**neo4j的cypher语句**

**——18301092 陈佳林**

**（1）在命名图形的流模式（stream mode）下运行 Louvain。**

**CALL gds.louvain.stream(**

**graphName: String,**

**configuration: Map**

**)**

**YIELD**

**nodeId: Integer,**

**communityId: Integer,**

**intermediateCommunityIds: Integer[]**

**（2）在命名图形上以统计模式（stats mode）运行 Louvain。**

**CALL gds.louvain.stats(**

**graphName: String,**

**configuration: Map**

**)**

**YIELD**

**createMillis: Integer,**

**computeMillis: Integer,**

**postProcessingMillis: Integer,**

**communityCount: Integer,**

**ranLevels: Integer,**

**modularity: Float,**

**modularities: Integer[],**

**communityDistribution: Map,**

**configuration: Map**

**（3）在命名图形上以变异模式（mutate mode）运行 Louvain。**

**CALL gds.louvain.mutate(**

**graphName: String,**

**configuration: Map**

**)**

**YIELD**

**createMillis: Integer,**

**computeMillis: Integer,**

**mutateMillis: Integer,**

**postProcessingMillis: Integer,**

**communityCount: Integer,**

**ranLevels: Integer,**

**modularity: Float,**

**modularities: Integer[],**

**communityDistribution: Map,**

**configuration: Map**

**（4）在命名图形上以写入模式（write mode）运行 Louvain。**

**CALL gds.louvain.write(**

**graphName: String,**

**configuration: Map**

**)**

**YIELD**

**createMillis: Integer,**

**computeMillis: Integer,**

**writeMillis: Integer,**

**postProcessingMillis: Integer,**

**nodePropertiesWritten: Integer,**

**communityCount: Integer,**

**ranLevels: Integer,**

**modularity: Float,**

**modularities: Integer[],**

**communityDistribution: Map,**

**configuration: Map**

**（5）在匿名图形上以写入模式（write mode）运行 Louvain。**

**CALL gds.louvain.write(configuration: Map)**

**YIELD**

**createMillis: Integer,**

**computeMillis: Integer,**

**writeMillis: Integer,**

**postProcessingMillis: Integer,**

**nodePropertiesWritten: Integer,**

**communityCount: Integer,**

**ranLevels: Integer,**

**modularity: Float,**

**modularities: Integer[],**

**communityDistribution: Map,**

**configuration: Map**

**EXEMPLE：**

**创建图：**

**CREATE**

**(nAlice:User {name: 'Alice', seed: 42}),**

**(nBridget:User {name: 'Bridget', seed: 42}),**

**(nCharles:User {name: 'Charles', seed: 42}),**

**(nDoug:User {name: 'Doug'}),**

**(nMark:User {name: 'Mark'}),**

**(nMichael:User {name: 'Michael'}),**

**(nAlice)-[:LINK {weight: 1}]->(nBridget),**

**(nAlice)-[:LINK {weight: 1}]->(nCharles),**

**(nCharles)-[:LINK {weight: 1}]->(nBridget),**

**(nAlice)-[:LINK {weight: 5}]->(nDoug),**

**(nMark)-[:LINK {weight: 1}]->(nDoug),**

**(nMark)-[:LINK {weight: 1}]->(nMichael),**

**(nMichael)-[:LINK {weight: 1}]->(nMark);**

**创建图形并将其存储在图形目录中：**

**CALL gds.graph.create(**

**'myGraph',**

**'User',**

**{**

**LINK: {**

**orientation: 'UNDIRECTED'**

**}**

**},**

**{**

**nodeProperties: 'seed',**

**relationshipProperties: 'weight'**

**}**

**)**

**估计运行算法的内存要求：**

**CALL gds.louvain.write.estimate('myGraph', { writeProperty: 'community' })**

**YIELD nodeCount, relationshipCount, bytesMin, bytesMax, requiredMemory**

**Stream mode下的运行：**

**CALL gds.louvain.stream('myGraph')**

**YIELD nodeId, communityId, intermediateCommunityIds**

**RETURN gds.util.asNode(nodeId).name AS name, communityId, intermediateCommunityIds**

**ORDER BY name ASC**

**Stats mode下运行：**

**CALL gds.louvain.stats('myGraph')**

**YIELD communityCount**

**Mutate mode下运行：**

**CALL gds.louvain.mutate('myGraph', { mutateProperty: 'communityId' })**

**YIELD communityCount, modularity, modularities**

**Write mode下运行：**

**CALL gds.louvain.write('myGraph', { writeProperty: 'community' })**

**YIELD communityCount, modularity, modularities**

**在加权图上运行：**

**CALL gds.louvain.stream('myGraph', { relationshipWeightProperty: 'weight' })**

**YIELD nodeId, communityId, intermediateCommunityIds**

**RETURN gds.util.asNode(nodeId).name AS name, communityId, intermediateCommunityIds**

**ORDER BY name ASC**

**指定种子选手下的运行：**

**CALL gds.louvain.stream('myGraph', { seedProperty: 'seed' })**

**YIELD nodeId, communityId, intermediateCommunityIds**

**RETURN gds.util.asNode(nodeId).name AS name, communityId, intermediateCommunityIds**

**ORDER BY name ASC**

**查看迭代的运行过程：**

**CREATE (a:Node {name: 'a'})**

**CREATE (b:Node {name: 'b'})**

**CREATE (c:Node {name: 'c'})**

**CREATE (d:Node {name: 'd'})**

**CREATE (e:Node {name: 'e'})**

**CREATE (f:Node {name: 'f'})**

**CREATE (g:Node {name: 'g'})**

**CREATE (h:Node {name: 'h'})**

**CREATE (i:Node {name: 'i'})**

**CREATE (j:Node {name: 'j'})**

**CREATE (k:Node {name: 'k'})**

**CREATE (l:Node {name: 'l'})**

**CREATE (m:Node {name: 'm'})**

**CREATE (n:Node {name: 'n'})**

**CREATE (x:Node {name: 'x'})**

**CREATE (a)-[:TYPE]->(b)**

**CREATE (a)-[:TYPE]->(d)**

**CREATE (a)-[:TYPE]->(f)**

**CREATE (b)-[:TYPE]->(d)**

**CREATE (b)-[:TYPE]->(x)**

**CREATE (b)-[:TYPE]->(g)**

**CREATE (b)-[:TYPE]->(e)**

**CREATE (c)-[:TYPE]->(x)**

**CREATE (c)-[:TYPE]->(f)**

**CREATE (d)-[:TYPE]->(k)**

**CREATE (e)-[:TYPE]->(x)**

**CREATE (e)-[:TYPE]->(f)**

**CREATE (e)-[:TYPE]->(h)**

**CREATE (f)-[:TYPE]->(g)**

**CREATE (g)-[:TYPE]->(h)**

**CREATE (h)-[:TYPE]->(i)**

**CREATE (h)-[:TYPE]->(j)**

**CREATE (i)-[:TYPE]->(k)**

**CREATE (j)-[:TYPE]->(k)**

**CREATE (j)-[:TYPE]->(m)**

**CREATE (j)-[:TYPE]->(n)**

**CREATE (k)-[:TYPE]->(m)**

**CREATE (k)-[:TYPE]->(l)**

**CREATE (l)-[:TYPE]->(n)**

**CREATE (m)-[:TYPE]->(n);**

**CALL gds.louvain.stream({**

**nodeProjection: 'Node',**

**relationshipProjection: {**

**TYPE: {**

**type: 'TYPE',**

**orientation: 'undirected',**

**aggregation: 'NONE'**

**}**

**},**

**includeIntermediateCommunities: true**

**}) YIELD nodeId, communityId, intermediateCommunityIds**

**RETURN gds.util.asNode(nodeId).name AS name, communityId, intermediateCommunityIds**

**ORDER BY name ASC**