Lecture Notes VII: Classic and Modern Data Clustering – Part II

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

November, 2020

Similarity based / graph clustering

Spectral clustering

Affinity propagation

Reading HTF Ch.: , Murphy Ch.:

Similarity based clustering

- **Paradigm:** the features we observe are measures of **similarity/dissimilarity** between pairs of data points, e.g

	points	features
Image segmentation	pixels	distance in color space or location, separated by a contour, belong to same texture
Social network	people	friends, coworkers, phone calls, emails
Text analysis	words	appear in same context

- The features are summarized by a single **similarity measure** S_{ij}
  - e.g $S_{ij} = e^{\sum_k \alpha_k \text{feature}_k(i,j)}$ for all points i, j
  - symmetric $S_{ij} = S_{ji}$
  - non-negative $S_{ij} \geq 0$
- We want to put points that are similar to each other in the same cluster, dissimilar points in different clusters
- Problem is often cast as a **graph cut** problem
  - points = graph nodes, similarity S_{ij} = weight of edge ij
  -

Paradigms for grouping

- Graph cuts
  remove some edges $\Longrightarrow$ disconnected graph
  the groups are the connected components
- By similar behavior
  nodes i, j in the same group iff i, j have the same pattern of connections w.r.t other nodes
- By Embedding
- map nodes $V = \{1, 2, \ldots, n\} \longrightarrow \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^d$ then use standard classification and clustering methods

Definitions

- $V = \{1, 2, \ldots, n\}$
- node **degree** or **volume**

$$D_i = \sum_{j \in V} S_{ij}$$

- **volume** of cluster $C \subseteq V$

$$D_C = \sum_{i \in C} D_i$$

- **cut** between subsets $C, C' \subseteq V$

$$\sum_{i \in C} \sum_{j \in C'} S_{ij}$$

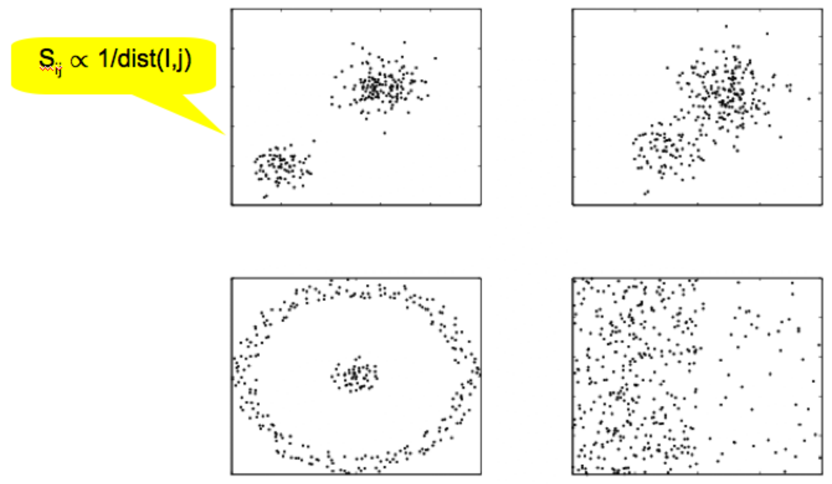
- **Multiway Normalized Cut** of a partition $\Delta = \{C_{1:K}\}$ of V

$$MNCut(\Delta) = \sum_{k=1}^{K} \sum_{k' \neq k} \frac{Cut(C_k, C_{k'})}{D_{C_k}}$$

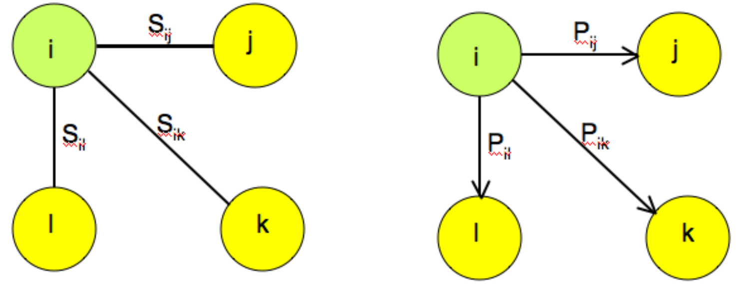
in particular, for $K = 2$,

$$MNCut(C, C') = Cut(C, C') \left(\frac{1}{D_C} + \frac{1}{D_{C'}} \right)$$

Motivation for MNCut



A random walks view



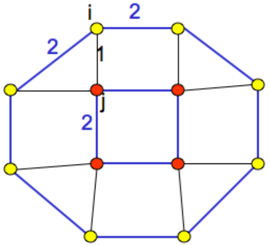
- Define

$$P_{ij} = \frac{S_{ij}}{D_i} \quad \text{for all } i, j \in V$$

- in matrix notation $P = D^{-1}S$ where $P = [P_{ij}]$, $D = \mathrm{diag}(D_1, \ldots D_n)$
- P defines a **random walk** over the graph nodes V

Grouping from the random walks point of view

- Idea: group nodes together if they transition in the same way to other clusters



$$P_{i,red} = Pr[\, i \rightarrow red \,|\, i] = \sum_{j \in red} P_{ij}$$

i	$P_{i,red}$	$P_{i,yellow}$
🟡	1/5	4/5
🟡	1/5	4/5
🟡	1/5	4/5
🟡	1/5	4/5
🟡	1/5	4/5
🟡	1/5	4/5
🟡	1/5	4/5
🟡	1/5	4/5
🔴	2/3	1/3
🔴	2/3	1/3
🔴	2/3	1/3
🔴	2/3	1/3

... is the same as grouping by embedding

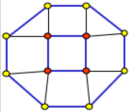
- **embedding** of V = mapping from V into $\mathbb{R}^d$
- Wanted: similar points embedded near each other
  ideally, points in the same cluster mapped to the same point in $\mathbb{R}^d$

Another look at $P_{i,C}$

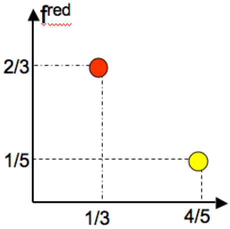
$P_{\cdot,red} \equiv f^{red}$

$P_{\cdot,yel} \equiv f^{yel}$

a **piecewise constant** function

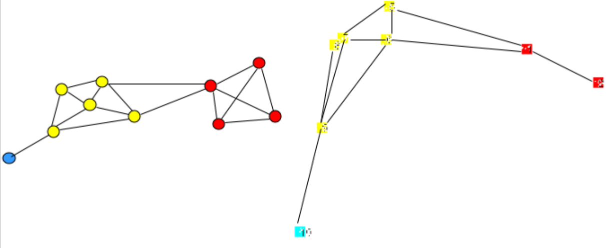


i	$P_{i,red}$	$P_{i,yellow}$
	1/5	4/5
	1/5	4/5
	1/5	4/5
	1/5	4/5
	1/5	4/5
	1/5	4/5
	1/5	4/5
	1/5	4/5
	2/3	1/3
	2/3	1/3
	2/3	1/3
	2/3	1/3



Some questions

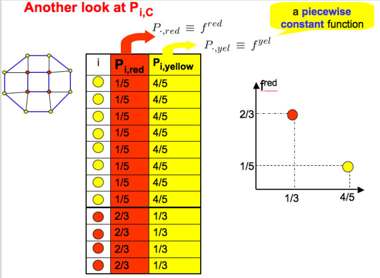
- Not all graphs embed perfectly



- How many dimensions do we need?
- Nice, but we need to know the clusters in advance. . .

Lumpability

- A vector v is **piecewise constant** w.r.t a clustering Δ iff $v_i = v_j$ whenever i, j in same $C \in \Delta$

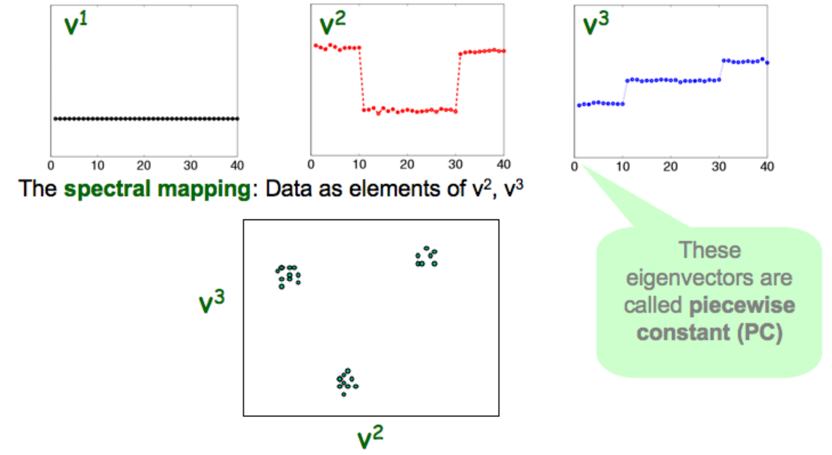


i	$P_{i,red}$	$P_{i,yellow}$
yellow	1/5	4/5
yellow	1/5	4/5
yellow	1/5	4/5
yellow	1/5	4/5
yellow	1/5	4/5
yellow	1/5	4/5
yellow	1/5	4/5
yellow	1/5	4/5
red	2/3	1/3
red	2/3	1/3
red	2/3	1/3
red	2/3	1/3

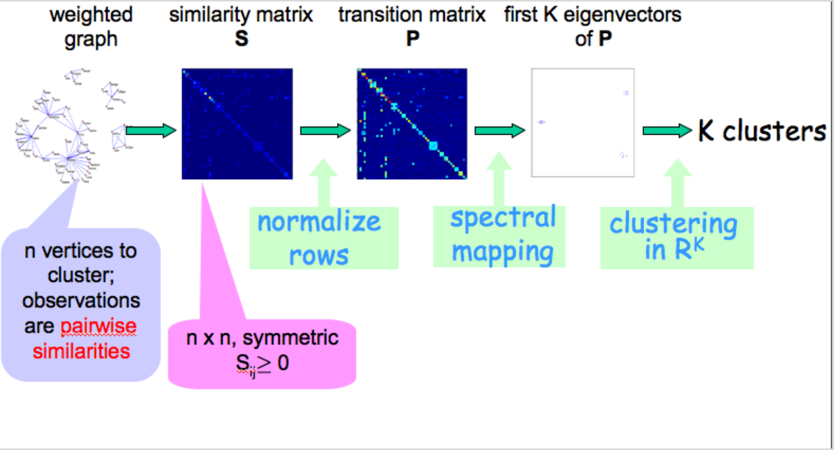
- **Theorem [Lumpability]**[Meila&Shi 2001] Let S be a similarity matrix and Δ a clustering with K clusters. Then P has K **piecewise constant** eigenvectors w.r.t Δ **iff**

$$\sum_{j \in C'} P_{ij} = R_{CC'} \quad \text{whenver } i \in C, \text{ for all } C, C' \in \Delta$$

The spectral mapping



Spectral clustering in a nutshell



Spectral clustering

An algorithm based on [Meilă and Shi, 2001b] and [Ng et al., 2002].

Spectral Clustering Algorithm

Input Similarity matrix S, number of clusters K

1. Transform S: Set $D_i = \sum_{j=1}^{n} S_{ij}$, $j = 1:n$ the **node degrees**.
   Form the **transition matrix** $P = [P_{ij}]_{ij=1}^{n}$ with

   $$P_{ij} \leftarrow S_{ij}/D_i, \text{ for } i,j = 1:n$$

2. Compute the largest K eigenvalues $\lambda_1 = 1 \geq \lambda_2 \geq \ldots \geq \lambda_K$ and eigenvectors $\mathbf{v}_1, \ldots \mathbf{v}_K$ of P.
3. Embed the data in principal subspace Let $V = [\, \mathbf{v}_2 \; \mathbf{v}_3 \; \ldots \; \mathbf{v}_K \,] \in \mathbb{R}^{n \times K}$, $\mathbf{x}_i \leftarrow$ i-th row of V.
4. (orthogonal initialization) Find K initial centers by
   4.1 take μ_1 randomly from $\mathbf{x}_1, \ldots \mathbf{x}_n$
   4.2 for $k = 2, \ldots K$ set $\mu_k = \operatorname{argmin}_{\mathbf{x}_i} \max_{k' < k} \mu_{k'}^T \mathbf{x}_i$.
5. Run the K-means algorithm on the "data" $\mathbf{x}_{1:n}$ starting from the centers $\mu_{1:K}$.

Properties of spectral clustering

- Arbitrary cluster shapes (main advantage)
- Elegant mathematically
- Practical up to medium sized problems
  - Running time (by Lanczos algorithm) $\mathcal{O}(nk)$/iteration.
- Works well when K known, not too large
  estimating K [Azran and Ghahramani, 2006]
- Depend heavily on the similarity function (main problem)
  learning the similarities
  [Meilă and Shi, 2001a],[Bach and Jordan, 2006],[Meilă et al., 2005],[Shortreed and Meilă, 20
- Outliers become separate clusters (user must adjust K accordingly!)
- Very popular, many variants which aim to improve on the above
  Diffusion maps [Nadler et al., 2006]: normalize the eigenvectors $\lambda_k^t v^k$
- Practical fix, when K large: only compute a fixed number of eigenvectors $d < K$. This avoids the effects of noise in lower ranked eigenvectors

Affinity propagation

- **Idea** Each item $i \in \mathcal{D}$ finds an **exemplar** item $k \in \mathcal{D}$ to "represent" it
- Affinity Propagation is to spectral clustering what Mean Shift is to K-means
- number of exemplars not fixed in advance
- quantities of interest
  - similarities s_{ij}, $i \neq j$ (given)
  - **availability** a_{ik} of k for i = how much support there is from other items for k to be an exemplar
  - **responsibility** r_{ik} that measures how fit is k to represent i, as compared to other possible candidates k'.
  - diagonal elements s_{ii} represent **self-similarities**
    - larger s_{ii} $\Rightarrow$ more likely i will become an exemplar $\Rightarrow$ more clusters

Affinity Propagation

Affinity Propagation Algorithm [Frey and Dueck, 2007]

Input Similarity matrix $S = [s_{ik}]_{ik=1}^{n}$, parameter $\lambda = 0.5$

Iterate the following steps until convergence

1. $a_{ik} \leftarrow 0$ for $i, k = 1 : n$
2. for all i
   - 2.1 Find the best exemplar for i: $s^* \leftarrow \max_k (s_{ik} + a_{ik})$, $A_i^* \leftarrow \underset{k}{\operatorname{argmax}} \, (s_{ik} + a_{ik})$ (can be a set of items)
   - 2.2 for all k update responsibilities

$$r_{ik} \leftarrow \begin{cases} s_{ik} - s^*, & \text{if } k \notin A_i^* \\ s_{ik} - \max_{k' \notin A_i^*} (s_{ik} + a_{ik}) & \text{otherwise} \end{cases}$$

3. for all k update availabilities
   - 3.1 $a_{kk} \leftarrow \sum_{i \neq k} [r_{ik}]_+$ where $[r_{ik}]_+ = r_{ik}$ if $r_{ik} > 0$ and 0 otherwise.
   - 3.2 for all i, $a_{ik} \leftarrow \min\{0, \; r_{kk} + \sum_{i' \neq i, k} [r_{i'k}]_+\}$
4. Assign an exemplar to i by $k(i) \leftarrow \underset{k'}{\operatorname{argmax}} (r_{ik'} + a_{ik'})$

Azran, A. and Ghahramani, Z. (2006).
Spectral methods for automatic multiscale data clustering.
In *Computer Vision and Pattern Recognition*, pages 190–197. IEEE Computer Society.

Bach, F. and Jordan, M. I. (2006).
Learning spectral clustering with applications to speech separation.
Journal of Machine Learning Research, 7:1963–2001.

Frey, B. J. and Dueck, D. (2007).
Clustering by passing messages between data points.
Science, 315:973–976.

Meilă, M. and Shi, J. (2001a).
Learning segmentation by random walks.
In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13, pages 873–879, Cambridge, MA. MIT Press.

Meilă, M. and Shi, J. (2001b).
A random walks view of spectral segmentation.
In Jaakkola, T. and Richardson, T., editors, *Artificial Intelligence and Statistics AISTATS*.

Meilă, M., Shortreed, S., and Xu, L. (2005).
Regularized spectral learning.
In Cowell, R. and Ghahramani, Z., editors, *Proceedings of the Artificial Intelligence and Statistics Workshop(AISTATS 05)*.

Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I. (2006).
Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators.
In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 955–962, Cambridge, MA. MIT Press.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002).
On spectral clustering: Analysis and an algorithm.
In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Shortreed, S. and Meilă, M. (2005).
Unsupervised spectral learning.
In Jaakkola, T. and Bachhus, F., editors, *Proceedings of the 21st Conference on Uncertainty in AI*, pages 534–544, Arlington,Virginia. AUAI Press.

Verma, D. and Meilă, M. (2003).
A comparison of spectral clustering algorithms.
TR 03-05-01, University of Washington.