

# Coring method for clustering a graph

Thang Le  
Department of Computer Science  
Rutgers University

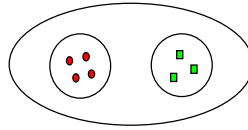
2007

## Outline

- Introduction
- Layered clustering by quasi-concave set function optimization
- The coring method for clustering an undirected graph
- Preliminary experimental results
- Method advantages and issues
- Future work
- Conclusions

## Introduction

- **Clustering:** Partition a set into subsets such that every element of a subset is more similar to other elements of the same subset and less similar to elements of other subsets



- Many clustering applications in data mining, bioinformatics, computer vision, market research, VLSI design, etc.
- Graph-based clustering methods work on graphs – a popular and powerful form of data representation

3

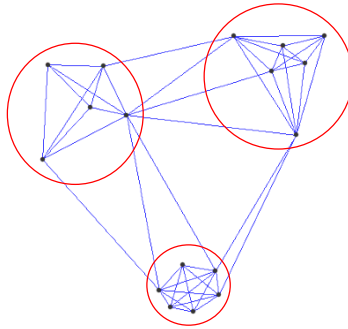
## Proximity graphs

- Undirected graph  $G = (V, E, W)$ .  
 $i, j \in V$ :  $w_{ij} \in W$  is the weight of edge  $(i, j) \in E$ ,  $w_{ij} = 0$  if  $(i, j) \notin E$ ,  
 $w_{ij} \in \{0, 1\}$  for unweighted graphs,  $w_{ij} \geq 0$  for weighted graphs
- **Proximity graphs:** Edge weight  $w_{ij}$  represents the degree of **similarity** between data objects corresponding to nodes  $i$  and  $j$
- Proximity graph is a natural representation for data in fields such as social networks, interaction networks, or web hyperlink data
- If data is represented in a feature space, a proximity graph can be derived from pairwise distances between data points in the feature space, so graph-based methods can be applied for data analysis

4

## Graph clustering

- Discover strongly or densely connected subgraphs that are weakly or sparsely connected to each other
- Clustering a proximity graph results in subgraphs corresponding to subsets of similar objects
- Many practical approaches: Layered clustering, spectral clustering, random walks, thresholding MST, maximum cliques, etc.

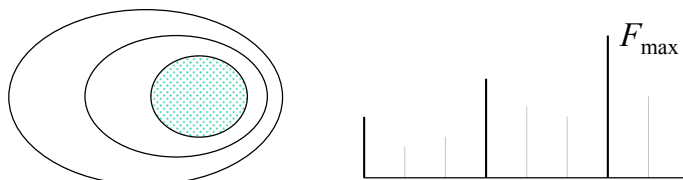


5

## Layered clustering

[B. Mirkin, I. Muchnik]

- The density of a set is measured by a set function  $F$  which is the minimum value of linkages between the set and its elements
- The subset with the global maximum value of  $F$  is referred to as the densest subset of the set (the maximizer)
- Layered clusters are chain-nested subsets such that the  $F$  value of each subset is greater than the  $F$  values of subsets which are not part of it



6

## Clustering by quasi-concave function optimization

- Linkage function  $\pi(i, H)$  measures the similarity of a node  $i$  with a subgraph  $H$ . Generally,  $\pi(i, H) = \sum_{j \in H} w_{ij}$
- Set function  $F(H)$  measures the proximity of the nodes of  $H$ :  

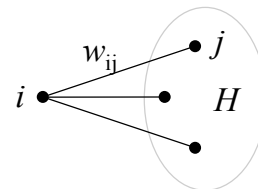
$$F(H) = \min_{i \in H} \pi(i, H)$$
- Subgraph  $H^*$  is a **maximizer** for a graph if  $H^* = \operatorname{argmax}_{H \subseteq V} F(H)$
- The largest maximizer is discovered and extracted as one cluster. Repeat the extraction on the remaining graph until it is empty
- The largest  $H^*$  can be found efficiently if  $F(H)$  is ‘quasi-concave’:  
i.e.,  $\forall H_1, H_2: F(H_1 \cup H_2) \geq \min(F(H_1), F(H_2))$
- $F(H)$  is quasi-concave iff  $\pi(i, H)$  is monotonically increasing:  
i.e.,  $\forall i \in H_1 \subseteq H_2: \pi(i, H_1) \leq \pi(i, H_2)$

7

## Examples of monotone linkage functions

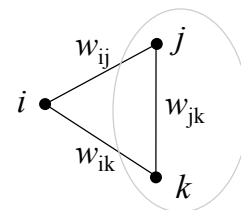
- Pairwise-based linkage functions:

- $\pi(i, H) = \sum_{j \in H} w_{ij}$
- $\pi(i, H) = \max_{j \in H} w_{ij}$



- Consistent triplet linkage function [J. Yun et al.]:

- To exploit higher-level relation among data objects
- $i, j, k \in V$ :  $\{i, j, k\}$  is a consistent triplet if  $w_{ij}, w_{ik}, w_{jk} \geq \theta > 0$  and  $i, j, k$  satisfy an additional constraint such as they share a common property, i.e., higher-level relation



- $\pi(i, H) =$  number of consistent triplets involving  $i$  in  $H$

8

## Procedure to find the largest maximizer

$t \leftarrow 1$

$H_t \leftarrow V$

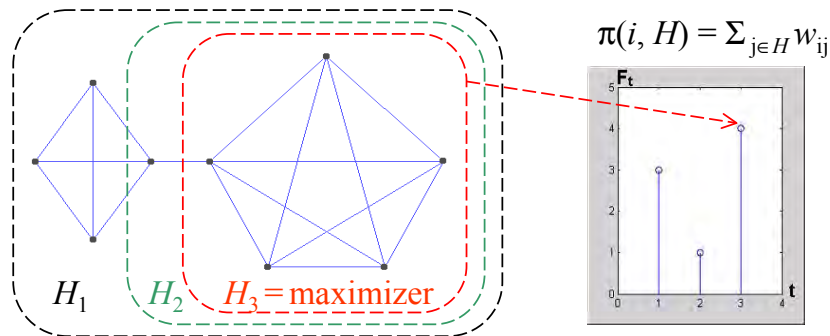
**While**  $H_t$  is nonempty

$F_t \leftarrow \min_{i \in H_t} \pi(i, H_t)$

$H_{t+1} \leftarrow H_t - \{i \mid i \in H_t \wedge \pi(i, H_t) = F_t\}$

$t \leftarrow t + 1$

**Return** the maximizer is the largest  $H_{t^*}$  such that  $F_{t^*} = \max_t F_t$



9

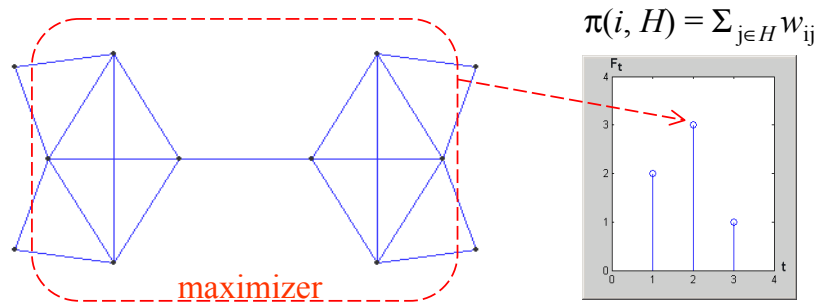
## Time complexity

- A straightforward implementation of the procedure runs in  $O(|V|^2 \tau)$  where  $\tau$  is time for evaluating linkage function  $\pi$  [A. Vashist et al.]
- The overall complexity to extract all clusters in a graph is  $O(n|V|^2 \tau)$  where  $n$  is the number of times applying the procedure
- With  $\pi(i, H) = \sum_{j \in H} w_{ij}$  the procedure running time can be reduced:
  - $O(|E| + |V|) = O(|E|)$  for unweighted graphs using count sort
  - $O(|E| + |V| \log |V|)$  for weighted graphs using Fibonacci heaps

10

## Limitations of clustering based on the maximizer

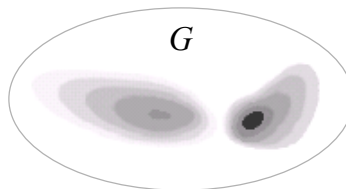
- For highly connected graphs, the maximizer covers all the graph
- Expected clusters (maximal cliques) are inside the maximizer
- ‘Orphan’ nodes are left out when the maximizer is removed
- Lack of a control parameter to cluster a dataset at different scales



11

## Dense core assumption

- **Assumption:** *Given a graph, every cluster in it has one dense core surrounded by sparser regions*



- **Cluster core:** The dense core of a cluster
- **Core nodes:** Nodes in cluster cores
- **Core set:** The set of core nodes
- **Core graph:** The subgraph consisting of core nodes

12

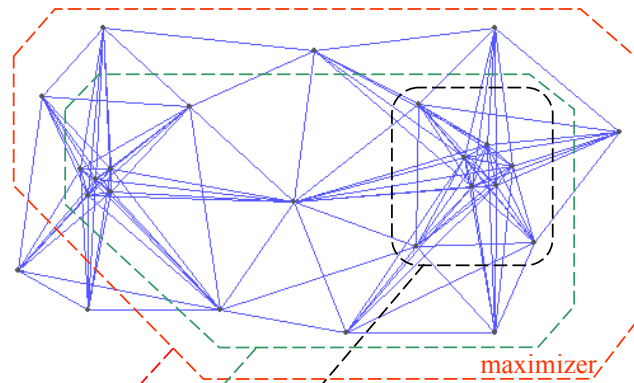
## Example: Weighted graph with cluster cores

A weighted graph of 25 nodes and 126 edges:

$$w_{ij} = \frac{[\max_{x,y \in V} d(x,y)] - d(i,j)}{\max_{x,y \in V} d(x,y)}$$

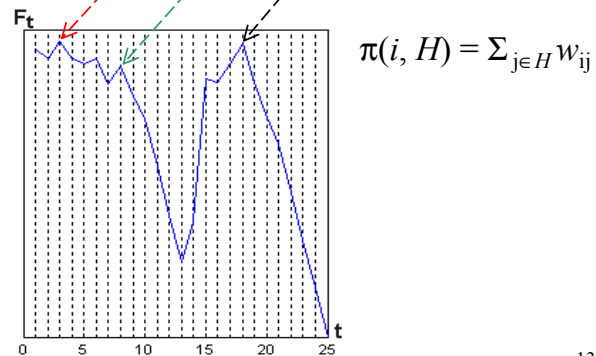
where  $d(x,y)$  is Euclidean distance between  $x$  and  $y$

Therefore,  $w_{ij} \in [0, 1]$



### Important properties:

- $F_t$  sequence progresses from outer to inner layers
- Large downhill sections correspond to cluster cores

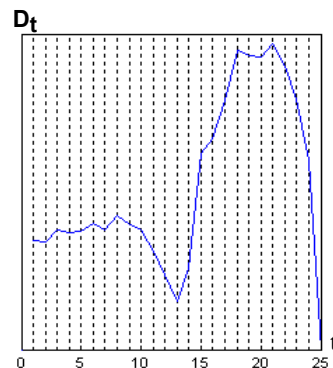
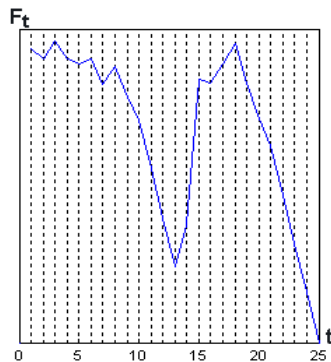


13

## Minimum density measurement

- $\pi(i, H) = \sum_{j \in H} w_{ij}$  can measure the density at node  $i$  in subgraph  $H$
- Define  $D(H)$  to measure the minimum density of subgraph  $H$   
 $D(H)$  is  $F(H)$  normalized by the size of  $H$ :

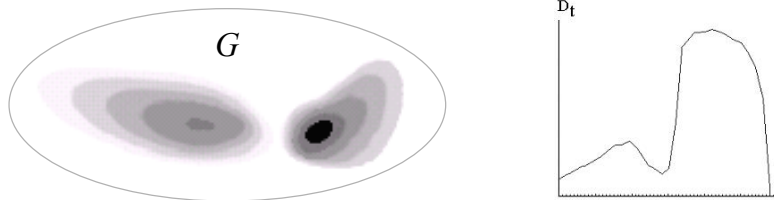
$$D(H) = \frac{F(H)}{|H|} = \frac{\min_{i \in H} \pi(i, H)}{|H|} = \frac{\min_{i \in H} \sum_{j \in H} w_{ij}}{|H|}$$



14

## How to identify core nodes

- Core nodes can be identified by analyzing the change of minimum density value  $D$  while continuously removing the weakest node
- If the weakest node is in a sparse region, then  $D$  value will increase when the node is removed, i.e., the next node has a higher density
- If the removal of the weakest node causes a significant drop in  $D$  value, then the node is highly connected with a set of stronger nodes in a dense region. It is potentially a core node since its removal greatly reduces the density of nodes connecting to it



15

## Coring procedure to cluster a proximity graph

**Input:** Proximity graph  $G$  and input parameters

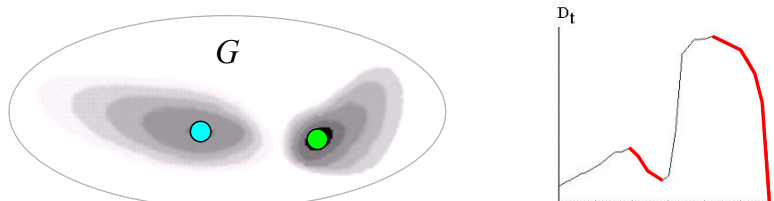
**Output:** Clustering of  $G$

Step 1: Compute the sequence of density variation

Step 2: Identify core nodes according to the input parameters

Step 3: Partition the set of core nodes into groups (cluster cores)

Step 4: Expand the cluster cores into full clusters



16

## Step 1: Compute the density variation sequence

$t \leftarrow 1$

$H \leftarrow V$

**While**  $H$  is nonempty

$F \leftarrow \min_{i \in H} \pi(i, H)$

$D_t \leftarrow F \div |H|$

$M_t \leftarrow \{i \mid i \in H \wedge \pi(i, H) = F\}$

**If**  $M_t$  consists of more than one connected component

**then**  $M_t \leftarrow$  the smallest connected component

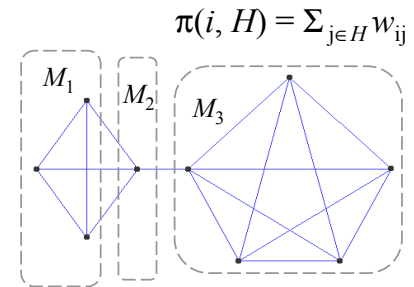
**If**  $|M_t| > 1$  **and** there's no edge connecting  $M_t$  with  $H - M_t$

**then** remove one node from  $M_t$

$H \leftarrow H - M_t$

$t \leftarrow t + 1$

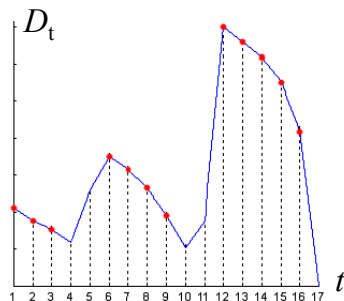
**Return** sequences of  $D_t$  and  $M_t$  with  $t = \langle 1, 2, \dots, T-1, T \rangle$



17

## Step 2: Identify core nodes

- Two control parameters:  $\alpha \in [0, 1)$  and  $\beta \in \mathbb{N}$
- Elements of  $M_t$  are core nodes if  $D_t$  satisfies two conditions:
  - Rate of decrease:  $R_t = (D_t - D_{t+1}) \div D_t > \alpha$
  - $\exists k: D_t \in \{D_{k+1}, D_{k+2}, \dots, D_{k+\beta}\}$ , where  $D_{k+1}, D_{k+2}, \dots, D_{k+\beta}$  are  $\beta$  consecutive density values that also satisfy condition (1)



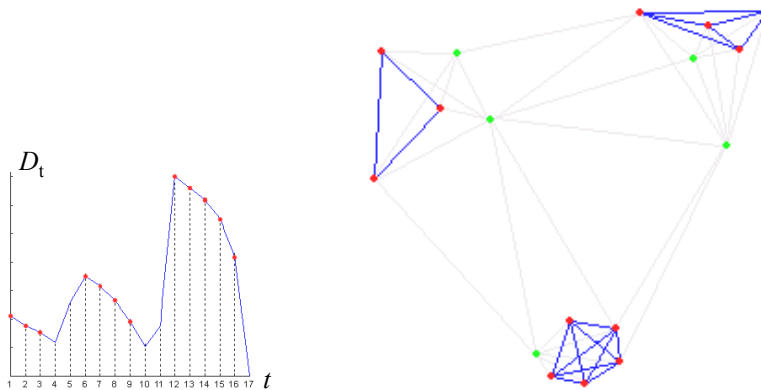
Red  $D_t$ s satisfy conditions (1) and (2) with  $\alpha = 0$  and  $\beta = 1 \mid 2 \mid 3$

- Sort the list of  $R_t$ s ( $> 0$ ), so  $D_t$ s are **ranked**. Set  $\alpha$  to the  $R_t$  value located at a relative position  $\delta$  on the sorted list. So  $\alpha$  can be replaced by  $\delta$

18

## Step 3: Partition core set into cluster cores

- Partition the core set into groups, each group represents a cluster core
- For sparse graphs, find connected components of the core graph, each component is considered as a cluster core

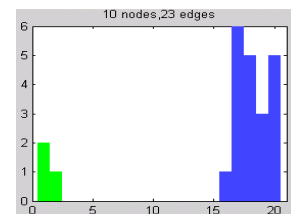
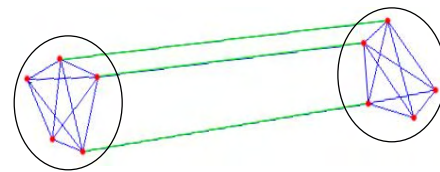


19

## Partition core set for highly connected graphs

- Core graph may be connected in one component

- Unweighted graphs: Apply again steps 1 and 2 to the core graph so as to get a smaller core set
- Weighted graphs: Visualize the histogram of edge weights and find a threshold to remove weak edges between cluster cores



- Agglomerative hierarchical clustering is another good method to partition core nodes:
  - A dendrogram is built and visualized
  - Cut the dendrogram to get cluster cores



20

## Step 4: Expand cluster cores

- Since  $M_t$  sequence built in step 1 goes from outer to inner layers:

**For**  $t = \langle T, T-1, \dots, 2, 1 \rangle$

**If**  $M_t$  contains non-core nodes

**then** assign nodes of  $M_t$  to the most similar cluster

- Assign node  $n$  to cluster  $C^* = \operatorname{argmax}_C S(n, C)$ , where  $S(n, C)$  measures the degree of similarity of node  $n$  with cluster  $C$
- $S(n, C)$  can be defined by one of these functions:
  - $\sum_{i \in C} w_{ni}$
  - $\max_{i \in C} w_{ni}$  (for weighted graphs only)
  - $\operatorname{average}_{i \in C} w_{ni}$  i.e.  $(\sum_{i \in C} w_{ni}) \div |C|$
  - $\operatorname{average}_{i \in C \wedge w_{ni} > 0} w_{ni}$  (for weighted graphs only)

21

## Cluster structure uncovering

- To reveal cluster structure, a confidence degree can be computed for non-core nodes using their similarities with the two most similar clusters
- Let  $C1$  be the most similar cluster and  $C2$  be the second most similar cluster of node  $n$ . The confidence degree of  $n$  is defined by:

$$\operatorname{confidence}(n) = \frac{S(n, C1) - S(n, C2)}{S(n, C1)}$$

- Further,  $S(n, C)$  may be computed taking into account confidence degrees of nodes of  $C$ , e.g.,  $S(n, C) = \max_{i \in C} [w_{ni} \cdot \operatorname{confidence}(i)]$

22

## Time complexity of the method

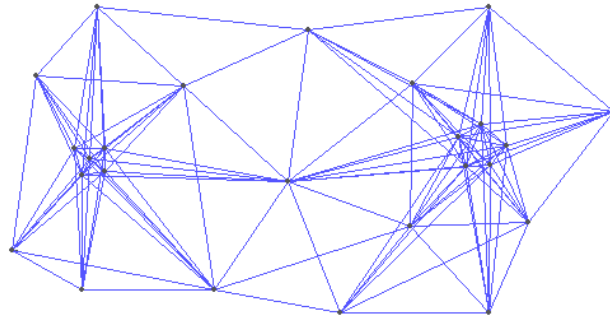
- Time complexity of implementations on graphs with adjacency-list representation:
  - Step 1:  $O(|E| + |V| \log|V|)$  for weighted graphs (similar to the complexity of finding the maximizer)
  - Step 2:  $O(|V|)$
  - Step 3:  $O(|E_c|)$  where  $E_c$  is the set of edges of the core graph
  - Step 4:  $O(|E|)$
- The total time is dominated by  $O(|E| + |V| \log|V|)$  of step 1 which is executed only once for all settings of parameters  $\delta$  and  $\beta$

23

## Example: Cluster a weighted graph

Cluster a weighted graph of  
25 nodes and 126 edges:

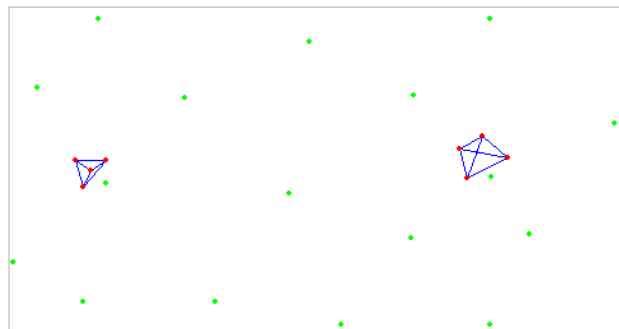
$$w_{ij} = \frac{[\max_{x,y \in V} d(x,y)] - d(i,j)}{\max_{x,y \in V} d(x,y)}$$



$$\delta = 40\%$$

$$\beta = 1$$

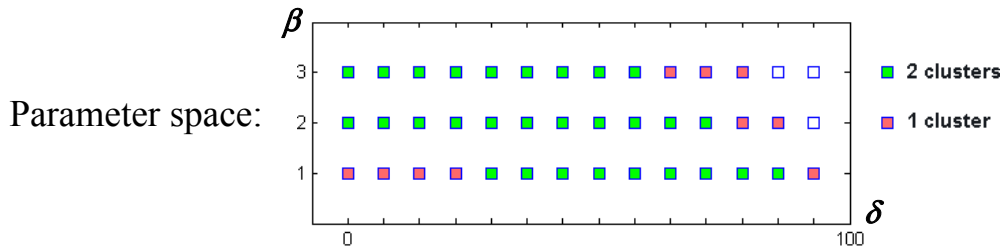
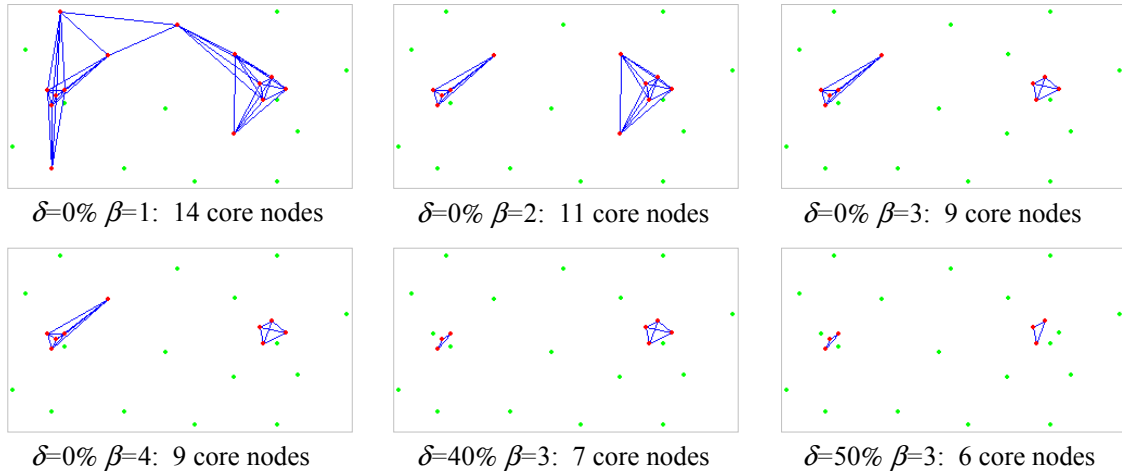
8 core nodes separated in  
2 connected components



24



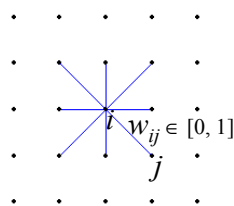
## Effects of the parameters



26

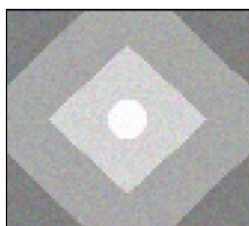
## Experiment 1: Image segmentation

- Partition a grayscale image into regions of nearby pixels that have a similar intensity
- Construct a proximity graph from an image: Nodes represent pixels. Edges weights measure the likelihood that two pixels belong to the same segment [J. Shi, J. Malik]

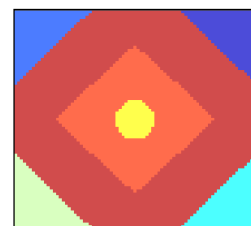
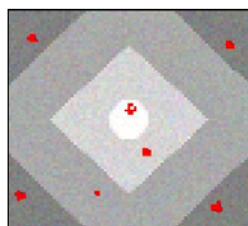


$$w_{ij} = \begin{cases} e^{-\left(\frac{I(i)-I(j)}{\sigma_I}\right)^2 - \left(\frac{d(i,j)}{\sigma_d}\right)^2} & \text{if } d(i,j) < r \\ 0 & \text{otherwise} \end{cases}$$

where  $I(x) \in [0, 1]$   $\sigma_I = 0.15$   $\sigma_d = 5$   $r = 6$



$\delta = 96\%$   
 $\beta = 2$

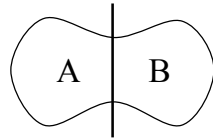


27

# Normalized cuts approach

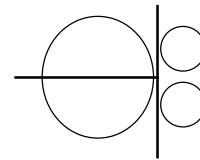
[J. Shi, J. Malik]

- One of spectral clustering approaches: Partition a graph by finding normalized cuts



$$Ncut(A, B) = \frac{cut(A, B)}{\sum_{u \in A, v \in V} W_{uv}} + \frac{cut(A, B)}{\sum_{u \in B, v \in V} W_{uv}}$$

- A NP-hard problem. Approximate a solution by the eigenvector of the second smallest eigenvalue of the normalized Laplacian matrix  $I - D^{-1}W$ , where  $D$  is the diagonal matrix of vertex degrees
- High complexity due to eigenvector computation. The normalizing factors make the criterion favor balanced clusters

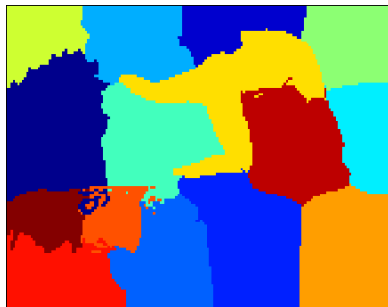


28

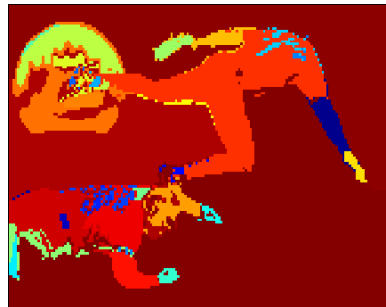
## Segmentation examples



[from J. Shi, J. Malik]



Segmentation by Normalized cuts



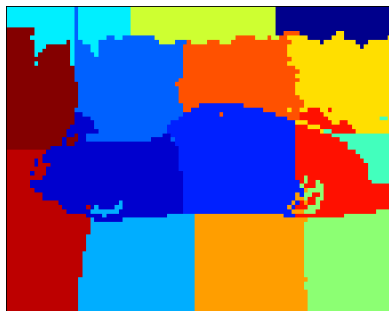
$\delta = 95.8\%$   $\beta = 2$ : 34 segments

29

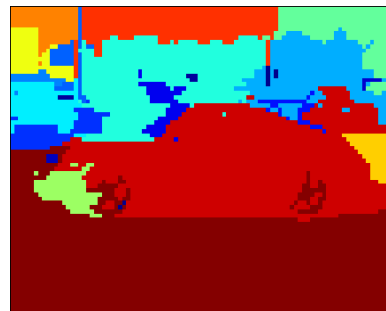
## Segmentation examples



[from A. Elgammal]



Segmentation by Normalized cuts



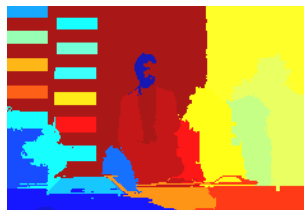
$\delta = 95.5\%$   $\beta = 2$ : 22 segments

30

## An image from Berkeley segmentation dataset



Test image #1



Our segmentation



Boundaries derived from our segmentation using 'Canny' edge detector



Ground truth boundaries by humans



Boundaries detected by Brightness Gradient method

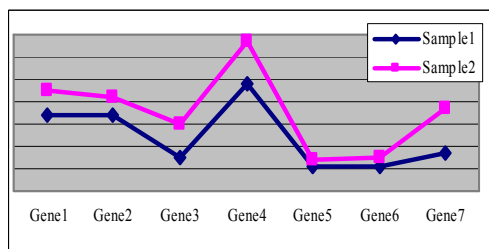


Boundaries detected by Texture Gradient method

31

## Experiment 2: Gene expression analysis

- [Http://microarray.princeton.edu/oncology/affydata/index.html](http://microarray.princeton.edu/oncology/affydata/index.html)
- The dataset contains 62 samples including 40 tumor and 22 normal colon tissues
- Each sample consists of a vector of 2000 gene expressions
- Set aside the labels and cluster the samples
- The Pearson correlation coefficient is to measure the similarity between shapes of expression patterns of every pair of samples:



$$r(X, Y) = \frac{1}{n} \sum_{i=1}^n \left( \frac{x_i - \bar{x}}{s_x} \right) \left( \frac{y_i - \bar{y}}{s_y} \right)$$

$\bar{x}, \bar{y}$ : means of  $X$  and  $Y$

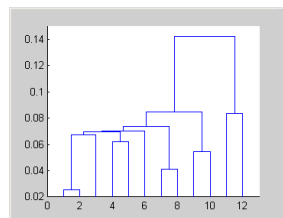
$s_x, s_y$ : standard deviations of  $X$  and  $Y$

32

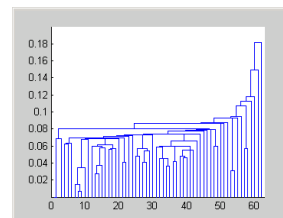
## Tissue clustering

Cluster the proximity graph, which is a complete graph of 62 nodes:

- 12 core nodes are identified with  $\delta = 50\%$  and  $\beta = 2$
- The dendrogram of the core set shows 2 well-separated groups. Cutting it at height 0.1 results in 2 cluster cores



Dendrogram of the core set



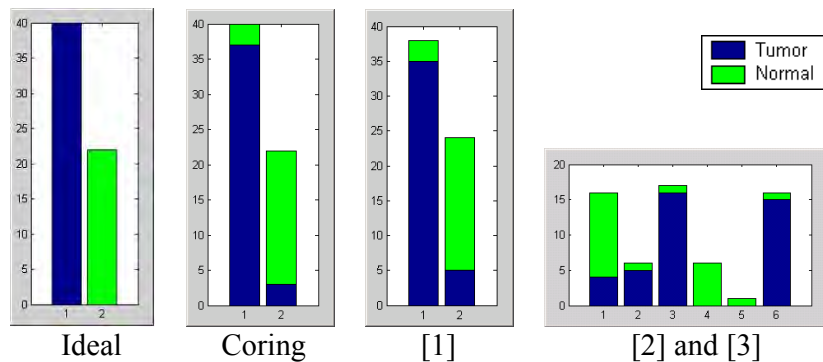
Dendrogram of the original data

- Expanding 2 cluster cores produces 2 clusters:
  - Cluster 1 has 40 samples: 37 tumor, 3 normal tissues
  - Cluster 2 has 22 samples: 3 tumor, 19 normal tissues

33

## Tissue clustering results

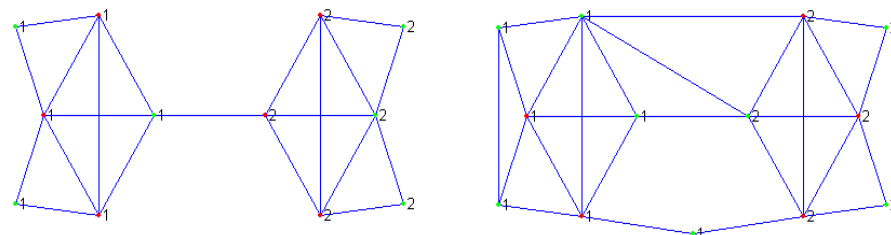
- [1] U. Alon et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci. USA*, 1999.
- [2] A. Ben-Dor et al., Clustering gene expression patterns. *Journal of Computational Biology*, 1999.
- [3] A. Bellaachia et al., A data mining algorithm for gene expression data. *Workshop on Data Mining in Bioinformatics (BIOKDD)*, 2002.



34

## Coring method advantages

- It works on both unweighted and weighted graphs
- Core nodes represent informative data objects
- Ability of manipulating core nodes by humans increases flexibility
- Cluster structures may be revealed by the ranking of core nodes and the confidence degrees of non-core nodes
- Robustness:



35

## Coring method advantages

- Advantages in comparison to the approach based on the maximizer:
  - Has a better time complexity because the time-consuming greedy procedure is run only once - for all parameter settings
  - Can work on highly connected graphs where the maximizer usually covers almost the whole graph
  - Creates no orphan nodes that cause singleton clusters
  - Provides parameters to adjust clustering results and allows human intervention and verification in the course of clustering (steps 2 and 3)

36

## Open issues

- It works best if every cluster has a strong dense core. If some cluster does not have a core or only has a faint core, it may be omitted due to not having representatives in the core set
- Range of parameter settings for good clustering is different for different graph structures. The best parameter range is learnt experimentally
- In case of complete or highly connected graphs, core nodes may be connected in one component, the core set is partitioned by visualization

37

## Future work

- Further experiment and evaluate clustering results using images, gene expression data, or protein sequences
- Perform comparative analysis on different approaches to test the reliability of the method
- Study the possibility of relaxing the dense core assumption or detecting its violations
- Test with new linkage functions such as the consistent triplets to exploit higher level relation of data objects or the linkage function for multipartite graph clustering

38

## Future work

- Analyze parameter space to infer good settings for parameters
- Partition the core set automatically for complete or strongly connected graphs
- New methods of expanding cluster cores for better boundaries between clusters
- Ways to detect cluster structures and recursively apply the method to cluster multi-scale data
- Application to directed graphs

39

## Conclusions

- Clustering methods by quasi-concave set function optimization have been studied. A new graph clustering method is proposed
- The method is simple and fast. Its two control parameters have clear interpretations
- Experiments have been done on synthetic and real datasets with encouraging results
- Further work on open issues, method improvements, experiments, clustering comparison and validation

40

## References

- [1] B. Mirkin and I. Muchnik, 2002, Layered clusters of tightness set functions, *Applied Mathematics Letters*, 15, 147-151
- [2] A. Vashist, C. Kulikowski, and I. Muchnik, 2005, Screening for ortholog clusters using multipartite graph clustering by quasi-concave set function optimization
- [3] H. Yun, 2005, Triplet-consistency graph clustering
- [4] H. Yun, A. Anghelescu and D. Fradkin, Graph clustering codes
- [5] <http://www.bioinfo.org.cn/lectures/index-13.html>
- [6] <http://www.ics.uci.edu/~eppstein/161/960201.html>
- [7] J. Shi and J. Malik, 2000, Normalized cuts and image segmentation, *IEEE Trans. Patt. Anal. Mach. Intel.*, 22, 8, 888-905
- [8] U. Alon et al., 1999, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci. USA*
- [9] A. Ben-Dor et al., 1999, Clustering gene expression patterns. *Journal of Computational Biology*
- [10] A. Bellaachia et al., 2002, E-CAST: A data mining algorithm for gene expression data. *Workshop on Data Mining in Bioinformatics (BIOKDD)*

41