Sparsification of Parallel Spectral Clustering[1]

Sandrine Mouysset
Ronan Guivarch

APO Team
Institut de Recherche en Informatique de Toulouse
University of Toulouse

VECPAR'12 - 10th International Meeting High Performance Computing for Computational Science

[1] This work was performed using HPC resources from CALMIP (Grant 2012-p0989)

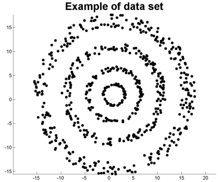
Clustering

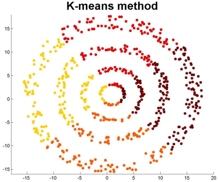
Goal:

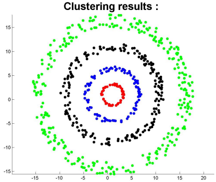
Partition a $n \times p$ data set in K clusters to obtain larger within-cluster affinity and lower between-clusters affinity

Some clustering methods based on:

- geometrical properties: K-means...
- spectral properties: Spectral clustering...







Outline

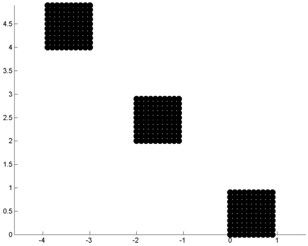
- Spectral Clustering
- Parallel Spectral Clustering
- Sparsification
- Numerical Results
- Conclusion & Future Works

Spectral Clustering

select dominant eigenvectors of a parametrized affinity matrix A in order to build a low-dimensional data space wherein data points are grouped into clusters

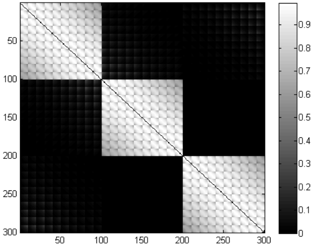
Main difficulties:

- How to (automatically) separate clusters one from the other?
  $\rightarrow$ Look for some full-unsupervising process
- How to perform clustering on large datasets (image segmentation)?
  $\rightarrow$ Parallelization using domain decomposition



(a) Data set (n=300)

1. Input: data set $\{x_i\}_{i=1..n}$



(b) Affinity matrix (step 2)

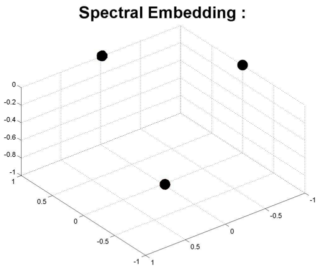
1. Input: data set $\{x_i\}_{i=1..n}$
2. Form the Gaussian affinity matrix $A \in \mathbb{R}^{n\times n}$ defined by:

$$A_{ij} = \begin{cases} e^{-\|x_i - x_j\|^2/2\sigma^2} & \text{if } i \neq j, \\ 0 & \text{otherwise} \end{cases}$$

(σ to be defined)

3. Construct the normalized matrix: $L = D^{-1}A$ with $D_{i,i} = \sum_{j=1}^{n} A_{ij}$

Algorithm *Ng, Jordan and Weiss*[2002]



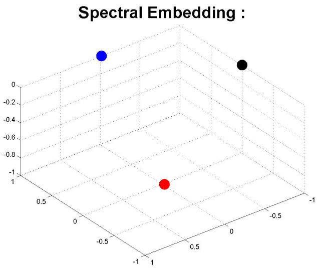
(c) Y's rows (step 4)

1. Input: data set $\{x_i\}_{i=1..n}$
2. Form the Gaussian affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by:

$$A_{ij} = \begin{cases} e^{-\|x_i - x_j\|^2/2\sigma^2} & \text{if } i \neq j, \\ 0 \text{ otherwise} & \end{cases}$$

(σ to be defined)

3. Construct the normalized matrix: $L = D^{-1}A$ with $D_{i,i} = \sum_{j=1}^{n} A_{ij}$
4. Construct the matrix $X = [X_1 X_2 .. X_k] \in \mathbb{R}^{n \times k}$ by stacking the k "largest" eigenvectors of L. Form the matrix Y by normalizing each of the X's rows. (k to be defined)



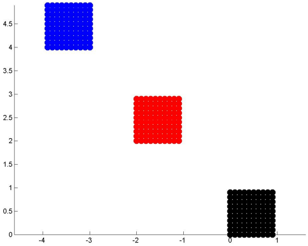
(d) Clustering of Y's rows (step 5)

1. Input: data set $\{x_i\}_{i=1..n}$
2. Form the Gaussian affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by:

$$A_{ij} = \begin{cases} e^{-\|x_i - x_j\|^2/2\sigma^2} & \text{if } i \neq j, \\ 0 \text{ otherwise} \end{cases}$$

(σ to be defined)

3. Construct the normalized matrix: $L = D^{-1}A$ with $D_{i,i} = \sum_{j=1}^{n} A_{ij}$
4. Construct the matrix $X = [X_1 X_2 .. X_k] \in \mathbb{R}^{n \times k}$ by stacking the k "largest" eigenvectors of L. Form the matrix Y by normalizing each of the X's rows. (k to be defined)
5. Treat each row of Y as a point in $\mathbb{R}^k$ and cluster them in k clusters via *K-means* method



(e) 3 well-separated clusters (step 6)

1. Input: data set $\{x_i\}_{i=1..n}$
2. Form the Gaussian affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by:

$$A_{ij} = \begin{cases} e^{-\left\|x_i - x_j\right\|^2/2\sigma^2} & \text{if } i \neq j, \\ 0 \text{ otherwise} \end{cases}$$

(σ to be defined)

3. Construct the normalized matrix: $L = D^{-1}A$ with $D_{i,i} = \sum_{j=1}^{n} A_{ij}$
4. Construct the matrix $X = [X_1 X_2 .. X_k] \in \mathbb{R}^{n \times k}$ by stacking the k "largest" eigenvectors of L. Form the matrix Y by normalizing each of the X's rows. (k to be defined)
5. Treat each row of Y as a point in $\mathbb{R}^k$ and cluster them in k clusters via *K-means* method
6. Assign the original point x_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Parameters to define

Two main problems arise:

- Choice of the Gaussian affinity parameter σ
- Estimating the number of clusters k

Gaussian affinity parameter

Heuristic based on the density of data distribution which includes both number of data n and its dimension p:

$$\sigma = \frac{\max_{1 \leq i,j \leq n} \|x_i - x_j\|}{n^{\frac{1}{p}}}$$

Number of clusters k

For different values of k

- Index the affinity matrix per cluster which provides a block structure;
- Compute the mean ratio between all off-diagonal blocks and the diagonal ones in Frobenius norms;
- Reach the optimal value k which minimizes the mean ratio.

$\rightarrow$ full-unsupervising process

Interpretation of Gaussian affinity matrix as discretization of Heat kernel

- Affinity between two data points x_i and x_j

$$A_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2} \right)$$

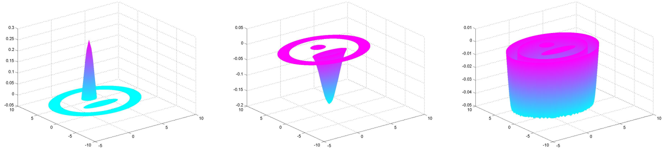
- Heat kernel in free space

$$K_t(x-y) = (4\pi t)^{-\frac{p}{2}} \exp\left(-\frac{\|x-y\|^2}{4t} \right)$$

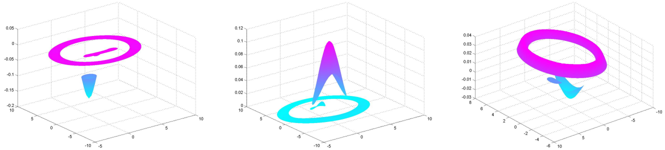
- Finite Element Theory

Interpretation of Gaussian affinity matrix as discretization of Heat kernel

Eigenfunctions for Heat equation with Dirichlet boundary conditions:



Eigenvectors of the affinity matrix A:



We can prove that[2]:

1. eigenfunctions for bounded and free space Heat equation are asymptotically close when t goes to 0,
2. difference between eigenvectors of A and discretized eigenfunctions of K_t is of an order of the distance between points inside the same cluster.

[2] Mouysset, S. and Noailles, J. and Ruiz, D., On an interpretation of Spectral Clustering via Heat equation and Finite Elements theory, Proceedings of International Conference on Data Mining and Knowledge Engineering, 2010

Interpretation of Gaussian affinity matrix as discretization of Heat kernel: towards the parallelization

Spectral Clustering as a "connected components" method

- Possibility to split the domain into subdomains;
- Applying Spectral Clustering into subdomains resumes in restricting the support of L^2 particular eigenfunctions;
- No alteration of the global partition: the eigenvectors carry the geometrical property, and so the clustering property.

→ Parallel strategy: decomposition with overlaps

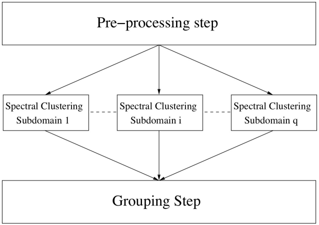


Figure: Principle of parallel Spectral clustering for q subdomains

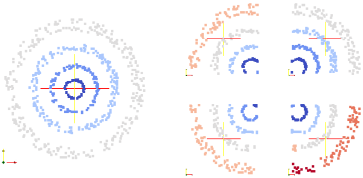


Figure: Target example: intersection and subdomains

- Total number of processes = q
- Grouping step: gather local partitions in a global partition (transitive relation[3])

[3] Mouysset, S. and Noailles, J., Ruiz, D. and Guivarch, R. On a strategy for Spectral Clustering with parallel computation, *VECPAR'10, High Performance Computing for Computational Science: 9th International Conference*, 2010.

Sparsification of Spectral Clustering

Despite the domain decomposition, the main computation cost problems are:

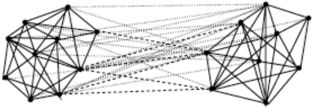
- Construction of the full affinity matrix by subdomain (memory cost):
  $\rightarrow$ Sparsification of the affinity matrix by thresholding
- Computation of their largest eigenvectors (time cost):
  $\rightarrow$ Adapted eigensolver for sparse matrix by using ARPACK

Despite the domain decomposition, the main computation cost problems are:

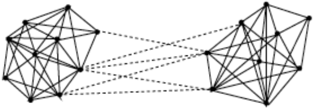
- Construction of the full affinity matrix by subdomain (memory cost):
  $\rightarrow$ Sparsification of the affinity matrix by thresholding
- Computation of their largest eigenvectors (time cost):
  $\rightarrow$ Adapted eigensolver for sparse matrix by using ARPACK

Thresholding interpretation

Strengthens the piece-wise constancy of the dominant eigenvectors and so, the affinity between points among the same cluster and the separability between clusters



(a) Without thresholding



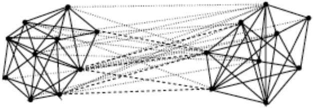
(b) With thresholding

Despite the domain decomposition, the main computation cost problems are:

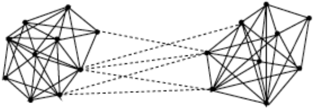
- Construction of the full affinity matrix by subdomain (memory cost):
  $\rightarrow$ Sparsification of the affinity matrix by thresholding
- Computation of their largest eigenvectors (time cost):
  $\rightarrow$ Adapted eigensolver for sparse matrix by using ARPACK

Thresholding interpretation

Strengthens the piece-wise constancy of the dominant eigenvectors and so, the affinity between points among the same cluster and the separability between clusters



(a) Without thresholding



(b) With thresholding

- we drop $A(i,j), i \neq j$ if $\|x_i - x_j\| <= \textit{threshold}$.

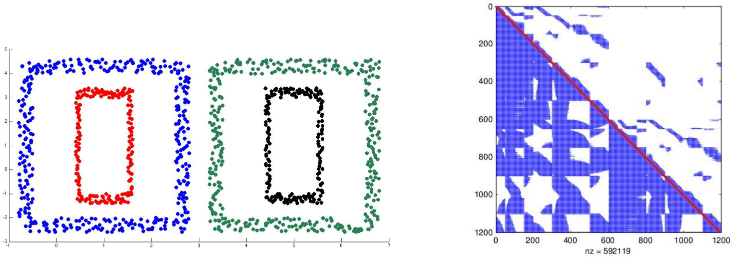


Figure: Geometrical example: affinity matrix with and without thresholding

- *threshold* $= 10 * \sigma$

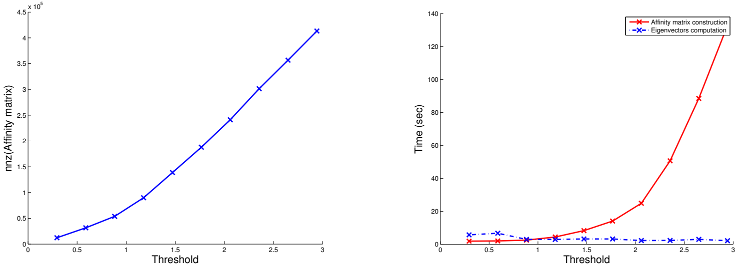


Figure: Geometrical example: Memory cost and Timings function of the threshold

construction of affinity matrix:

- $A(i,j) = e^{-\left\|x_i - x_j\right\|^2 / 2\sigma^2}$
- we drop $A(i,j)$, if $\|x_i - x_j\| <=$ *threshold*
- less computation (exponential)

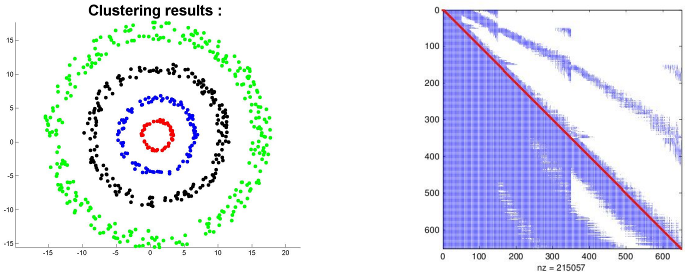
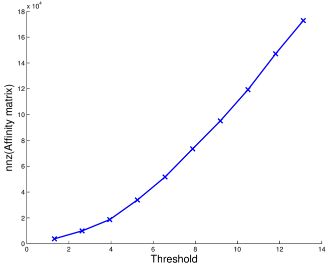


Figure: Geometrical example: affinity matrix with and without thresholding

- *threshold* $= 10 * \sigma$

Sparsification of Spectral Clustering: MATLAB results



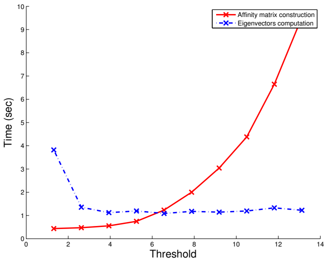


Figure: Geometrical example: Memory cost and Timings function of the threshold

Affinity measure

The affinity measure between the pixel (ij) and (rs) includes both geometrical and color in affinity definition (3D or 5D data set):

$$d\left(I_{ij}, I_{rs}\right) = \sqrt{\left(\frac{i-r}{l}\right)^2 + \left(\frac{j-s}{m}\right)^2 + \left(\frac{I_{ij}-I_{rs}}{256}\right)^2}$$

Original data

Sparsification of Parallel Spectral Clustering: parallel results[a]

[a] Hyperion cluster Altix ICE 8200 with 352 nodes (bi-Intel "Nehalem" EX quad-core)

- $threshold = factor * \sigma$
- 20 subdomains

Figure: Example of image segmentation: Original data, clustering result without and with thresholding ($factor = 1$, $n = 128612$ points)

Sparsification of Spectral Clustering: parallel results

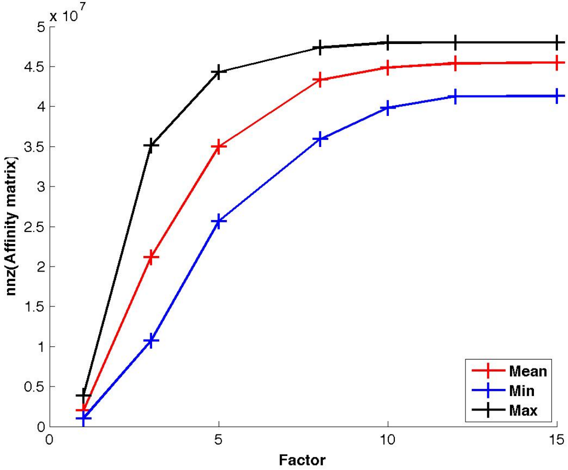


Figure: Example of image segmentation: Memory cost function of the factor

Conclusion and perspectives

Conclusion

- *Sparsification of affinity matrix construction by thresholding and using eigensolver ARPACK in the parallel version of the code;*
- *First parallel numerical results.*

Perspectives

- Tune ARPACK to optimize the time to construct the eigenvectors;
- Choose of the threshold (factor) function of the distance σ that optimizes memory and computational costs with a good segmentation result;

Figure: Original data, clustering with thresholding ($factor = 1$ and $factor = 0.9$)

- Study of the robustness: bigger images, hyper-spectral images (more than 20M points)...

Thank you for your attention.