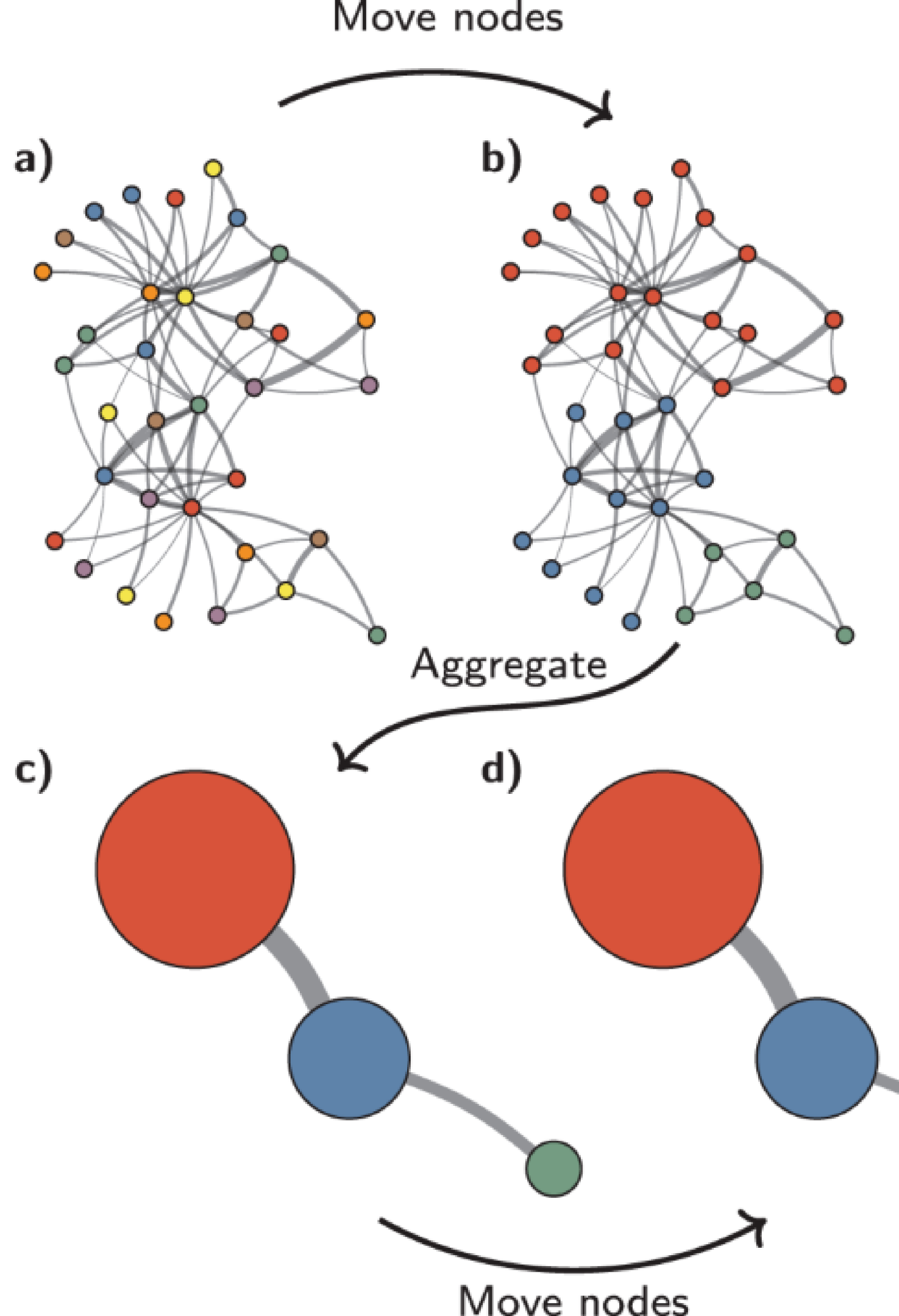


Algoritmo de Louvain



IIC2433 - Minería de Datos
Primer Semestre 2023

EL ALGORITMO

A grandes razgos

1 - Asignación inicial



Se debe asignar a cada elemento del grafo una comunidad

2 - Cálculo de modularidad



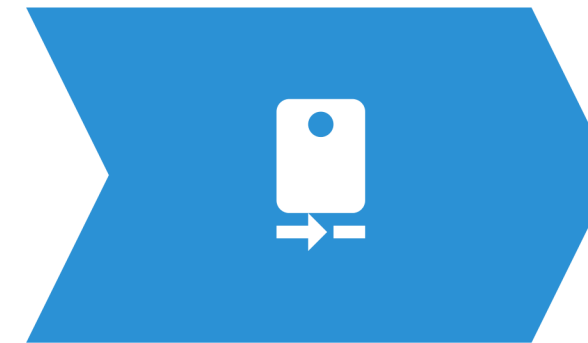
Se debe hacer el cálculo inicial de la modularidad

3 - Randomizar los nodos



Randomizamos la manera en que recorremos los nodos

4 - Asignación de cada nodo



Para cada nodo, buscamos asignar la mejor comunidad, en el orden que calculamos en 3

5 - Repetir pasos



Cuando asignemos todos los nodos, volvemos a realizar el paso 3. En caso de que ningún nodo cambie, termina el algoritmo *

1.

Asignación inicial

- Guardaremos en nuestra Clase Comunidades:
 - La lista de nodos para recorrerlos
 - El grafo
 - Una lista de Comunidades (recomendación: Class)
- Debemos definir inicialmente una comunidad como un nodo por sí solo
- Debemos calcular la modularidad con la fórmula que se encuentra en el paper indicado

2.

Cálculo de la modularidad

NO ES LA FÓRMULA QUE SE USA

$$M = \frac{1}{2m} \sum_{i,j} (A_{ij} - p_{ij}) \delta(c_i, c_j) = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j)$$

Donde:

- m = Cantidad de aristas del grafo
- $A_{\{ij\}}$ = El peso de la arista que conecta i y j
- k_i = Grados del nodo i
- k_j = Grados del nodo j
- $\delta(c_i, c_j) = 1$ si los nodos están en la misma comunidad. 0 e.o.c

3.

Randomizar los nodos

Cambiar el orden en que se recorren los nodos, viene dado es parte de nuestros atributos globales



4.

Asignación de cada nodo

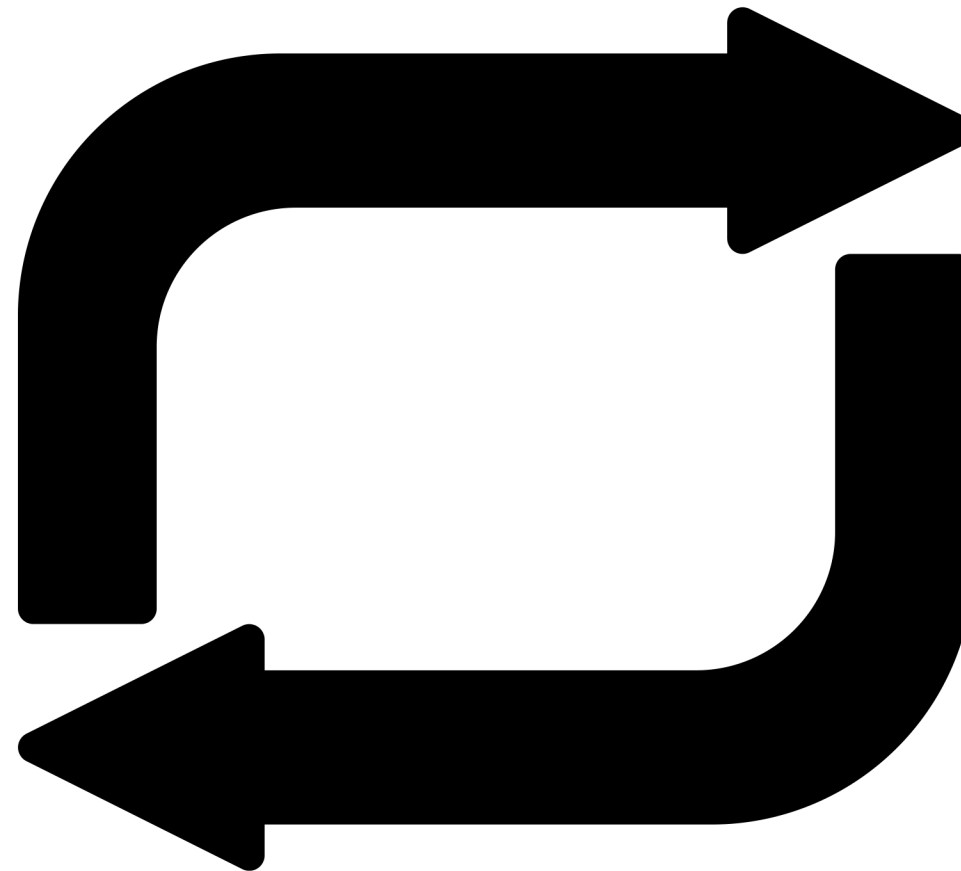
- Por cada nodo intentaremos en orden:
 - Sacarlo de la comunidad en que está (quitando la modularidad respectiva)
 - Asignar al nodo a la comunidad que tenga mejor incremento de modularidad
 - Recalcular modularidad
 - Nota: No calcular toda la modularidad nuevamente, usar delta:
 - NUEVAMENTE: FÓRMULA DEMOSTRATIVA, LA REAL PARA NUESTRO CASO ESTÁ EN EL PAPER

$$\Delta M = \left[\frac{\Sigma_{in} + 2k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

5.

Repetir pasos

- Repetiremos los pasos 3 y 4
- Dejamos de repetir cuando en el paso 4 para ningún nodo se haya mejorado la modularidad





6.

Último paso

- Repetiremos los 5 pasos anteriores, creando un nuevo grafo, haciendo merge de todas los nodos en uno solo
- La nueva arista de un nodo 1 a un nodo 2 (antes comunidad) tendrá el peso de la suma de todos los pesos de las aristas que hayan ido de la comunidad 1 a la 2
- Lo mismo con las aristas dentro de la misma comunidad, se representan como una arista hacia el nodo mismo

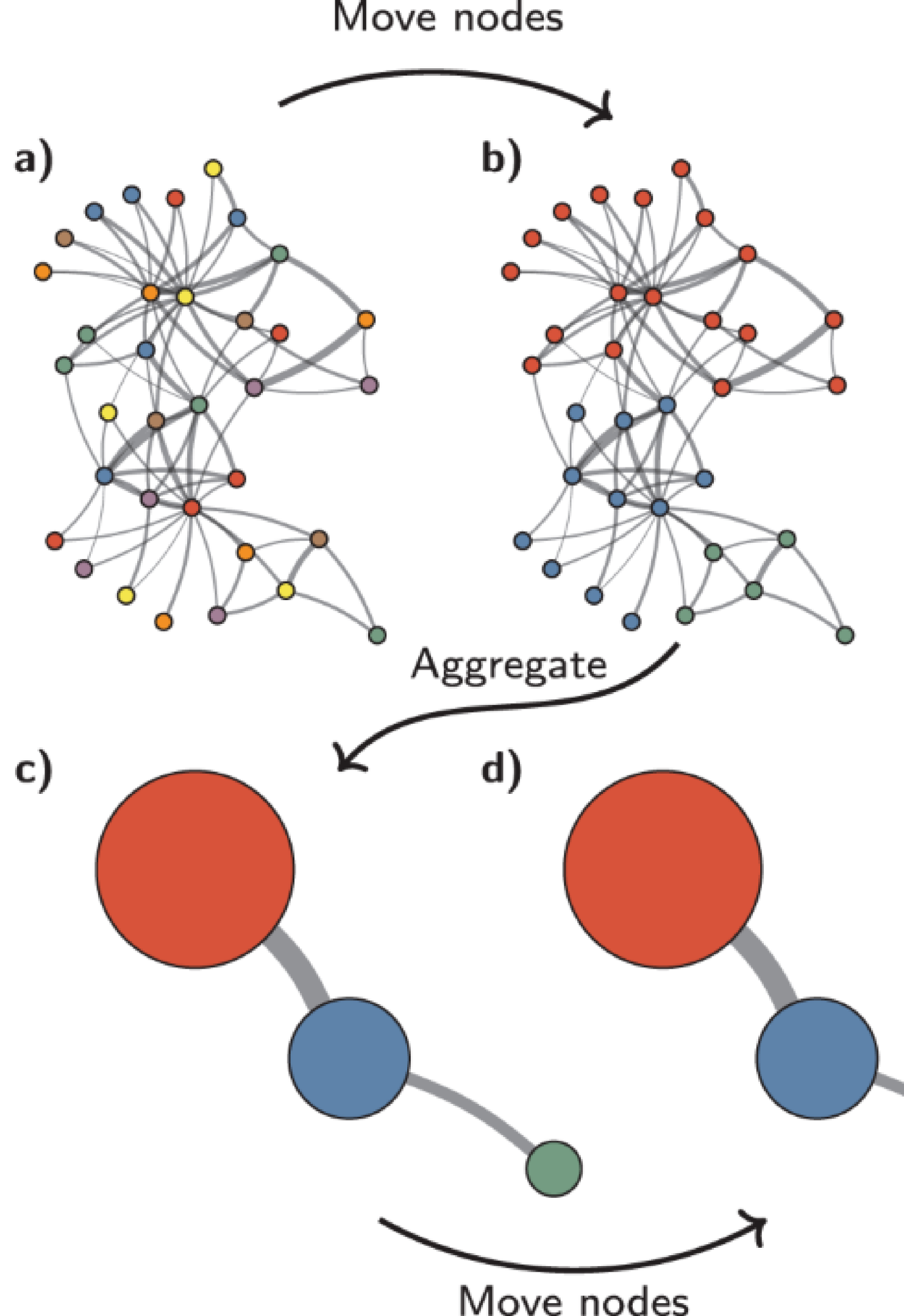
2.

Cálculo de la modularidad

$$Q_d = \frac{1}{m} \sum_{i,j} \left[A_{ij} - \frac{d_i^{\text{in}} d_j^{\text{out}}}{m} \right] \delta(c_i, c_j)$$

$$\Delta Q_d = \frac{d_i^C}{m} - \left[\frac{d_i^{\text{out}} \cdot \sum_{\text{tot}}^{\text{in}} + d_i^{\text{in}} \cdot \sum_{\text{tot}}^{\text{out}}}{m^2} \right]$$

Algoritmo de Louvain



IIC2433 - Minería de Datos
Primer Semestre 2023