Location Based Distributed Spectral Clustering for Wireless Sensor Networks

Presenter : Gowtham Muniraju

Gowtham Muniraju[1], Sai Zhang[1], Cihan Tepedelenlioglu[1], Mahesh K. Banavar[2], Andreas Spanias[1], Cesar Vargas-Rosales[3] and Rafaela Villalpando-Hernandez[3]

Sensip Center,[1]Arizona State University, [2]Clarkson University and [3]Tecnologico de Monterrey

Outline

Clustering

- K-means, EM & GMM
  - Uses compactness in the data to cluster than connectivity.
  - Literature: [Predd 2006, Yin 2014, Qin 2017, Zhou 2015, Forero 2012]

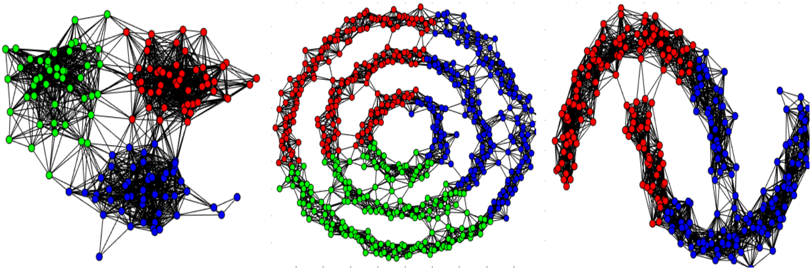


Figure: K-means type algorithm is effective for mixtures of Gaussian's but fails for arbitrary shapes such as, concentric circles, half-moons and spiral dataset.

Clustering

- Centralized Spectral Clustering
  - Effective on datasets with connectivity as well as compactness.
  - Projects the input data to Eigenspace to cluster.
  - Key works: [Ng 2001, Luxburg 2007, Shi 2000]

- Distributed Spectral Clustering ??
  - Euclidean distance matrix completion + Gradient descent [Scardapane 2016]
  - With minimal data exchange and avoid matrix completion ?

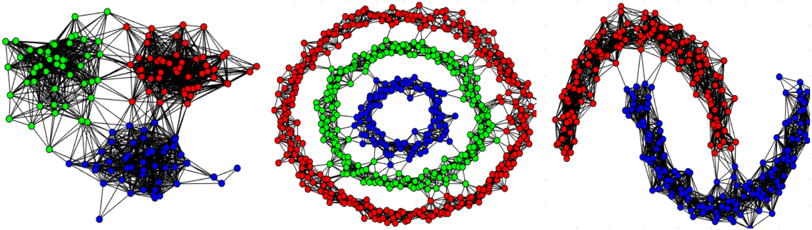
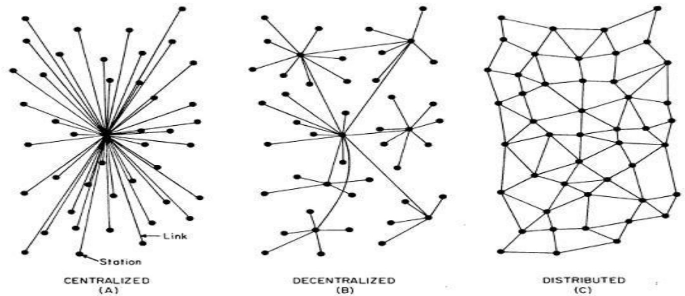


Figure: Spectral clustering works well for compact dataset like mixture od Gaussian's and also for datasets with connectivity structure, such as double-moons and concentric circles.

Motivation

- Motivation
  - Gathering data at a fusion center creates data congestion.
  - Vulnerable to cyber attacks and sensitive information loss.
  - WSN's is a source for a large set of unlabeled data.
  - Thus, appropriate labeling mechanism is required.
  - Clustering with minimal information exchange.



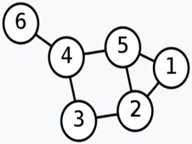
Source: Baran, Paul. *"On distributed communications networks."* IEEE transactions on Communications Systems 12, no. 1, 1964

Applications

- Potential Applications
  - Clustering and data labeling.
  - Learn the connectivity structure of the sensor deployment.
  - Selection of anchor nodes and cluster heads.
  - Limits data transmission, network traffic & contention for channel.
  - Information flow in the network.
  - Detect the change in sensors position.
- Proposed Solution
  - Fully Distributed processing.
  - Minimal information exchange.
  - Utilize the communication topology.
  - Correlation between sensors location and measurements for data labeling.

System Model

- Graph representation of distributed network
  - Distributed network with N nodes.
  - Undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, communications among neighbors.
  - Degree matrix $\mathbf{D}$: Diagonal matrix with the degrees of the nodes.
  - Adjacency matrix $\mathbf{A}$: $a_{ij} = 1$ if $\{i, j\} \in \mathbb{E}$ and $a_{ij} = 0$, otherwise.
  - Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ used to characterize network.
  - Connectivity of sensor network, $\lambda_2(\mathbf{L})$ and Fiedler vector $u_2(\mathbf{D})$

Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 2&-1&0&0&-1&0\\ -1&3&-1&0&-1&0\\ 0&-1&2&-1&0&0\\ 0&0&-1&3&-1&-1\\ -1&-1&0&-1&3&0\\ 0&0&0&-1&0&1 \end{pmatrix}$

Source: http://kuanbutts.com/2017/10/21/spectral-cluster-berkeley/

Problem Statement

- No fusion center or sink node.
- Goal : cluster the sensors in a distributed way, based on their position without sharing the location information in the network.
- DSC over K-means, EM or GMM, due to its effectiveness (as in Fig)
- Extended to clustering on data measurements assuming high correlation between sensor's location and data measurements

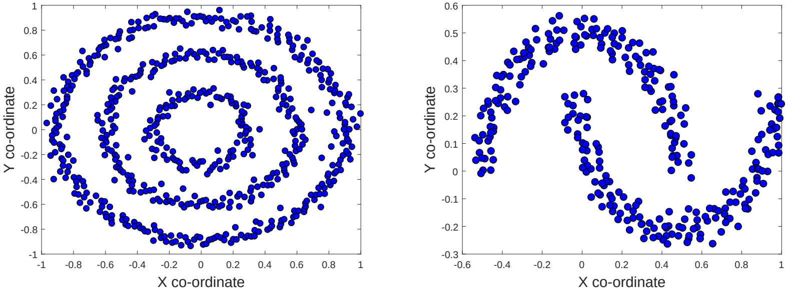
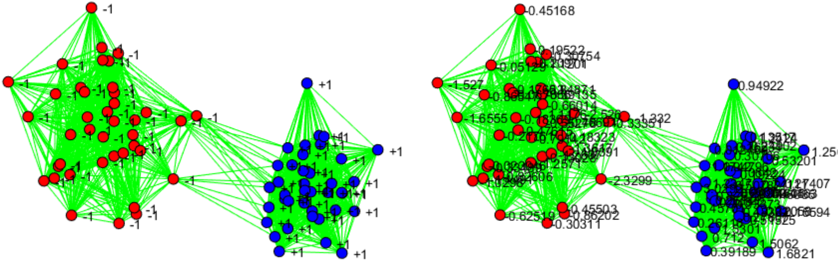


Figure: Sensors deployed in arbitrary shapes

Centralized Spectral Clustering

- SC : Approximation of a graph partitioning problem
- Prob : Find a partition of a graph such that the edges between different groups have a very low weight and edges within a group have high weight.



(a) $f \in \{+1, -1\}$ (b) $f \in \mathbb{R}$

Figure: NP hard optimization problem and its relaxed version.

Relaxed Minimization Problem

- The relaxed optimization problem is,

$$\min_{\mathbf{f}\in\mathbb{R}} \mathbf{f}^T\mathbf{L}\mathbf{f}$$

$$\text{subject to } \mathbf{f} \perp \mathbf{1}, \mathbf{f} \neq \mathbf{0}.$$

  By **Rayleigh-Ritz** theorem : choose the **f** as the eigenvector corresponding to the smallest non-zero eigenvalue of **L**, i.e *Fiedler vector*.

- **Algorithm**
  - Define the similarity graph
  - Compute the eigenvectors of K smallest eigenvalues
  - Cluster the eigenvectors

Distributed Spectral Clustering

- Assumptions
  - 1-connected component graph
  - Sensor can communicate with other sensors within a radius of ϵ
  - Absence of communication noise.
- Tasks to be computed in a distributed way !!
  - Define the similarity graph
  - Use power iteration to compute the Fiedler vector
  - Cluster the Fiedler vector
- **Similarity Graph**
  - ϵ - **neighborhood method** : nodes pairwise Euclidean distance less than ϵ are assumed connected.
  - Does not require an explicit construction, induced naturally by the ϵ and the location of the nodes.

Distributed Fiedler vector computation

- Matrix transformations and the power iteration method
- Compute the eigenvector corresponding to the second smallest eigenvalue, $u_2(L)$. [Lorenzo 2014]

$$\mathbf{Z} = \mathbf{I} - \alpha\mathbf{L} - \frac{1}{N}\mathbf{1}\mathbf{1}^T = \mathbf{W} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$$

$$\mathbf{u}^{t+1} = \frac{\mathbf{Z}\mathbf{u}^t}{||\mathbf{Z}\mathbf{u}^t||}, t \geq 0$$

where $u^{(0)}$ is an initial random vector from a continuous distribution and $0 < \alpha < 1/\lambda_N(L)$.

- **Distributed computation of Fiedler vector**

$$u^t_{avg} = \text{avgconsensus}(\mathbf{u}^t)$$

$$g^t_i = u^t_i - \alpha \sum_{j \in \mathbb{N}_i} (u^t_i - u^t_j) - u^t_{avg}$$

$$u^{t+1}_i = \frac{g^t_i}{||\mathbf{g}^t||}$$

Distributed K-means

Every node is associated with an element of the Fiedler vector. So, use a clustering algorithm on the Fiedler vector.

- Distributed K-means algorithm
  - Input: Fiedler vector $\mathbf{u_2} = [u_2^1, u_2^2, \ldots u_2^N]$, K
  - Every node generates $\boldsymbol{\mu} = [\mu_1, \ldots \mu_K]$ from rand$(-1, 1)$
  - Repeat until convergence
    - $\rho_{ki} = |u_i - \mu_k|$
    - Cluster assignment : $clusterindex = \underset{\mathrm{k}}{\operatorname{argmin}}(\rho_{ki})$
    - Update centroid : $\mathcal{U}_k = \{u_i | (i \in clusterindex = k\}$
    - $\mu_k = \operatorname{avgconsensus}(\mathcal{U}_\mathrm{k})$
    - centroid information exchange
    - Flood : $(0, \ldots, \mu_k, \ldots, 0)$
    - Update : $(0, \ldots, \mu_k, \ldots, 0) \leftarrow (\mu_1, \ldots, \mu_k, \ldots, \mu_K)$

Simulations

- Parameters
  - N $= 600$
  - K$= 3$
  - $\epsilon = 0.3$
  - $\alpha = 0.02$ as $\lambda_N^{-1}(\mathbf{L}) = 0.024$

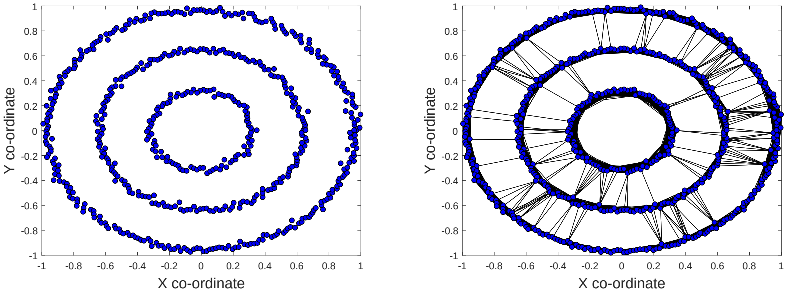
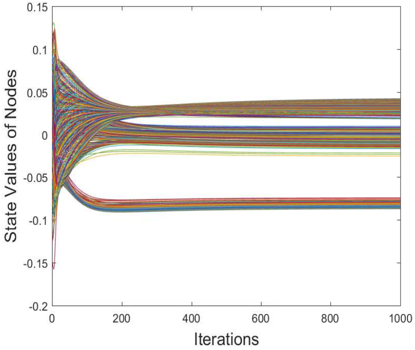


Figure: Synthetic data of 2-D sensor locations & similarity graph

Simulations



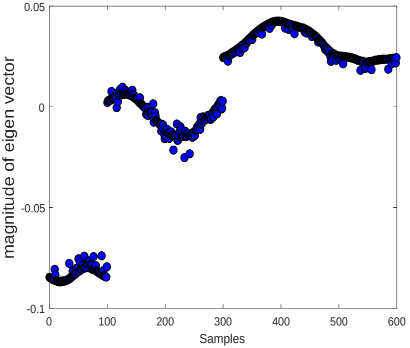


Figure: Convergence of nodes to the Fiedler vector by distributed power iteration

Simulations

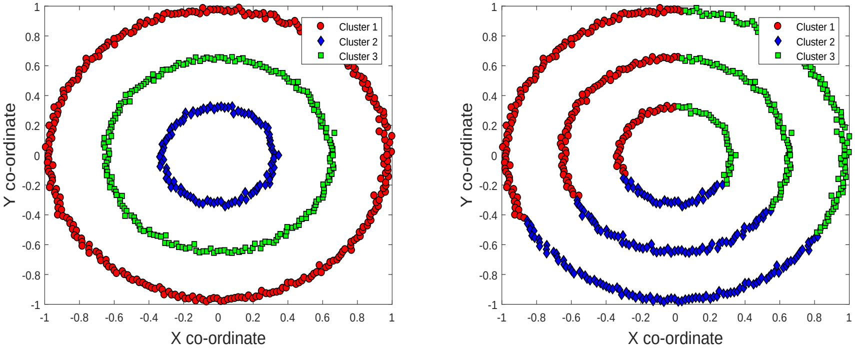


Figure: Distributed Spectral clustering vs K-means algorithm for K = 3

Extensions - Local Gaussian Kernel

- Convergence of the Fiedler vector is improved by using a local Gaussian kernel. Let z represent the location co-ordinate (x, y)

$$A_{i,j} = \begin{cases} e^{-\frac{||z_i - z_j||^2}{\sigma^2}} & \{i,j\} \in \mathbb{E} \\ 0 & \{i,j\} \notin \mathbb{E} \end{cases}$$

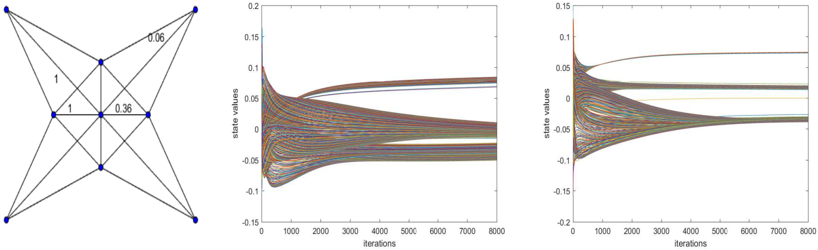


Figure: Scaling the edges by using a local Gaussian kernel is observed to improve the convergence characteristics of Fiedler vector

Extensions - DBSCAN

- DBSCAN [Ester 1996] instead of K-means
  - Input parameter to the algorithm are ϵ and *MinPts*
  - Criteria : to form a cluster a node has to have *MinPts* of nodes within ϵ radius.
  - ϵ can be a value less than communication radius.
  - Advantages
    - eliminates the input parameter K.
    - recognizes outliers.

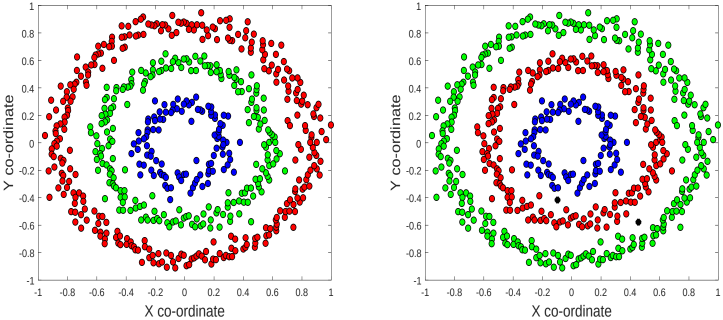


Figure: Using DBSCAN on Fiedler vector has very similar results as kmeans

Conclusion

- Designed and implemented SC in a distributed way without any fusion center in the network.
- Distributed eigenvector computation + Distributed K-means clustering, to cluster the input dataset into K groups.
- All nodes converge to a value in the Fiedler vector of the $\mathbf{L}$
- The location information is only used to establish the network topology and this information is not exchanged in the network.
- DSC usually performs better than the K-means algorithm as the eigenvector of $\mathbf{L}$ is a better feature space to cluster than the input dataset.

Main References

[1] U. von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395 - 416, Springer, 2007.

[2] P. Baran, “On distributed communications networks ,” *IEEE Transactions on Communications Systems*, vol. 12, no. 1, pp. 1 - 9, 1964.

[3] A. Y Ng, M. I. Jordan, Y. Weiss et al., “On spectral clustering: Analysis and an algorithm,” in *NIPS*, vol. 14, 2001.

[4] P. Di Lorenzo and S. Barbarossa, “Distributed estimation and control of algebraic connectivity over random graphs,” *IEEE Transactions on Signal Processing*, 2014.

[5] J. Qin, W. Fu, H. Gao, and W. X. Zheng, “ Distributed k -Means Algorithm and Fuzzy c - Means Algorithm for Sensor Networks Based on Multiagent Consensus Theory,” *IEEE Trans. on Cybernetics*, 2017.

[6] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “ Consensus and cooperation in networked multi-agent systems,” in *Proceedings of the IEEE*, 2007.


Thank you