Spectral Clustering Analysis

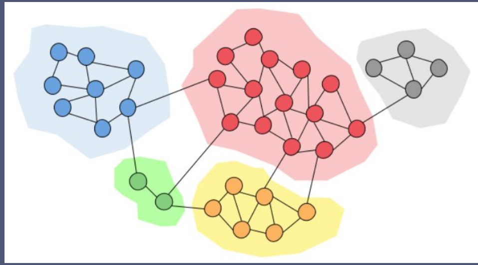
Network Statistics for Data Science (2AMS30)

Andrea Mangrella
Maarten van Sluijs
Roëlle Bänffer

Community detection

- Communities: groups of vertices having high probability of being connected to each other
- Goal: predict community labels based on the edges
- Internal and external degree
- Artificial benchmarks
- Different types of uses: Predicting behavior, Anomaly detection, Recommendation systems, Improving tasks efficiency

$$Let\ G = (V, E)\ be\ a\ graph\ with\ m\ communities\ C_1, \ldots, C_m \subset V.$$



$$d_i^{int}(C) = \sum_{j \in C} 1\left[(i, j) \in E\right] \qquad d_i^{ext}(C) = d_i - d_i^{int}$$

Technical details

A set of objects {x_1, x_2,...,x_n}
Pairwise similarity s_i,j

- Generally Gaussian: $s\left(x_i, x_j\right) = exp\left(-||x_i - x_j||^2 / \left(2\sigma^2\right)\right)$

Construct a similarity graph G = (V,E)

- ε-neighbourhood graph

- K-nearest neighbour graph

- Mutual k-nearest neighbour

Properties of G:

- Weighted adjacency matrix W
- Degree matrix D (diagonal)

Technical details

A graph's Laplacian matrix is needed for clustering:

- Unnormalized Laplacian:
  L = D - W

- Normalized Laplacians:
  Symmetric laplacian
  L_sym = I - D^-0.5WD^0.5

  Random walk laplacian
  L_rw = I - D^-1W

# of connected components = multiplicity of eigenvalue 0

Unnormalized Clustering algorithm

Given a similarity matrix S = ℝ^nxn and k clusters do:

1. Construct one of the similarity graphs
2. Compute laplacian L
3. Compute first k eigenvectors u_1,...,u_k from L
4. Construct U as:
5. Represent point y_i as row i of U
6. Cluster points y using kmeans

$$U = \begin{pmatrix} u_{11} & \cdots & u_{k1} \\ \vdots & & \vdots \\ u_{1i} & \cdots & u_{ki} \\ \vdots & & \vdots \\ u_{1n} & \cdots & u_{kn} \end{pmatrix}$$

von Luxberg (2007)

Normalized Clustering algorithm

Given a similarity matrix $S = \mathbb{R}^{nxn}$ and k clusters do:

1. Construct one of the similarity graphs
2. Compute laplacian L
3. Compute first k eigenvectors u_1,...,u_k from the problem L*u = λ*D*u
4. Construct U as before
5. Represent point y_i as row i of U
6. Cluster points y using kmeans

1. Construct one of the similarity graphs
2. Compute laplacian L
3. Compute first k eigenvectors u_1,...,u_k from L_sym
4. Form U as before
5. Form matrix T by normalizing the columns of U
6. Represent point y_i as row i of T
7. Cluster points y using kmeans

von Luxberg (2007)

Pros/Cons of Spectral Clustering:

PROS:

- Easy to compute thanks to advanced linear algebra libraries
- Often outperform standard methods
- Simple to implement

CONS:

- Not always reliable, especially when dealing with sparse graphs
- Choice of clustering algorithm for eigenvectors can introduce issues related to initialization and local optima.
- Deciding the right number of clusters: usually scientist rely on heuristics and similarity measures.

In this presentation I decided to focus on proving empirically that Spectral Clustering is heavily influenced by *Sparsity*

To prove we are going to compare the results of Spectral Clustering by using different metrics and exploiting the **Stochastic Block Method** random graph proprieties to generate dense and sparse graphs with deducible community ground truths for all nodes!

Metrics used to analyze the Results:

Mutual Information:

For both scores we are mainly going to refer to their adjusted version, as it's more reliable in a graph setting.

An *Adjusted Index* is defined as:

Mutual information is how (on average) the change we see in one variable is related to the change in another.

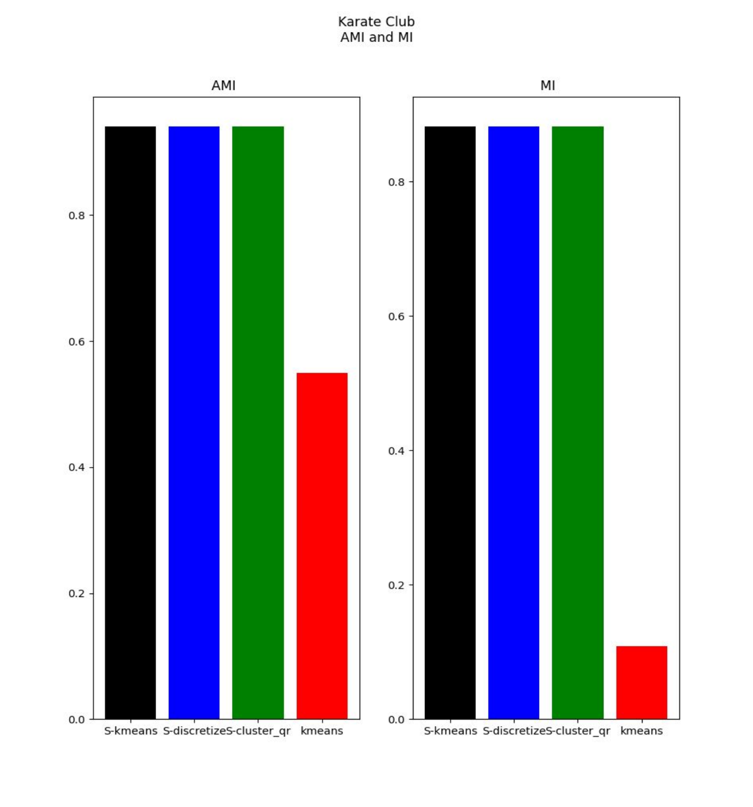
$$\text{Mutual Information} = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left[\frac{p(x,y)}{p(x)p(y)}\right]$$

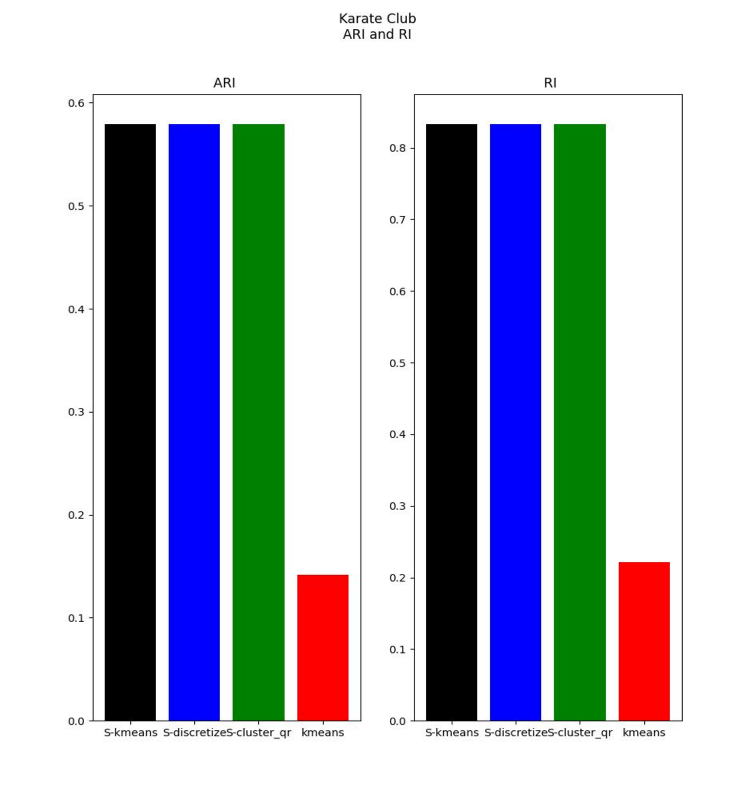
Rand Index:

The Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are correctly assigned (in respect to the ground truths.

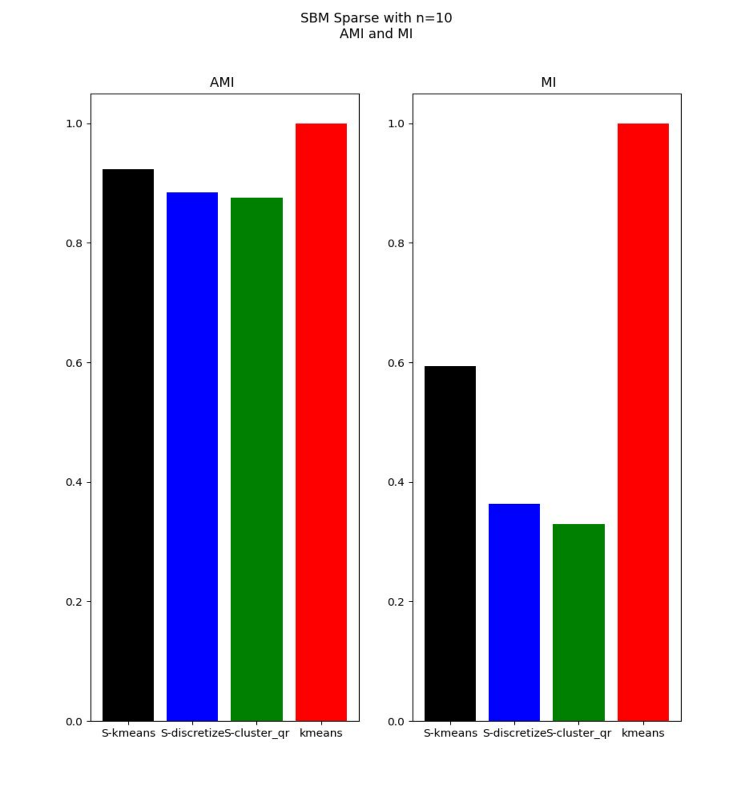
Rand Index = (number of agreeing pairs) / (number of pairs)

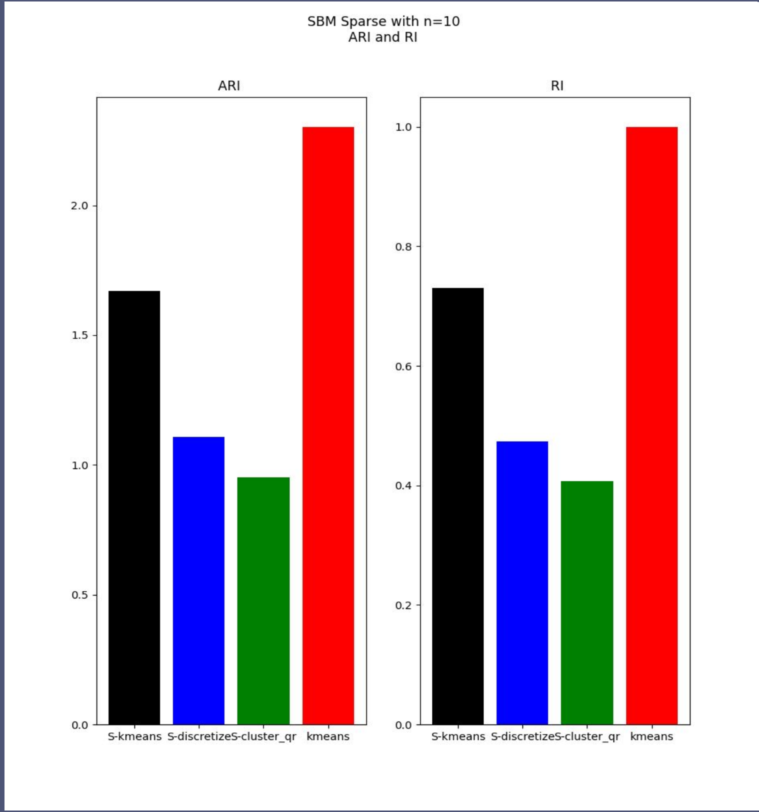
Karate Club Analysis:



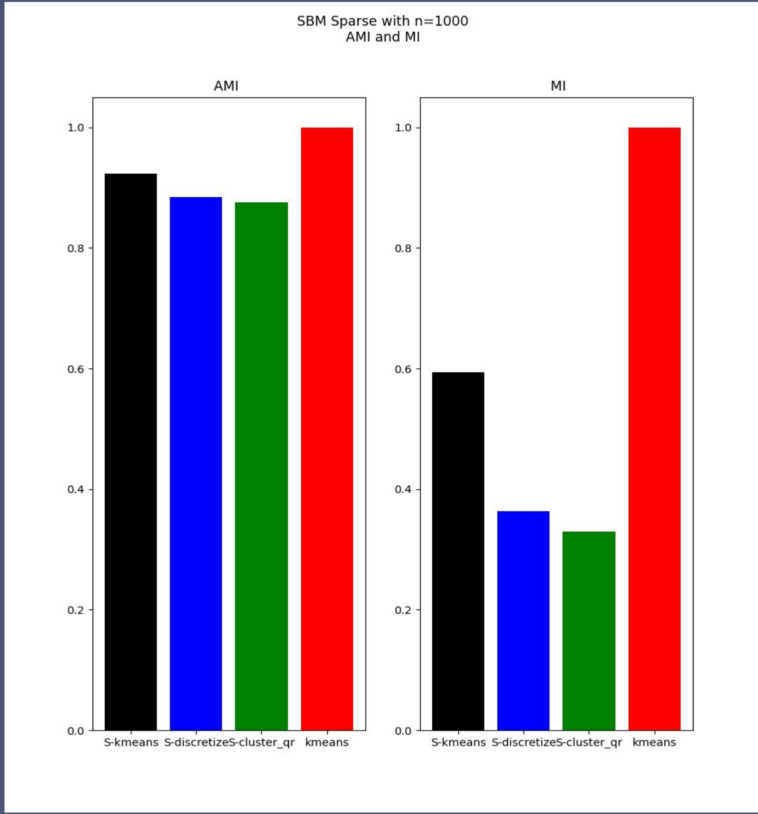


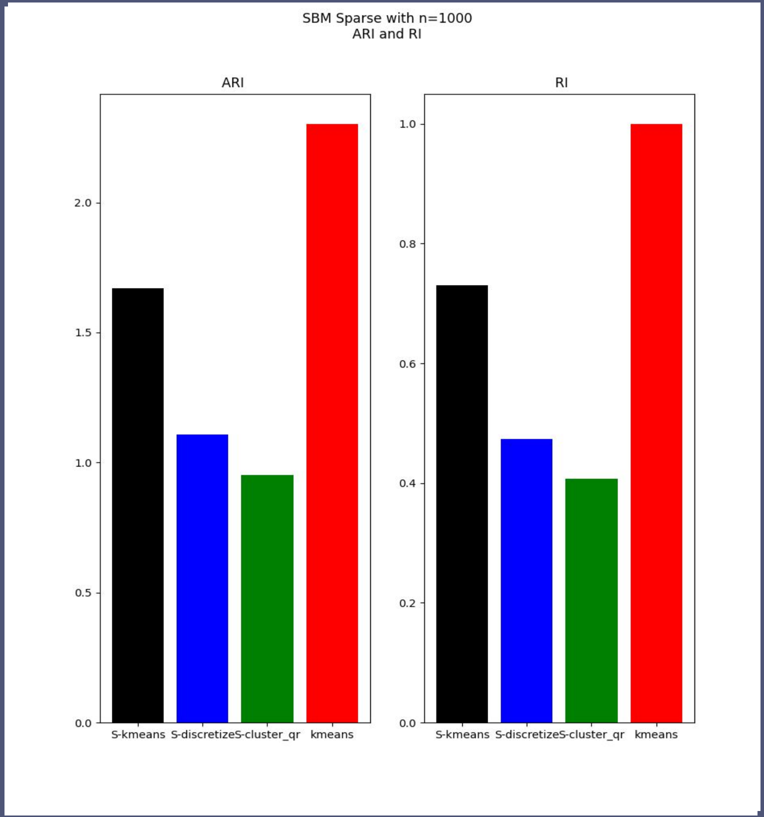
SBM Analysis (n=10):





SBM Analysis (n=1000):





Bibliography:

- Fortunato, Santo, and Darko Hric. "**Community detection in networks: A user guide.**" Physics reports 659 (2016): 1-44. Section 4.3
- Von Luxburg, Ulrike. "**A tutorial on spectral clustering.**" Statistics and computing 17 (2007): 395-416.
- Karataş, A., & Şahin, S. (2018). Application Areas of Community Detection: A Review.
- Code for plotting the results shown in the presentation