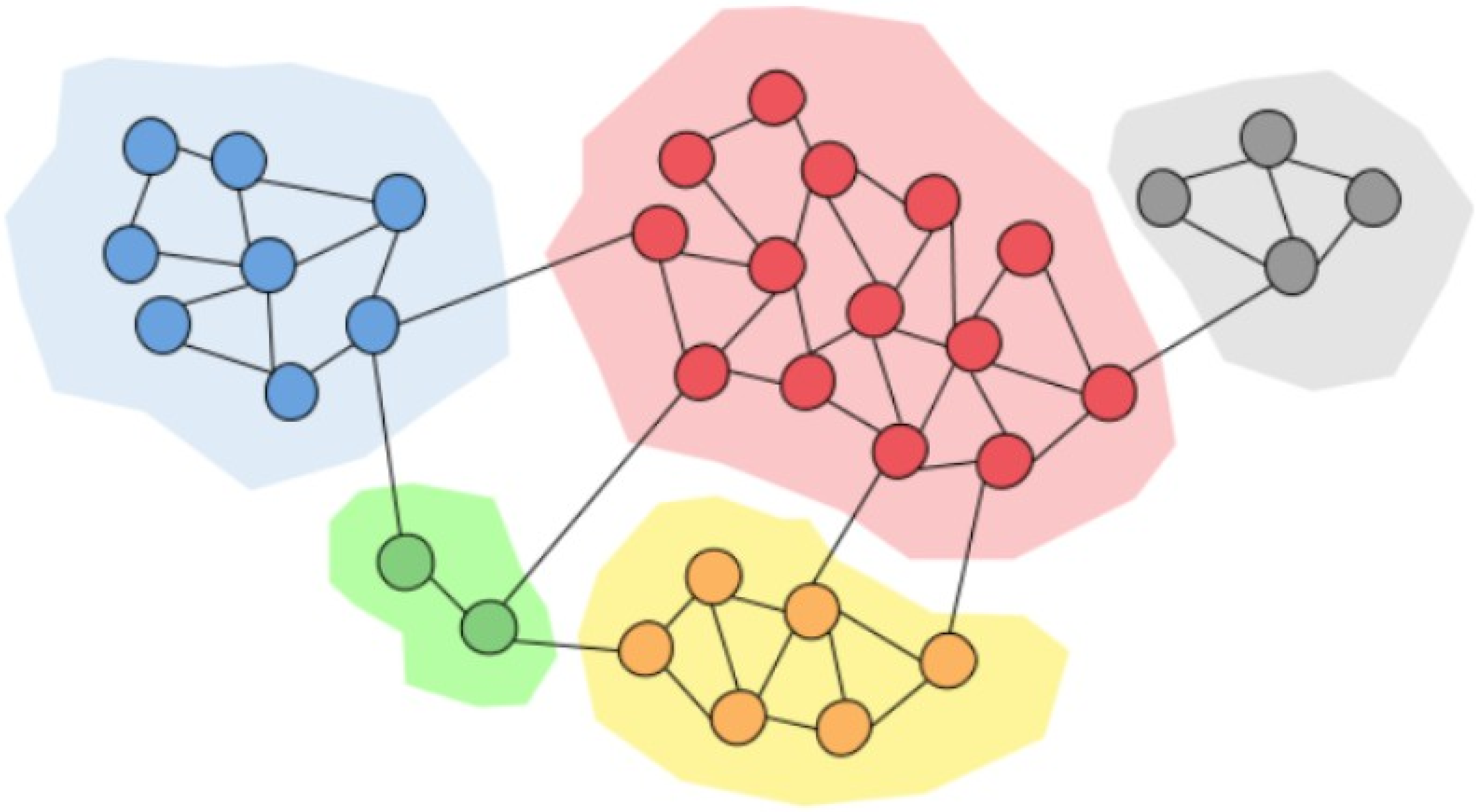
IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

- LOUVAIN -

Clustering en grafos

En redes sociales los clusters se denominan comunidades.

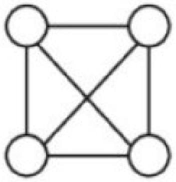


Una comunidad es un subgrafo denso

Clustering en grafos

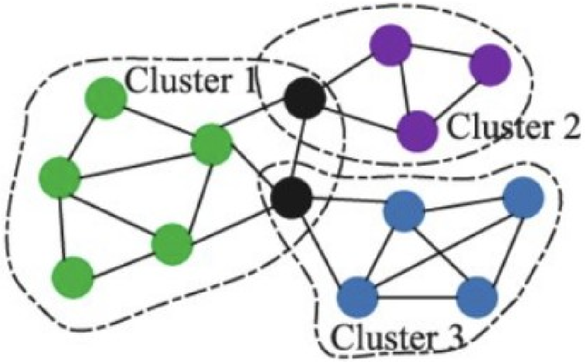
Densidad local según conectividad de subgrafos

- El subgrafo de máxima conectividad se denomina **clique**.



4-clique

- Una primera idea para particionar un grafo es buscar **cliques**:



3-clique clusters

- En la práctica, los **cliques** son poco frecuentes.

Clustering en grafos

$$G(E,V), \quad |V| = N, |E| = L$$

Strong and weak communities

Consideremos un subgrafo conexo C de N_C nodos en una red $G(E,V)$.

Clustering en grafos

$$G(E, V), \quad |V| = N, |E| = L$$

Strong and weak communities

Consideremos un subgrafo conexo C de N_C nodos en una red $G(E, V)$.

$$k_i^{\text{int}}(C) : \#\text{links that connect } i \text{ to } C$$

$$k_i^{\text{ext}}(C) : \#\text{links that connect } i \text{ to } G \setminus C$$

Clustering en grafos

$$G(E,V), \quad |V| = N, |E| = L$$

Strong and weak communities

Consideremos un subgrafo conexo C de N_C nodos en una red $G(E,V)$.

$$k_i^{\text{int}}(C) : \#\text{links that connect } i \text{ to } C$$

$$k_i^{\text{ext}}(C) : \#\text{links that connect } i \text{ to } G \setminus C$$

Strong community:

$$\forall i \in C, \quad k_i^{\text{int}}(C) > k_i^{\text{ext}}(C)$$

Clustering en grafos

$$G(E,V), \quad |V| = N, |E| = L$$

Strong and weak communities

Consideremos un subgrafo conexo C de N_C nodos en una red $G(E,V)$.

$k_i^{\text{int}}(C)$: #links that connect i to C

$k_i^{\text{ext}}(C)$: #links that connect i to $G \setminus C$

Strong community:

$$\forall i \in C, \quad k_i^{\text{int}}(C) > k_i^{\text{ext}}(C)$$

Weak community:

$$\sum_{i \in C} k_i^{\text{int}}(C) > \sum_{i \in C} k_i^{\text{ext}}(C)$$

Clustering en grafos

El problema del particionamiento de un grafo

- Bisección con tamaños fijos:

#particiones: $$\frac{N!}{N_1!N_2!}$$

Clustering en grafos

El problema del particionamiento de un grafo

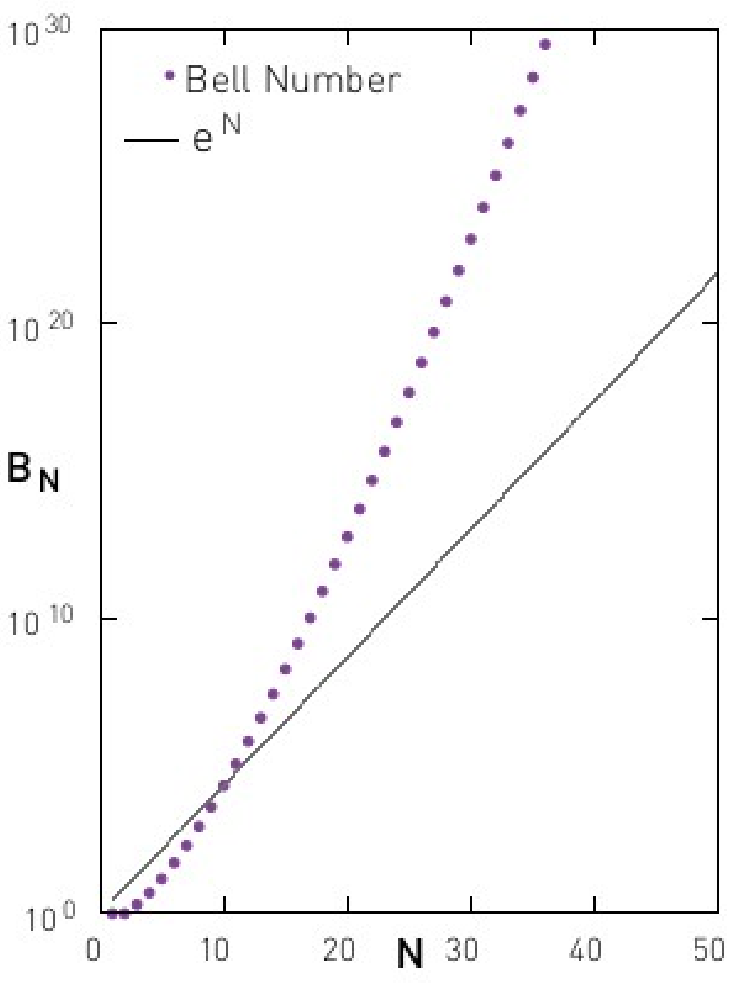
- Bisección con tamaños fijos:

#particiones: $\dfrac{N!}{N_1!N_2!}$

- Bisección con tamaños variables:

$$B_N = \frac{1}{e}\sum_{j=0}^{\infty}\frac{j^N}{j!}$$

Tamaño variable



Clustering en grafos

Hipótesis nula: Las redes conectadas por aristas *at random* carecen de estructura de comunidad.

Clustering en grafos

$$G(E, V), \quad |V| = N, |E| = L$$

Hipótesis nula: Las redes conectadas por aristas *at random* carecen de estructura de comunidad.

prob. de *random edge*

Modularidad $n_c : \#\text{communities}$

$$L_c : \#\text{links in } C,\ c \in 1, \ldots, n_c.$$

$$p_{ij} = \frac{k_i k_j}{2L}$$

Clustering en grafos

$$G(E, V), \quad |V| = N, |E| = L$$

Hipótesis nula: Las redes conectadas por aristas *at random* carecen de estructura de comunidad.

prob. de *random edge*

Modularidad n_c : #communities

L_c : #links in C, $c \in 1, \ldots, n_c$.

$$p_{ij} = \frac{k_i k_j}{2L}$$

$$M_c = \frac{1}{2L} \sum_{(i,j) \in C_c} (A_{ij} - p_{ij}).$$

Aristas observadas

Clustering en grafos

$$G(E,V), \quad |V| = N, |E| = L$$

Hipótesis nula: Las redes conectadas por aristas *at random* carecen de estructura de comunidad.

prob. de *random edge*

Modularidad n_c : #communities

L_c : #links in C, $c \in 1, \ldots, n_c$.

$$p_{ij} = \frac{k_i k_j}{2L}$$

$$M_c = \frac{1}{2L} \sum_{(i,j) \in C_c} (A_{ij} - p_{ij}).$$

Aristas observadas

…
demostrar ...

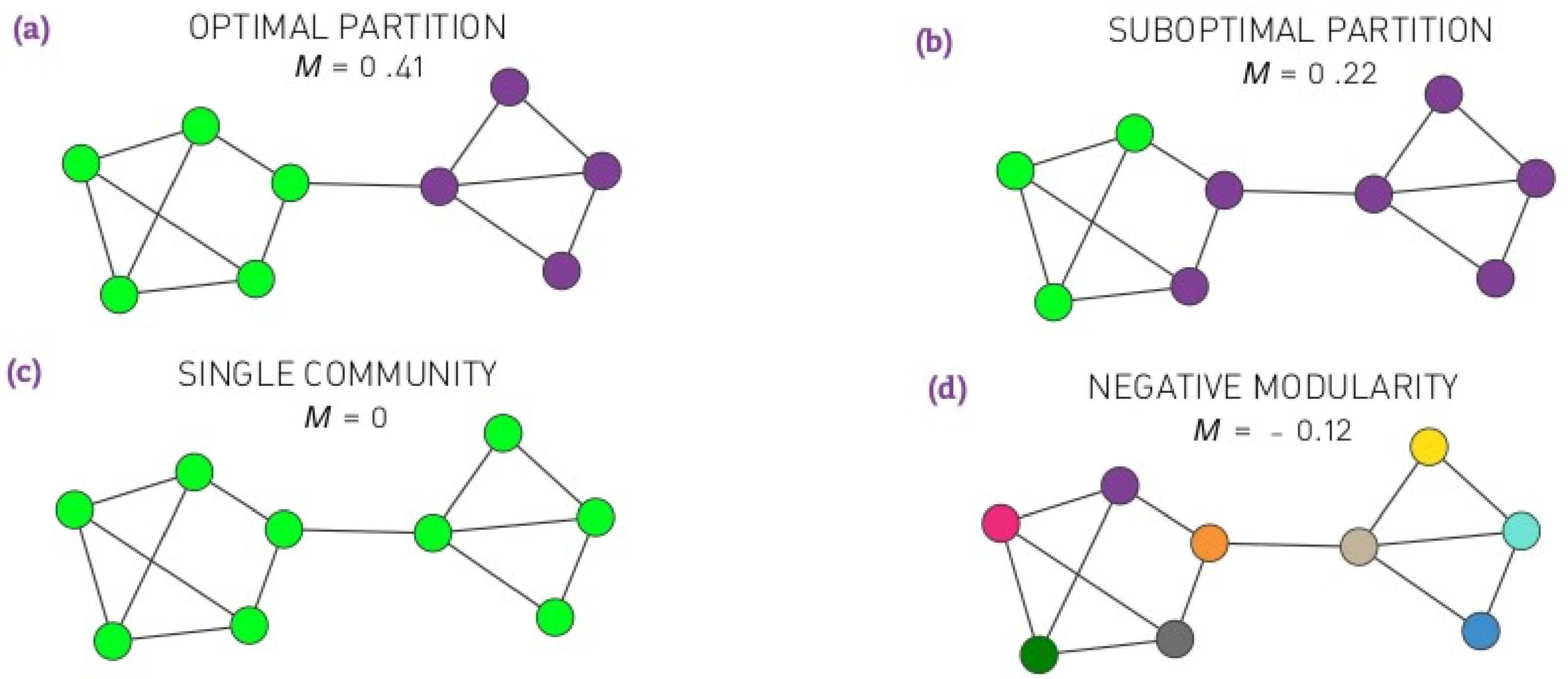
de links de todos los nodos hacia C

$$M = \sum_{c=1}^{n_c} \left[\frac{L_c}{2L} - \left(\frac{k_c}{2L} \right)^2 \right]$$

Modularidad

Clustering en grafos

Modularidad: $$M = \sum_{c=1}^{n_c}\left[\frac{L_c}{2L} - \left(\frac{k_c}{2L}\right)^2\right]$$



Si M es positivo, los subgrafos tienen más links que lo esperado (lo esperado es H_0).

$$M = \sum_{c=1}^{n_c} \left[\frac{L_c}{2L} - \left(\frac{k_c}{2L} \right)^2 \right]$$

Clustering en grafos

Algoritmo: optimización local, calcular la diferencia en modularidad si asigno un nodo a una comunidad dada.

Delta de modularidad

de links dentro de C

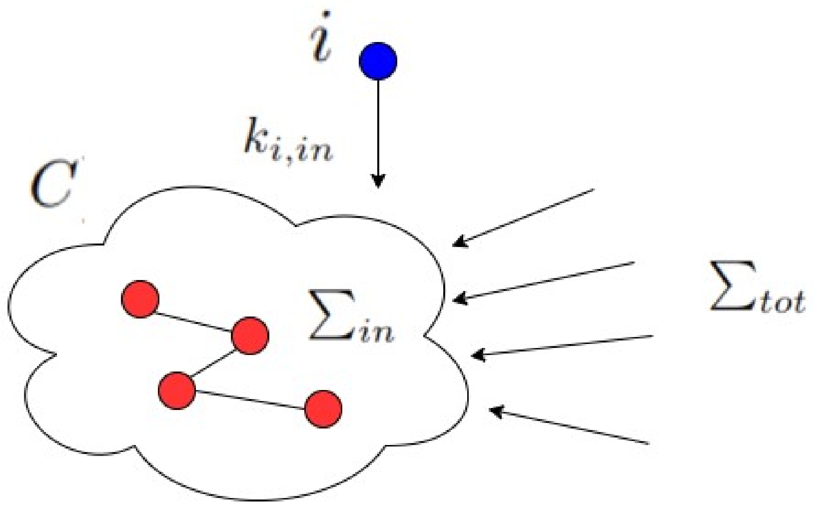
#de links desde *i* a nodos de C

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

de links de todos

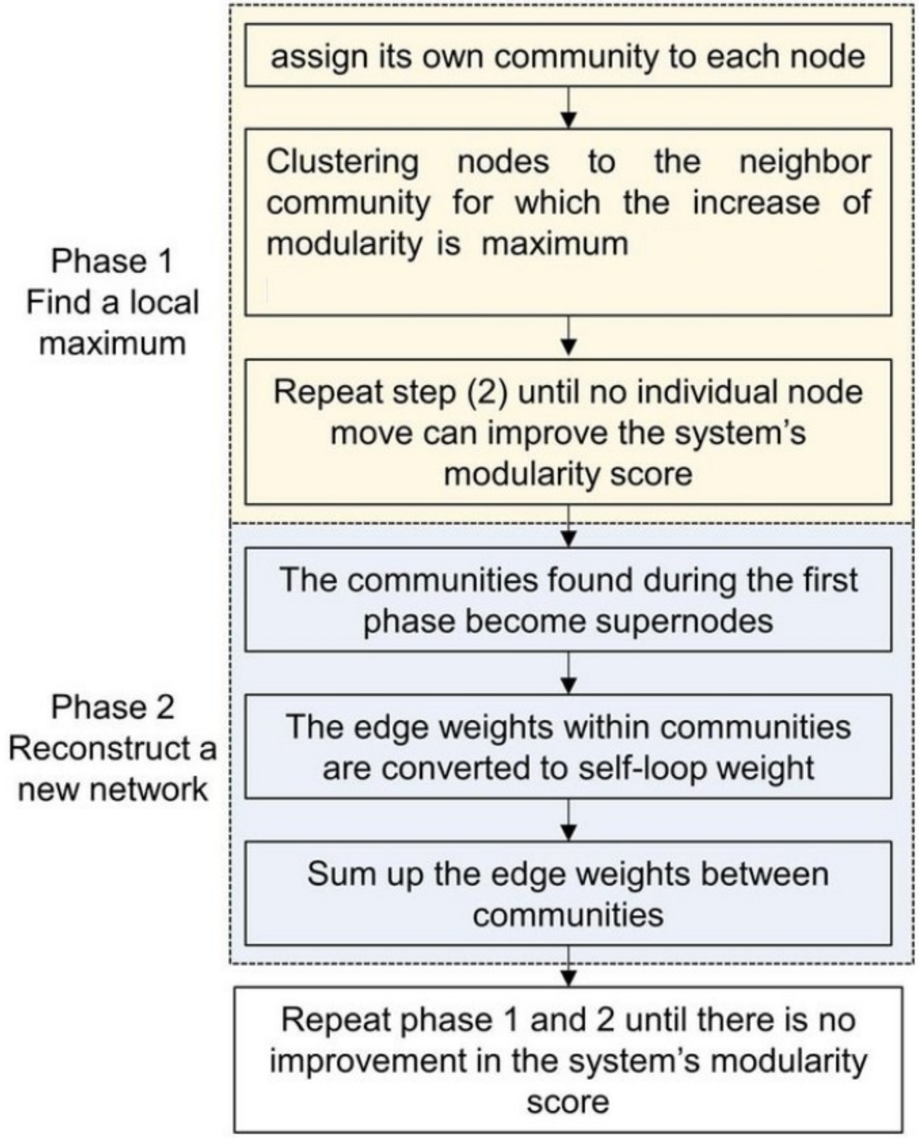
de links incidentes a *i*

los nodos hacia C (exc *i*)



Clustering en grafos

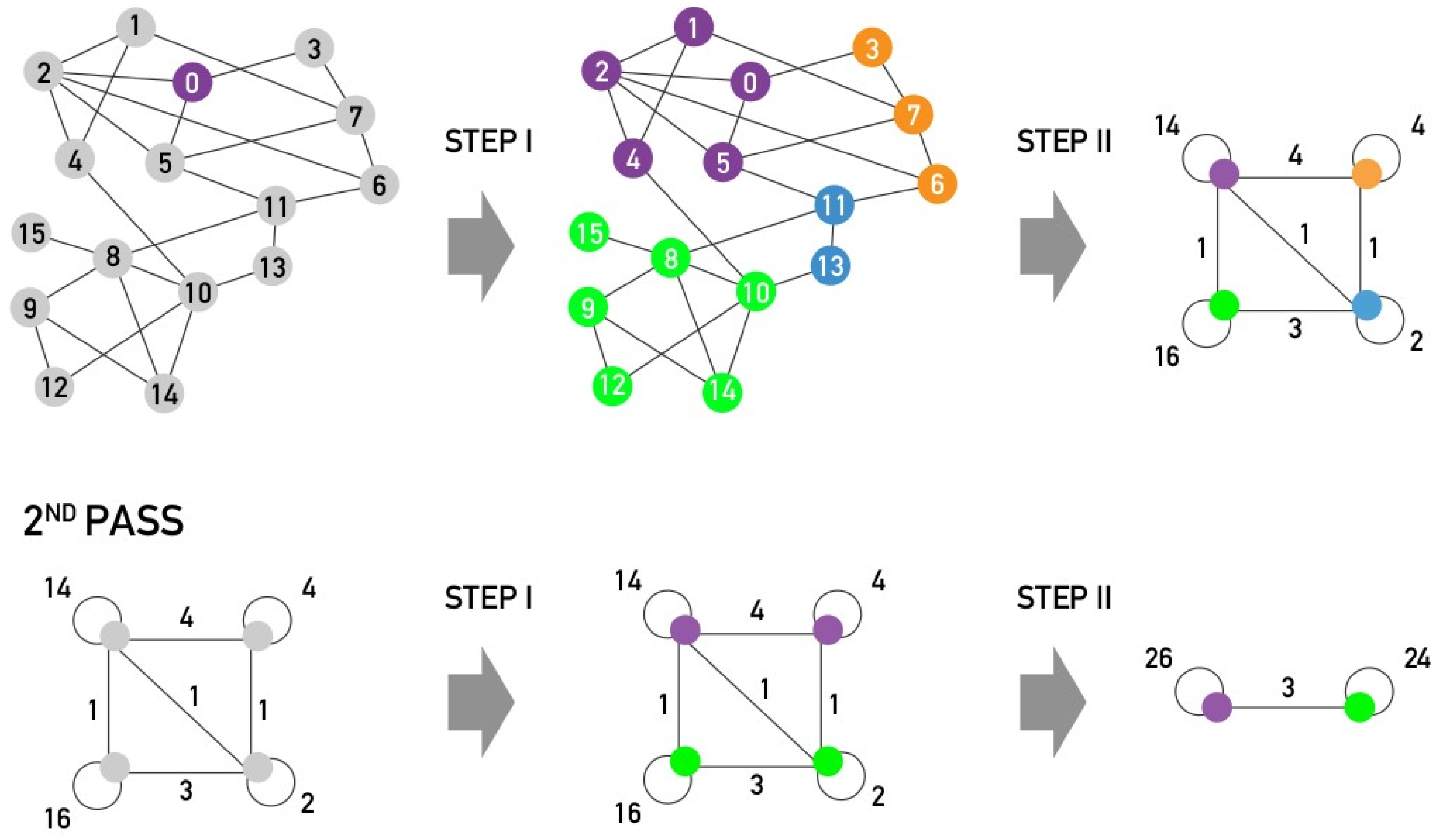
Algoritmo de Louvain



Clustering en grafos

Algoritmo de Louvain

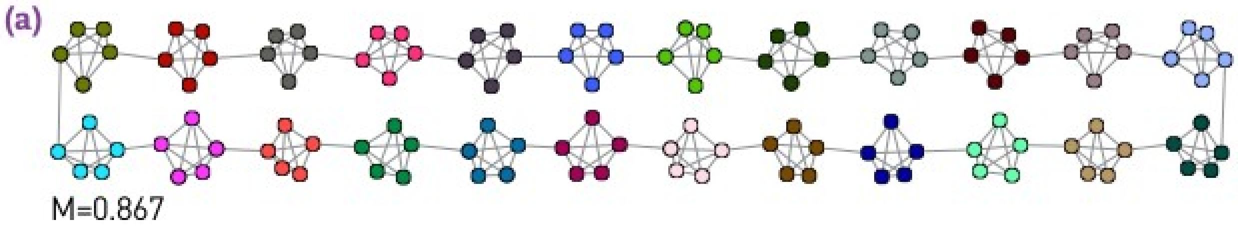
Ej.: 1ST PASS



Clustering en grafos

Algoritmo de Louvain

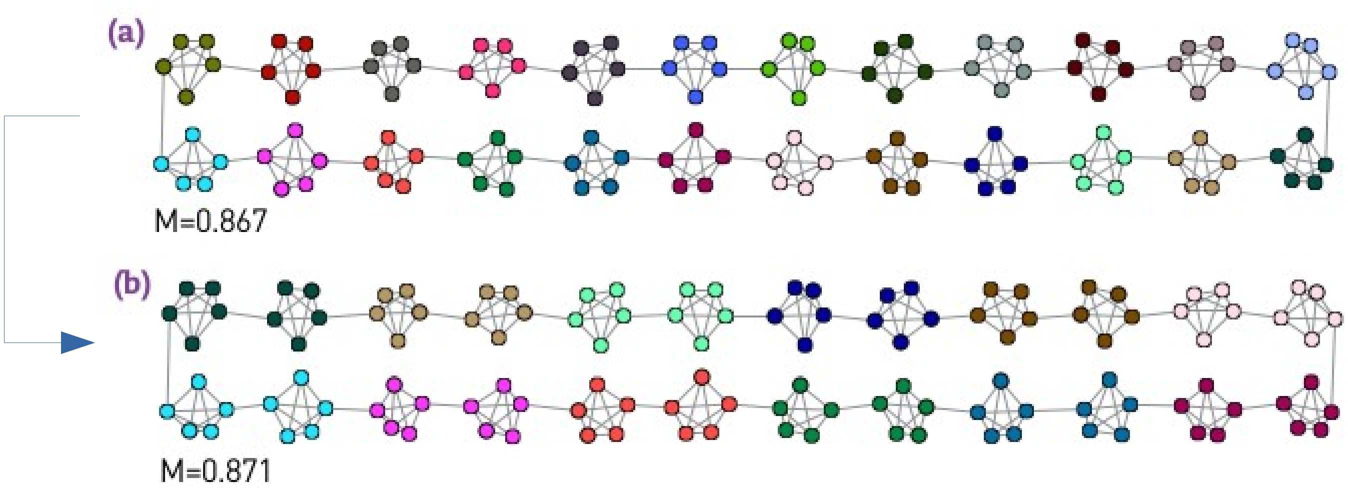
Limitación: Las variaciones en ΔM son muy pequeñas si n_c es muy grande.



Clustering en grafos

Algoritmo de Louvain

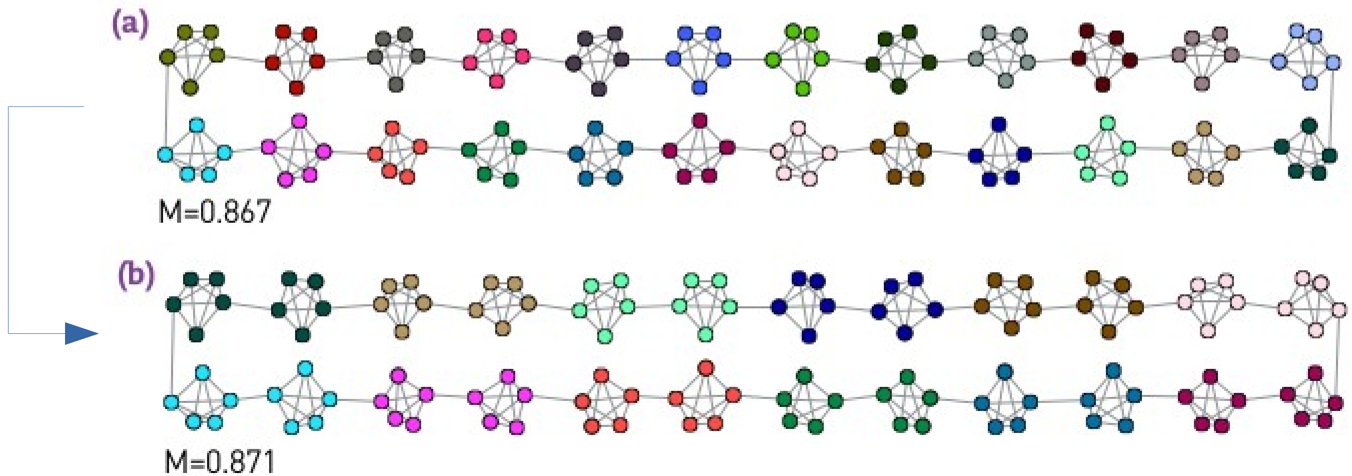
Limitación: Las variaciones en ΔM son muy pequeñas si n_c es muy grande.

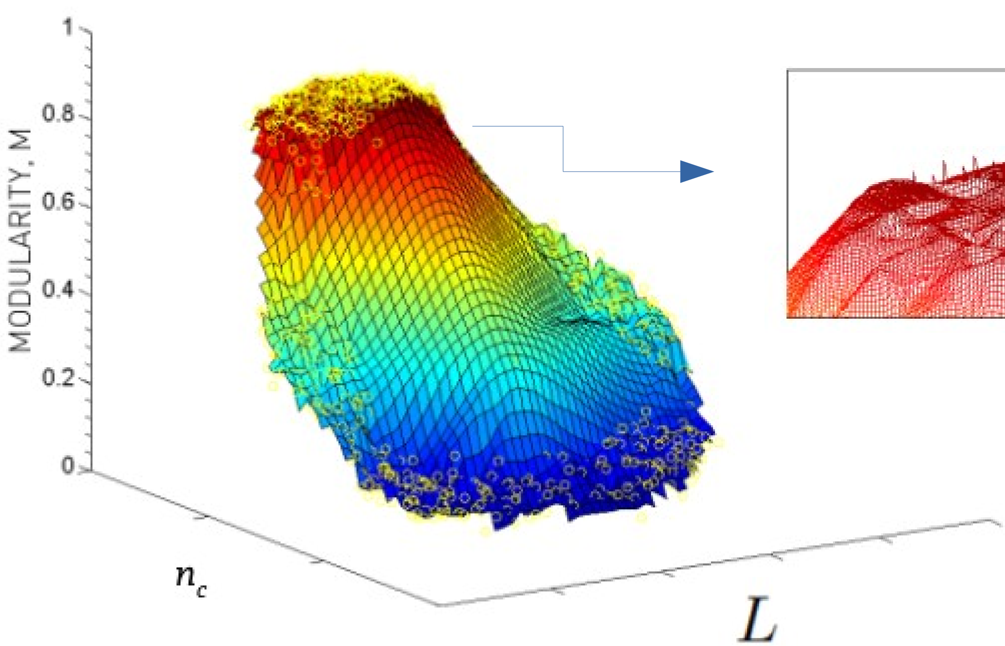


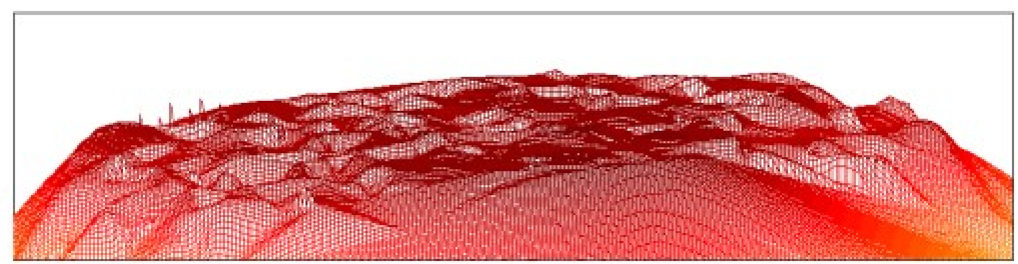
Clustering en grafos

Algoritmo de Louvain

Limitación: Las variaciones en ΔM son muy pequeñas si n_c es muy grande.







- SPECTRAL CLUSTERING -

Clustering en grafos

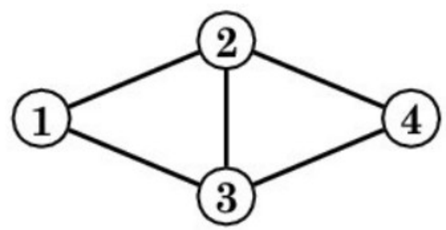
Matriz de adyacencia: $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$

Definiciones importantes:

Grado de un nodo: $d_u = \sum_{v \in V} \mathbf{A}[u, v].$

Matriz de grado: $\mathbf{D} \in \mathbb{R}^{|V| \times |V|}, \mathbf{D}[i, j] = d_{v_i}, v_i \in V$ y $i = j, \mathbf{D}[i, j] = 0, i \neq j.$

Laplaciano: $\mathbf{L} = \mathbf{D} - \mathbf{A}$



$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

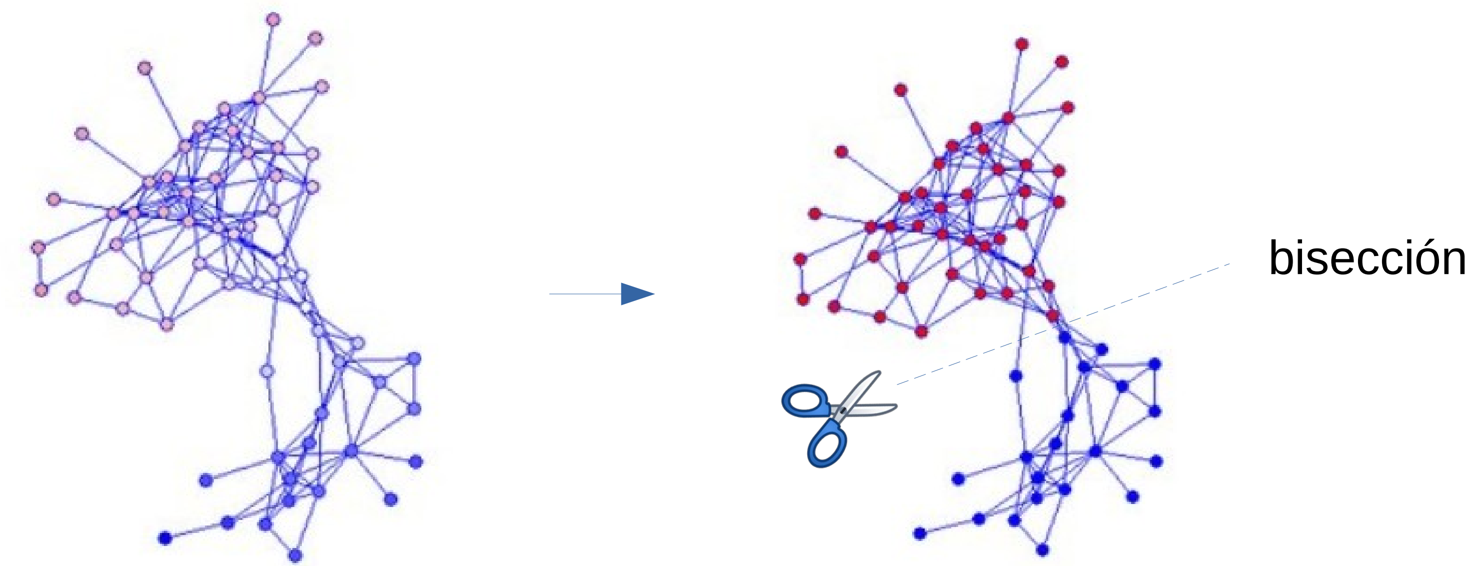
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Clustering en grafos

Bisección basada en el espectro del Laplaciano

$$
\begin{aligned}
&\min_{\mathbf{a}\in\mathbb{R}^{|\mathcal{V}|}} \ \mathbf{a}^\top \mathbf{L}\mathbf{a} \\
&\quad s.t. \\
&\qquad \mathbf{a} \perp \mathbf{1} \\
&\qquad \|\mathbf{a}\|^2 = |\mathcal{V}|.
\end{aligned}
\quad \longrightarrow \quad
\begin{cases}
u \in \mathcal{A} & \text{if } \mathbf{a}[u] \geq 0 \\
u \in \bar{\mathcal{A}} & \text{if } \mathbf{a}[u] < 0.
\end{cases}
$$



Clustering en grafos

Spectral clustering

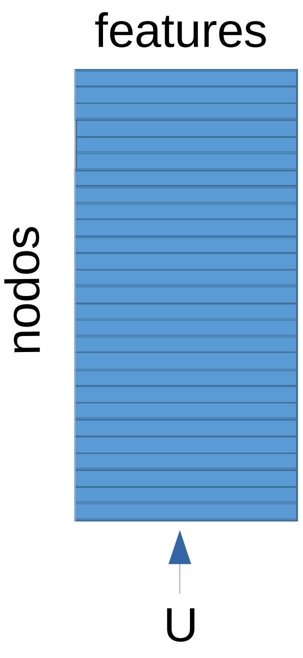
1. Find the K smallest eigenvectors of $\mathbf{L}$ (excluding the smallest): $\mathbf{e}_{|\mathcal{V}|-1}, \mathbf{e}_{|\mathcal{V}|-2}, \ldots, \mathbf{e}_{|\mathcal{V}|-K}$.

2. Form the matrix $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}|\times(K-1)}$ with the eigenvectors from Step 1 as columns.

3. Represent each node by its corresponding row in the matrix $\mathbf{U}$, i.e.,

$$\mathbf{z}_u = \mathbf{U}[u]\ \forall u \in \mathcal{V}.$$

4. Run K-means clustering on the *embeddings* $\mathbf{z}_u\ \forall u \in \mathcal{V}$.



Clustering en grafos

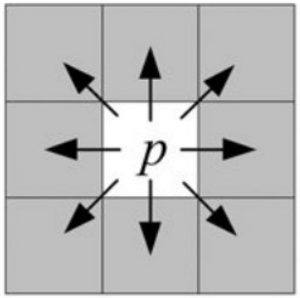
Spectral clustering en sklearn

Instalar el eigen solver para SVD amg:

```
> sudo pip install --upgrade pyamg
```

Segmentación con imagen de prueba (lena):

```
>>> import numpy, sklearn, scipy
>>> import matplotlib
>>> from matplotlib import pyplot as plt
>>> from sklearn.feature_extraction import image
>>> from sklearn.cluster import spectral_clustering
>>> lena = scipy.misc.lena()
>>> graph = image.img_to_graph(lena)
>>> graph.data = numpy.exp(-graph.data / lena.std())
>>> labels = spectral_clustering(graph, n_clusters=11, n_components=11,
       assign_labels='kmeans', eigen_solver='amg', n_init=1)
```



Usamos el solver amg para data sparse.

Clustering en grafos

Spectral clustering en sklearn

Grafo de proximidad (denso)

Clustering en digits:

```
>>> from sklearn import metrics
>>> X_dist = metrics.pairwise.pairwise_distances(X_red, metric='euclidean')
>>> X_sim = numpy.exp(-5*X_dist / X_dist.std())
>>> labels = spectral_clustering(X_sim, n_clusters=10, n_components=10,
        eigen_solver='arpack', assign_labels='kmeans', n_init=1)
```

Visualización:

```
> for i in range(X_red.shape[0]):
...     plt.text(X_red[i,0], X_red[i,1], str(y[i]),
...     color = plt.cm.spectral(labels[i]/10.),
...     fontdict = {'weight': 'bold', 'size': 8})
...
> plt.show()
```

Usamos el solver arpack ya que la data no es sparse