Applying bootstrap AMG in spectral clustering

Luisa Cutillo

School of Mathematics, University of Leeds
l.cutillo@leeds.ac.uk
Visiting academic, University of Sheffield
University Parthenope of Naples, IT

joint work with:
P. D'Ambra - Institute for Applied Computing, National Research Council of Italy (CNR)
and

P.S. Vassilevski - Portland State Univ. and CASC-LLNL, USA

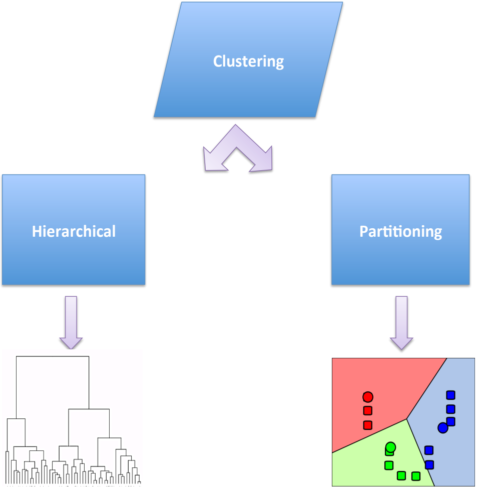
Sheffield, 6th of September , 2018

Wait a minute...did I say SPECTRAL?

Wait a minute...did I say SPECTRAL?

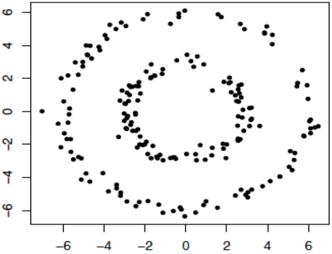
Ouch!

Clustering techniques: two categories



nonlinear separating hypersurfaces

- What if we consider non linear clusters?



- need of clustering methods that produce nonlinear separating hypersurfaces among clusters
- two big families: **kernel** and **spectral** clustering methods.

Kernel and Spectral methods

Kernel clustering

- Kernels allow to map implicitly data into a high dimensional feature space;
- computing a linear partitioning in this feature space results in a nonlinear partitioning in the input space.

Spectral clustering

- Construct a weighted graph from the initial data set;
- eigenvalue decomposition (spectrum) of the Laplacian matrix for dimensionality reduction $->$ clustering in fewer dimensions.

A unified view of Spectral and Kernel methods

- Hint: the **adjacency** between patterns in the spectral approach is the analogous of the **kernel functions** in kernel methods.
- explicit **mathematical proof** in *A survey of kernel and spectral methods for clustering* by M. Filippone *et. al.*
- In particular Kernel K-Means and Spectral clustering, with the ratio association as the objective function, are **perfectly equivalent** (shown by Dhillon *et al.*)

A unified view of Spectral and Kernel methods

- Hint: the **adjacency** between patterns in the spectral approach is the analogous of the **kernel functions** in kernel methods.
- explicit **mathematical proof** in *A survey of kernel and spectral methods for clustering* by M. Filippone *et. al.*
- In particular Kernel K-Means and Spectral clustering, with the ratio association as the objective function, are **perfectly equivalent** (shown by Dhillon *et al.*)

OMdays

Complex Networks Representation

Let $X = \{x_1, \dots, x_n\}$ and $W = (w_{ij} \geq 0)_{i,j=1,\dots,n}$ be a set of data and a matrix of similarities between pairs of vertices

Similarity Graph

$G = (V, E, W)$, a weighted undirected graph with $V = \{1, 2, \dots, n\}$ the vertex set, $E = \{(i, j) = (j, i) \mid w_{ij} > 0\}$ the edge set, and W the edge weight matrix

Complex Networks Representation

Let $X = \{x_1, \ldots, x_n\}$ and $W = (w_{ij} \geq 0)_{i,j=1,\ldots,n}$ be a set of data and a matrix of similarities between pairs of vertices

Similarity Graph

$G = (V, E, W)$, a weighted undirected graph with $V = \{1, 2, \ldots, n\}$ the vertex set, $E = \{(i,j) = (j,i) \mid w_{ij} > 0\}$ the edge set, and W the edge weight matrix

Graph Laplacian

The *Laplacian matrix of the graph G* is:

$$L = D - W \in \mathcal{R}^{n \times n},$$

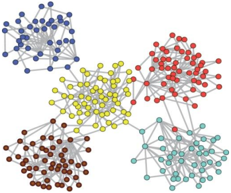
where $D = diag(d_i = \sum_{j=1}^{n} w_{ij})_{i=1,\ldots,n}$ is the diagonal matrix of weighted vertex degrees

Community detection

Communities/Clusters

Vertices groups with **dense connections within groups and only sparser connections between them**

- functional units such as cycles or pathways in *metabolic networks*
- collections of pages on a single topic on the web
- individuals contacts in social networks

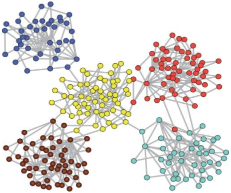


Community detection

Communities/Clusters

Vertices groups with **dense connections within groups and only sparser connections between them**

- functional units such as cycles or pathways in *metabolic networks*
- collections of pages on a single topic on the web
- individuals contacts in social networks



Community detection as mincut problem

Find a graph partition $V_1, \ldots, V_K$ minimizing:

$$RatioCut(V_1, \ldots, V_K) = \frac{1}{2} \sum_{k=1}^{K} \frac{W(V_k, \overline{V_k})}{|V_k|},$$

where $W(V_k, \overline{V_k}) = \sum_{i \in V_k, j \in \overline{V_k}} w_{ij}$ and $\overline{V_k}$ complement of V_k in V

Mincut as trace minimization problem

Given a partition $V_1, \dots, V_K$, let $h_k = (h_{1k}, \dots, h_{nk})^T$ and $H = (h_k)_{k=1,\dots,K} \in \mathcal{R}^{n \times K}$ be, where:

$$h_{ik} = \begin{cases} 1/\sqrt{|V_k|} & \text{if } x_i \in V_k \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, n; \; k = 1, \dots, K.$$

It holds:

$$RatioCut(V_1, \dots, V_K) = \sum_{k=1}^{K} (H^T L H)_{kk} = Tr(H^T L H), \text{ with } H^T H = I$$

Mincut as trace minimization problem

Given a partition $V_1, \ldots, V_K$, let $h_k = (h_{1k}, \ldots, h_{nk})^T$ and $H = (h_k)_{k=1,\ldots,K} \in \mathcal{R}^{n\times K}$ be, where:

$$h_{ik} = \begin{cases} 1/\sqrt{|V_k|} & \text{if } x_i \in V_k \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, n;\; k = 1, \ldots, K.$$

It holds:

$$RatioCut(V_1, \ldots, V_K) = \sum_{k=1}^{K} (H^T LH)_{kk} = Tr(H^T LH), \text{ with } H^T H = I$$

trace minimization problem for graph Laplacian

$$\min_{V_1,\ldots,V_K} Tr(H^T LH), \text{ subject to } H^T H = I$$

$(h_k)_{k=1,\ldots,K}$ first K eigenvectors of L are the solution

(Rayleigh-Ritz theorem)

Spectral Clustering

Using the first K eigenvectors of graph Laplacian as
low-dimension graph embedding (Euclidean) space
and applying a spatial clutering in the new space

Peng et al., *Partitioning Well-Clustered Graphs: Spectral Clustering Works!* JMLR, 2015

Our Proposal

We propose to use as graph embedding, the space spanned by the algebraically smooth vectors of the graph Laplacian, associated to an adaptive algebraic multigrid method for solving linear systems.

Algebraic MultiGrid (AMG)

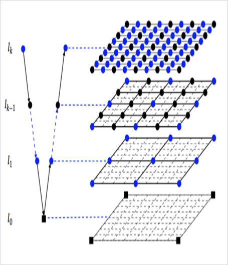
1. AMG are scalable iterative methods for solving large and sparse linear systems arising from modern applications
2. apply recursively a two-grid process: smoother iterations and a coarse-grid correction

Smooth Vectors of Graph Laplacian

$$L\mathbf{x} = \mathbf{b}, \qquad \mathbf{b} \text{ subject to} \sum_i b_i = 0$$

Algebraic MultiGrid (AMG)

1. Pre-smoothing: $\mathbf{x} = \mathbf{x} + M^{-1}(\mathbf{b} - L\mathbf{x})$
2. Residual restriction: $\mathbf{r}_c = P^T(\mathbf{b} - L\mathbf{x})$
3. Solution on coarse grid: $L_c\mathbf{e} = \mathbf{r}_c$, applying recursion
4. Error interpolation and solution update: $\mathbf{x} = \mathbf{x} + P\mathbf{e}$
5. Post-smoothing: $\mathbf{x} = \mathbf{x} + (M^T)^{-1}(\mathbf{b} - L\mathbf{x})$



Estimating smooth vectors

Laplacian graph L can be transformed to s.p.d matrix by rank-1 update:

$$L_S = L + \alpha \mathbf{q}\mathbf{q}^T, \ \alpha > 0$$

with $\mathbf{q}$ having non-zero entries $q_i = q_j = 1$ for an arbitrary edge $(i,j) \in E$

Smooth vectors can be estimated by applying iterative methods to the homogeneous system $L_S \mathbf{x} = \mathbf{0}$, starting from arbitrary $\mathbf{x}^0$:

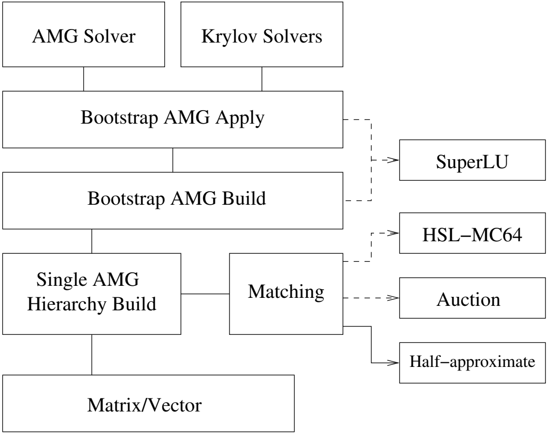
$$\mathbf{x}^\ell := (I - B^{-1} L_S)\mathbf{x}^{\ell-1} \ \ \ell = 1, \dots, \ \ell_{\max}$$

Smooth Vectors as effective embedding space

Effective embedding

algebraically smooth vectors of L_S computed by (good convergent) bootstrap AMG well capture the global connectivity of a graph

BootCMatch Software Framework



BootCMatch Software Framework.
Available at github.com/bootcmatch/BootCMatch/

D'Ambra et al., *BootCMatch: a Software package for Bootstrap AMG based on Graph Weighted Matching*, ACM TOMS, 2018

Quality Metrics for Clustering

Modularity Function

Graphs with strong community structure has large values of:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta_{v_i v_j}$$

defined as the fraction of the edges that fall within the groups minus the expected such fraction if edges were distributed at random.

Quality Metrics for Clustering

Modularity Function

Graphs with strong community structure has large values of:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta_{v_i v_j}$$

defined as the fraction of the edges that fall within the groups minus the expected such fraction if edges were distributed at random.

Variation of Information

A measure to compare partitions is the Variation of Information (VI):

$$VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}')$$

where $H(C)$ is the entropy associated with partition $\mathcal{C}$ and $I(C, C')$ is the mutual information between $\mathcal{C}$ and $\mathcal{C}'$, i.e., the information that one partition has about the other

Experimental Setting

BootCMatch+ Kmeans :

- Coarsening based on default parameters
- maximum number of bootstrap iterations (smooth vectors) *maxstages* $= 40$
- Kmeans Matlab post-processing;
- maximum modularity clustering out of 100 executions

Comparisons with **R** *igraph* package

Network comm. extract. methods:

- Louvain: a greedy modularity optimization method (Blondel et al., 2008)
- LeadingEigen: a method based on the leading eigenvector of modularity matrix (Newman, 2006)

Results on benchmarks from DC-SBM

Stochastic Block Model

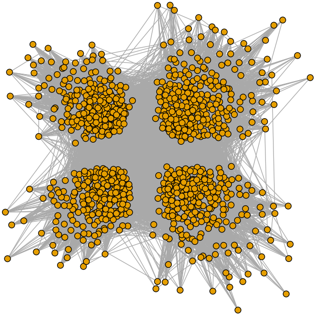
random graphs where the probability of having an edge between two nodes depends on the communities they belong to.

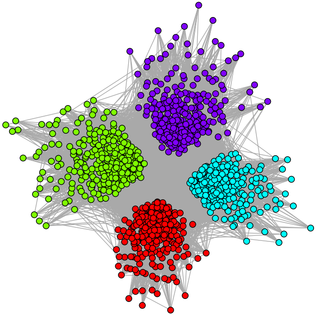
Degree Corrected SBM assumes vertex degree variability within communities, as in realistic networks.

- 144 graphs of increasing dimension $n = 1000,\ 2000,\ 3000,\ 4000$
- sparsity degree ranging in $[0.01, 0.35]$
- edge probability within each community M_{in} uniformly generated in $[0.3, 0.7]$, unique edge probability between any couple of communities $M_{out} \in [0.001, 0.8]$, corresponding to decreasing modularity
- different numbers of communities $K = 4,\ 8,\ 12,\ 16$ per each dimension, 9 graphs per each K numbered according to increasing modularity.

Cutillo et al., *An inferential procedure for community structure validation in networks, arXiv:1710.06611*

Clustering Results on DC-SBM: n=1000, K=4, Q=0.62



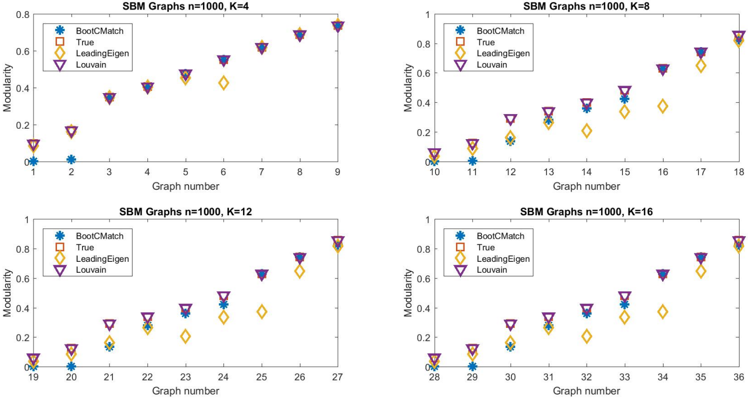


Original Graph (left); Clustering obtained with BootCMatch (right)

Output Parameters by Bootstrap AMG

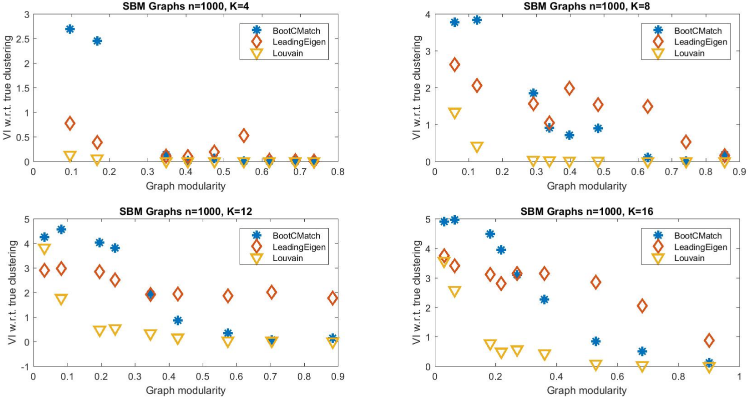
number of smooth vectors $d = 14$, corresponding to a convergence factor $\rho = 4.51 \times 10^{-9}$; computed modularity $Q = 0.6192$

Clustering Results on DC-SBM: modularity values



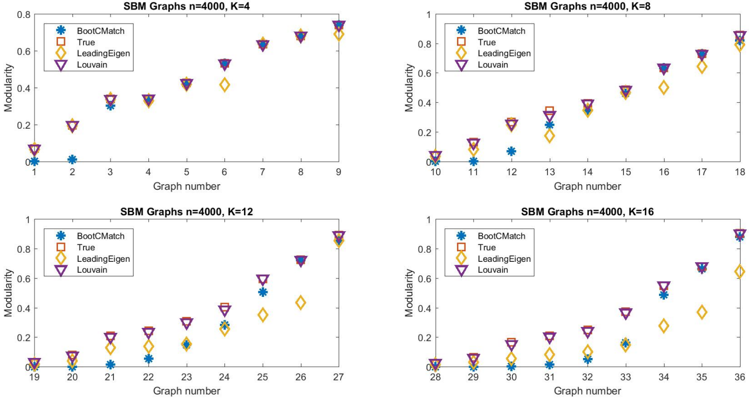
Comparison of modularity among different clustering

Clustering Results on DC-SBM: VI w.r.t. true clustering



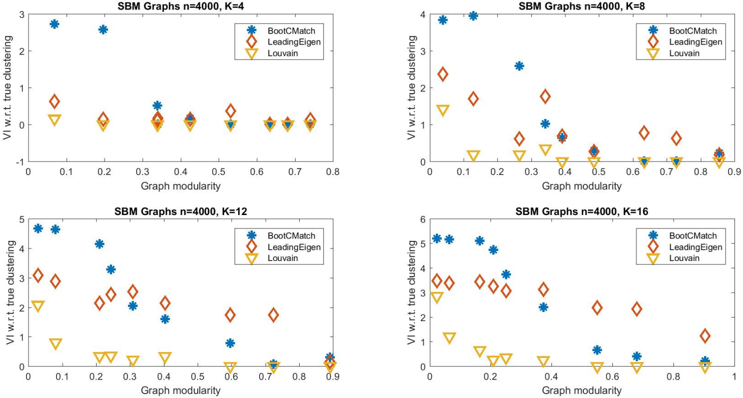
Comparison of different clustering VI w.r.t. to true clustering

Clustering Results on DC-SBM: modularity values



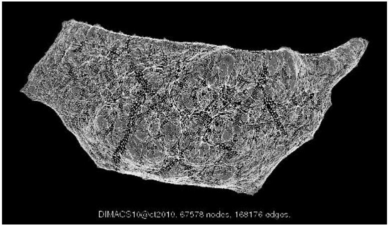
Comparison of modularity among different clustering

Clustering Results on DC-SBM: VI w.r.t. true clustering

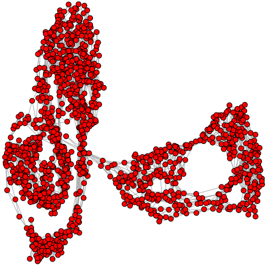


Comparison of different clustering VI w.r.t. to true clustering

Clustering Results on real networks



- ct2010, Dimacs 10th Collection: Connecticut State from Census and Tiger/Line 2010 Shapefiles, $n = 67578$ vertices and $m = 168176$ edges, sparsity 10^{-5}, min/max vertex degree 1 / 53



- immuno, *igraphdata* package collection: Immunoglobulin Interaction Network, $n = 1316$ vertices and $m = 6300$ edges, sparsity 10^{-3}, min/max vertex degree 3 / 17

Clustering Results on real networks

	BootCMatch			*LeadingEigen*			*Louvain*	
Name	*K*	*Q*	*VI*	K	Q	VI	K	Q
ct2010	39	0.954	1.57	20	0.230	4.040	80	0.964
immuno	21	0.821	1.55	12	0.863	1.03	9	0.826

Bootstrap AMG uses eigengap heuristic for setting number K of clusters: $|\sigma_{K+1} - \sigma_K| > 0.1$, with σ_r from SVD of smooth vectors.

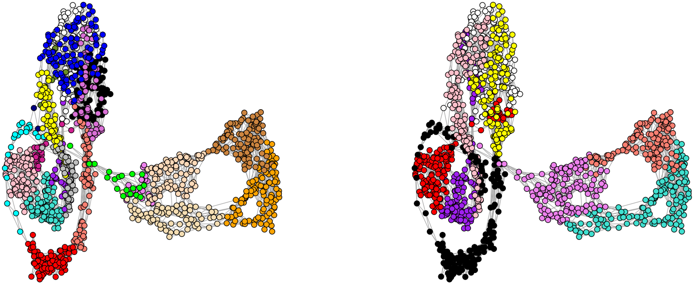


Figure: Clustering of Immuno Network. BootCMatch (left) and Louvain (right)

Some Remarks and Work in Progress

- Clustering based on bootstrap AMG gives very promising results for well clustered networks (medium/high values of modularity)
- It seems to overcome other methods based on spectral techinques (LeadinEigen)
- Spectral projection based on Bootstrap AMG has a linear complexity
- Using different spatial clustering (more reliable than K-means while dealing with small modularities) and comparisons in terms of execution times on very large networks are work in progress

Thanks for Your Attention

This work is partially performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

and

with support of the Energy oriented Center of Excellence for Computing Applications (www.eocoe.eu), funded by H2020 Program of EC, Project ID: 676629.