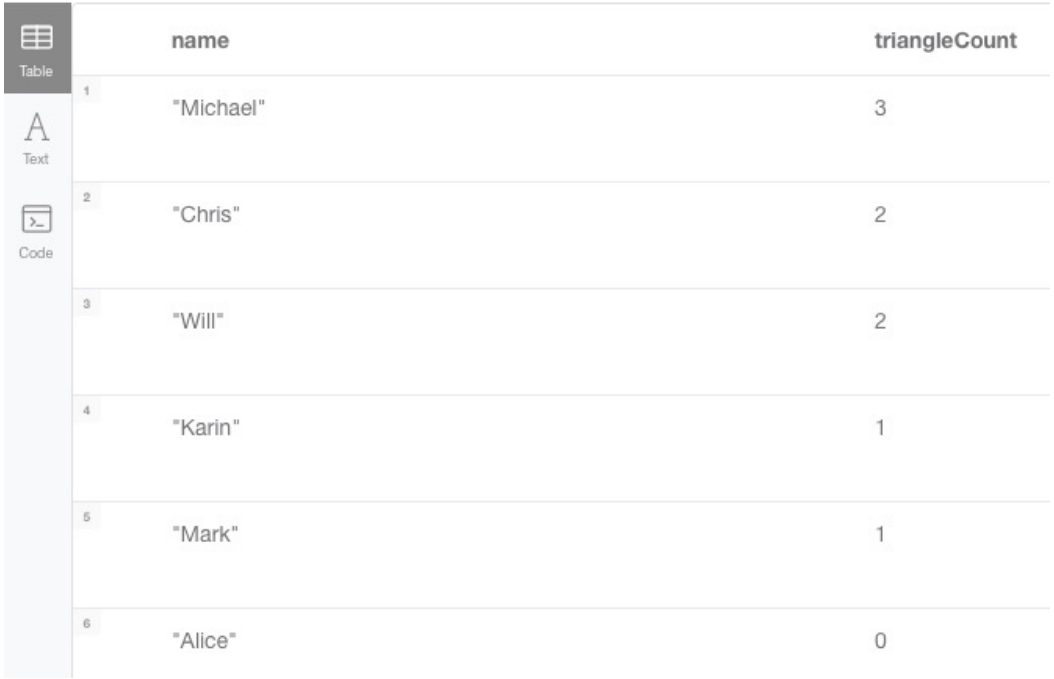
- Ejecución:

```
CALL
gds.triangleCount.stream('myGraph')

YIELD nodeId, triangleCount

RETURN gds.util.asNode(nodeId).name

AS name, triangleCount
ORDER BY triangleCount DESC
```

Detección de Comunidades:

Conteo de triángulos:

- Ejecución:



	name	triangleCount
1	"Michael"	3
2	"Chris"	2
3	"Will"	2
4	"Karin"	1
5	"Mark"	1
6	"Alice"	0

Detección de Comunidades:

Coeficiente local de clustering:

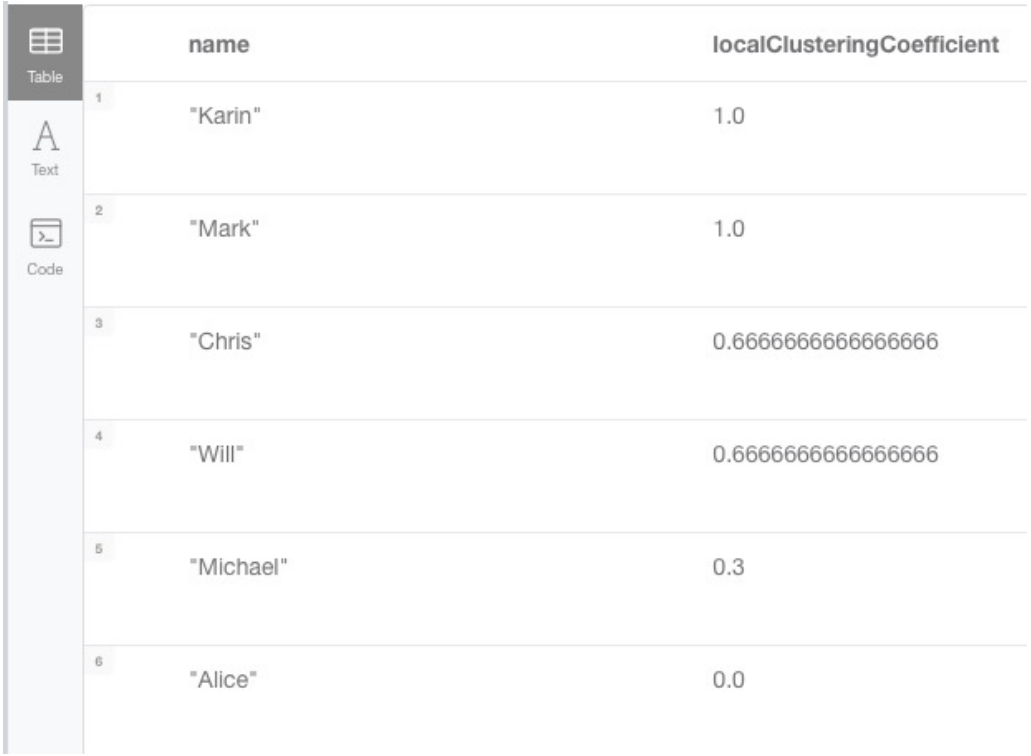
- Ejecución:

```
CALL
gds.localClusteringCoefficient.stream
('myGraph')

YIELD nodeId,
localClusteringCoefficient

RETURN gds.util.asNode(nodeId).name
AS name, localClusteringCoefficient

ORDER BY localClusteringCoefficient
DESC
```

Detección de Comunidades:

Resultados:

- Ejecución:

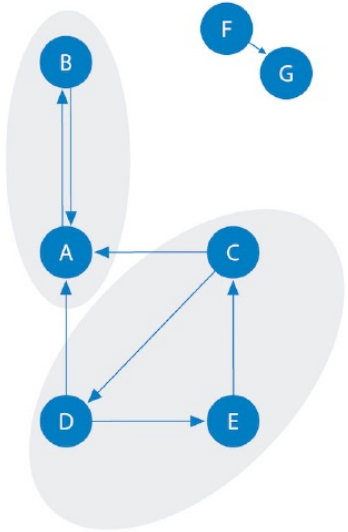


	name	localClusteringCoefficient
1	"Karin"	1.0
2	"Mark"	1.0
3	"Chris"	0.6666666666666666
4	"Will"	0.6666666666666666
5	"Michael"	0.3
6	"Alice"	0.0

Detección de Comunidades:

Componentes fuertemente conexas:

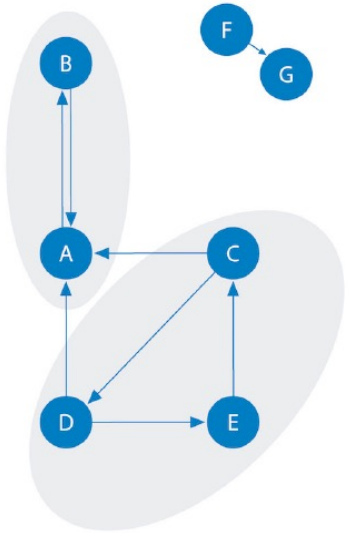
- Busca un conjunto de nodos en un grafo dirigido.
  - Cualquier nodo debe ser alcanzable desde cualquier otro en ambas direcciones
- Tiempo de procesamiento proporcional al número de nodos



Detección de Comunidades:

Componentes fuertemente conexas:

- Aplicaciones:
  - Estudiar la conectividad de una red para medir su rendimiento
  - Aplicación de algoritmos solamente sobre estas componentes



Detección de Comunidades:

Componentes fuertemente conexas:

- Ejecución:

```
CALL gds.alpha.scc.stream({
  nodeProjection: 'Person',
  relationshipProjection: 'KNOWS'
})

YIELD nodeId, componentId

RETURN gds.util.asNode(nodeId).name
AS Name, componentId AS Component

ORDER BY Component DESC
```

Detección de Comunidades:

Componentes fuertemente conexas:

- Ejecución:

Table | Text | Code

	Name	Component
1	"Mark"	5
2	"Will"	4
3	"Chris"	3
4	"Karin"	2
5	"Michael"	1
6	"Alice"	0

Neo4j: Detección de Comunidades

Big Data Aplicado

Julio Alberto López Gómez
JulioAlberto.Lopez@uclm.es

Big_Data Aplicado

Curso Especialización Inteligencia_Artificial y Big_Data