

# Graph/Spectral Clustering & community evaluation

M. Vazirgiannis

# Graph Clustering

- Clustering is one of the most widely used techniques for exploratory data analysis
- applications ranging from
  - statistics, computer science, biology to social sciences or psychology
- virtually every scientific field dealing with empirical data, people attempt to get a first impression on their data by trying to identify groups of “similar behavior” in their data.

# Graph Clustering - Definitions

- Let a graph:  $G=(V, E)$ ,  $|E|$  the number of edges of the graph
- Adjacency matrix  **$AG$**   $G = (V, E)$  is an  $n \times n$  matrix
  - $AG = (a_{ij})$  where  $a_{ij} = 1$  if  $(i,j)$  is in  $E$ ; 0 otherwise.

$$D = \begin{bmatrix} \deg(v_1) & & 0 & 0 \\ 0 & \deg(v_2) & & \\ 0 & & \dots & \\ 0 & & & \deg(v_n) \end{bmatrix}$$

- $\deg(v)$  = number of edges incident to a node  $v$
- A partition of the vertices  $V$  of a graph  $G = (V, E)$  into two nonempty sets  $S$  and  $V \setminus S$  is called a cut and is denoted by  $(S, V \setminus S)$ .
- The cut size is the number of edges that connect vertices in  $S$  to vertices in  $V \setminus S$ :
- $c(S, V \setminus S) = |\{\{v, u\} \in E \mid u \in S, v \in V \setminus S\}|$ .

# *k*-Means

- *k*-Means Clustering aims to retrieve clusters  $C_1, C_2, \dots, C_k$  that minimize objective function:

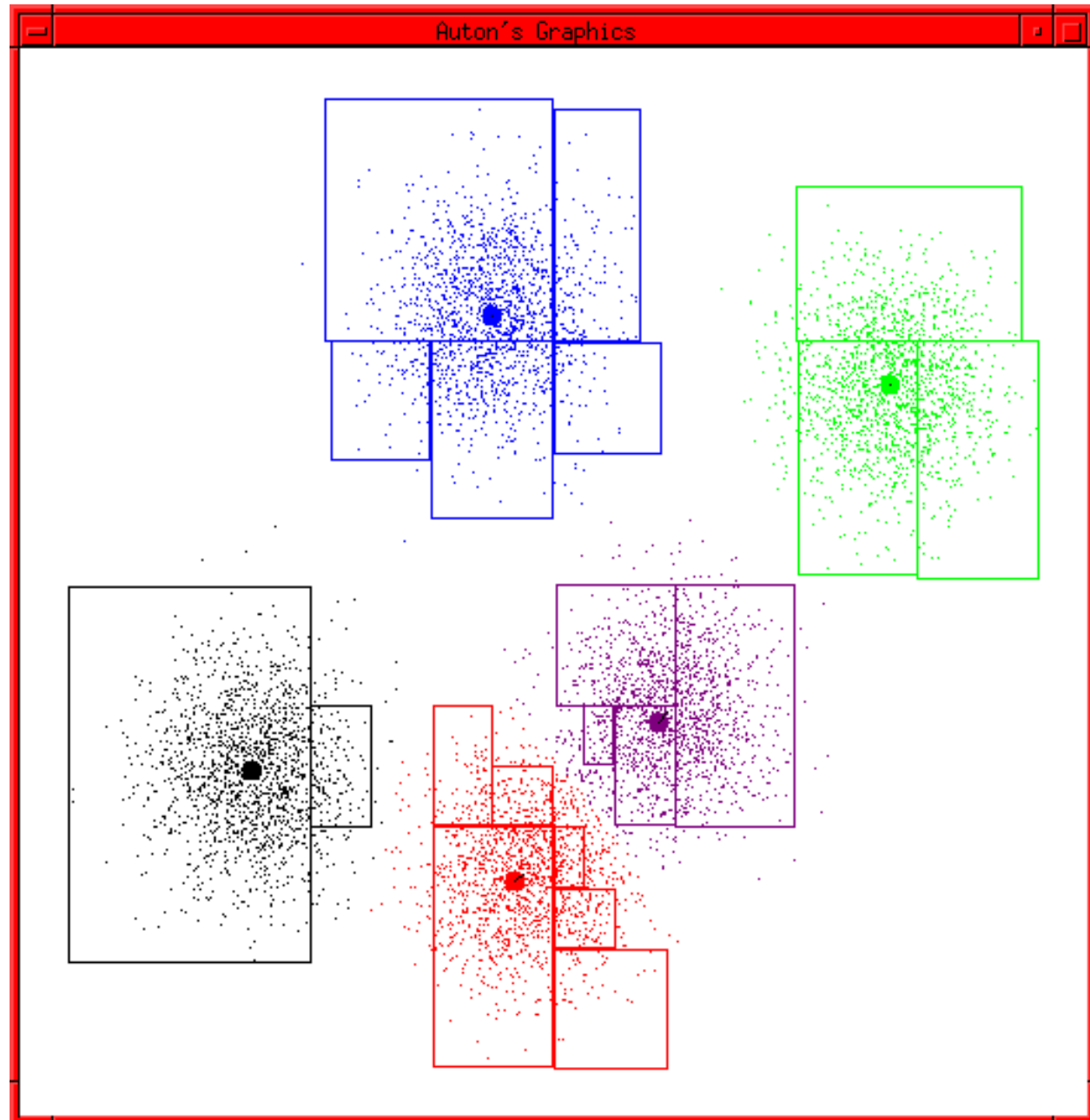
$$J_k = \sum_{j=1}^k \sum_{i \in C_k} \|x_i - m_j\|^2$$

- $m_j$  is the center of cluster  $C_k$
- Thus the clustering that minimizes the distances to cluster centers is retrieved.
- *NP*-Hard problem, even for  $k=2$ .
- Most popular heuristic: Lloyd's algorithm.

# Lloyd's heuristic algorithm

- Due to its popularity, Lloyd's heuristic is commonly referred to as  $k$ -Means algorithm.
- Algorithm:
  - Randomly select  $k$  elements from the dataset. The selected elements will form the initial cluster centroids.
  - Each element is assigned to the nearest cluster center.
  - Based on the formed clusters the new centers are computed.
  - Each element is re-assigned to the nearest cluster center.
  - The process is repeated until the process reaches a steady clustering.
- Advantages:
  - Fast and effective method for large datasets.
- Disadvantages:
  - Works well only for convex and dense clusters.

# K-means



# Spectral $k$ -Means

- $k$ -Means can be equivalently stated as a Trace maximization problem in the form:

$$\begin{aligned} \min_Y (\mathbf{Tr}(X X^T) - \mathbf{Tr}(Y^T X X^T Y)) &\equiv \\ \max_Y (\mathbf{Tr}(Y^T X X^T Y)) \end{aligned}$$

- $X$  object  $\times$  feature matrix,  $Y$  contains discrete cluster assignments.

$$Y_{ic} = \begin{cases} \frac{1}{\sqrt{|\pi_c|}} & \text{if object } i \in \pi_c \\ 0 & \text{otherwise} \end{cases}$$

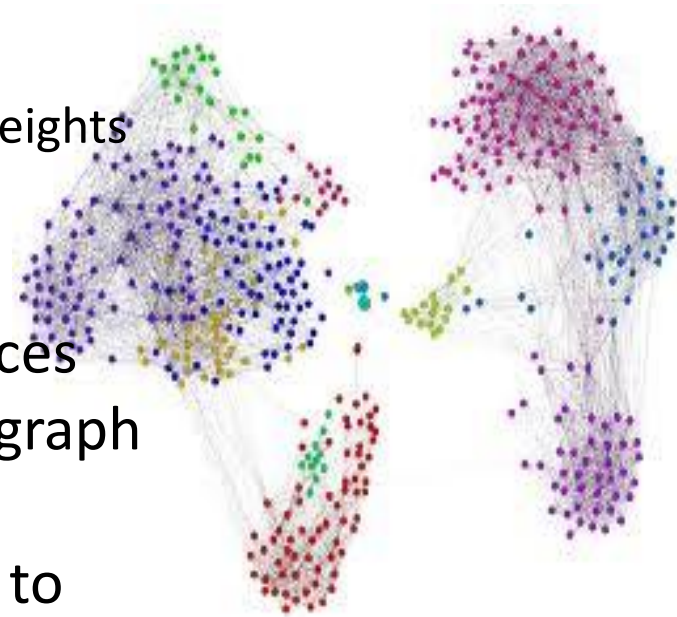
- If we relax  $Y$  to be any orthogonal matrix the solution  $Y$  will contain as columns the  $k$  dominant eigenvectors of  $XX^T$ .
- Need of extra step for discretizing solution.
- Trace of a square matrix  $n \times n$   $Tr(A) = \sum a_{ii}$
- $Tr(A)$  = sum of the [eigenvalues](#), and it is an [invariant](#) with respect to a [change of basis](#)

# Key elements of “Spectralization”

- Formulation of objective function as trace maximization/minimization problem.
- In case of discrete problems (such as clustering) need for constraint relaxation.
  - Also referred to as continuous relaxation or spectral relaxation.
- Extra step needed for discretizing continuous solutions.
  - Lloyd’s heuristic is commonly used for this process.

# Graph partitioning

- $k$ -Means works well for vector-data representations and balanced clusters.
- In many applications we are presented with:
  - pairwise similarities  $\Leftrightarrow$  a weighted graph.
  - similarity matrix  $W \Leftrightarrow$  a graph with edge weights  $e(i,j)=W(i,j)$ .



- It is generally agreed: “a subset of vertices forms a good cluster if the induced subgraph is dense, but there are relatively few connections from the included vertices to vertices in the rest of the graph”.
- Ref pg 32  
<http://dollar.biz.uiowa.edu/~street/graphClustering.pdf> )

# Graph partitioning

## *Cluster fitness measures - Density based*

- *Instance: An undirected graph  $G = (V, E)$ , a density measure*
- *$\delta(\cdot)$  defined over vertex subsets  $S$  in  $V$ , a positive integer  $k \leq n$ , and a rational number  $\xi$  in  $[0, 1]$ .*
- *Is there a subset  $S$  in  $V$  such that  $|S| = k$  and the density  $\delta(S) > \xi$ ?*
- *Asahiro et al. study the general question:*
- *Given a graph  $G = (V, E)$ , is there a  $k$ -subgraph in  $G$  with at least  $f(k)$  edges?*
- *NP – complete!*

# Graph partitioning

## *Cluster fitness measures – Connectivity based*

- The aim is to cluster the datasets/partition the graph.
- In this setup how can we “intuitively” formulate the mathematical problem of clustering?
- Lets try to derive the clustering that minimizes the sum of weights between objects in different clusters.

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- For  $k$ -clusters.

$$\text{cut}(A_1, \dots, A_k) := \sum_{i=1}^k \text{cut}(A_i, \bar{A}_i)$$

- Minimizing cut directly does not balance for cluster size.
  - Will tend to create a cluster with one element and the other with the rest.

# Graph partitioning

## *Cluster fitness measures – Connectivity based*

- Two popular objective functions that balance for cluster sizes:

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

$$\text{Ncut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}.$$

- Deriving the clustering that minimizes these objective functions is *NP-Hard*.
- ...but they can be stated as Trace minimization problems.
- Can be approximated by spectral methods.
- Notation:
  - $W$ = the object similarity matrix.
  - $D$ = diagonal matrix with diagonal values  $d_{i,i} = \sum_{j=1}^n w_{ij}$

# Ratio Cut /Spectral Clustering

- In graph-partitioning a popular clustering objective is Ration Cut.

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

- Equivalent Trace minimization problem

$$\min_Y \text{Tr}(Y^T (D - W)Y)$$

- If we relax  $Y$  to be any orthogonal matrix  $Y$ , solution will contain as columns the  $k$ -eigenvectors that correspond to the smallest eigenvalues of Unnormalized Graph Laplacian.

$$L = D - W.$$

- In the case of 2-way clustering the solution is derived by the eigenvector that corresponds to the *second smallest eigenvalue*.

# Normalized Cut /Spectral Clustering

- In graph-partitioning a popular clustering objective is Normalized Cut (*NCut*)

$$NCut(A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \overline{A_i})}{vol(A_i)}$$

- Equivalent Trace optimization problem

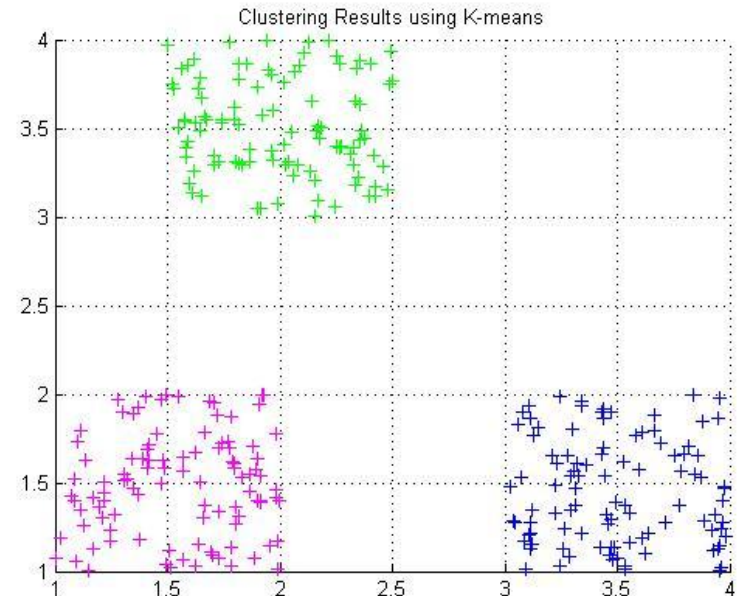
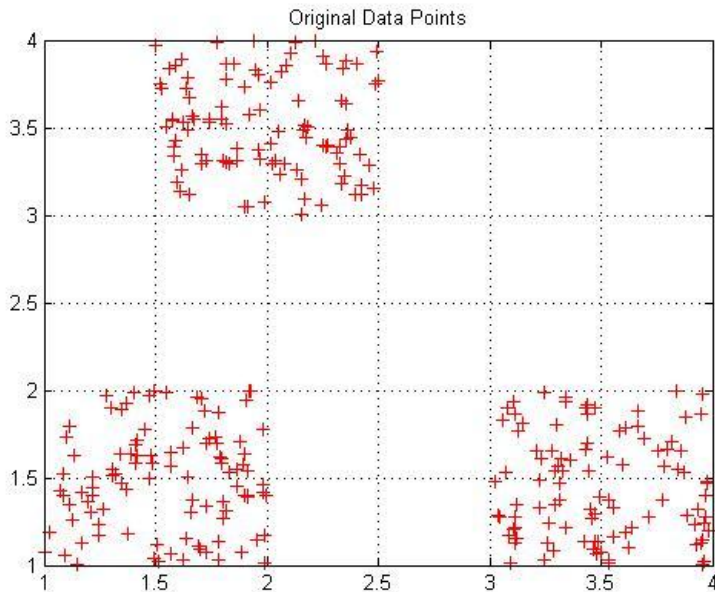
$$\min_Y \mathbf{Tr}(Y^T (I - D^{-1/2} W D^{-1/2}) Y)$$

- If we relax  $Y$  to be any orthogonal matrix  $Y$ , solution will contain as columns the  $k$ -eigenvectors that correspond to the smallest eigenvalues of Normalized Graph Laplacian.

$$L = I - D^{-1/2} W D^{-1/2}$$

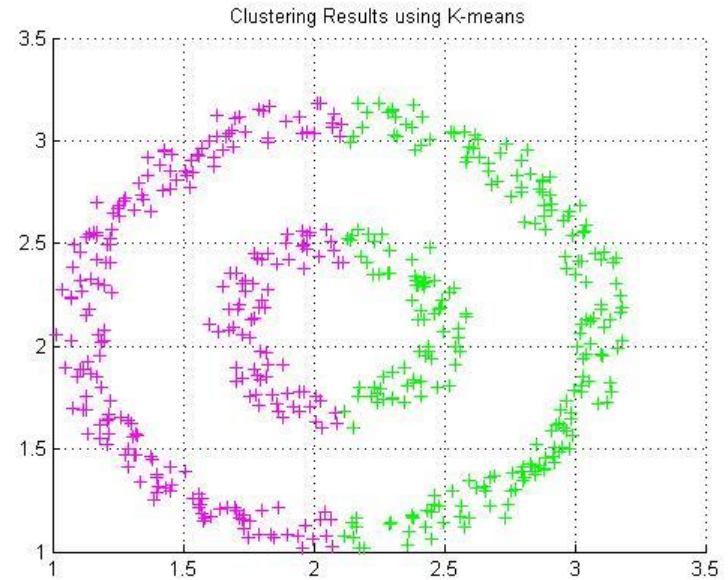
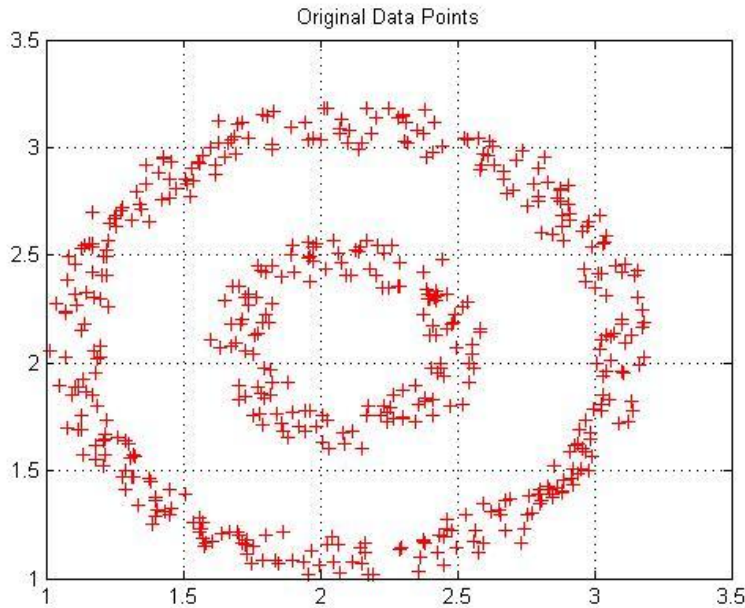
- In the case of 2-way clustering the solution is derived by the eigenvector that corresponds to the *second smallest eigenvalue*.

# K-means Example 1



- K-means is good at finding dense areas that are defined by a convex region

# K-means Example 2



- On non-convex clusters k-means will give bad results due to the tendency to find equal-sized clusters.

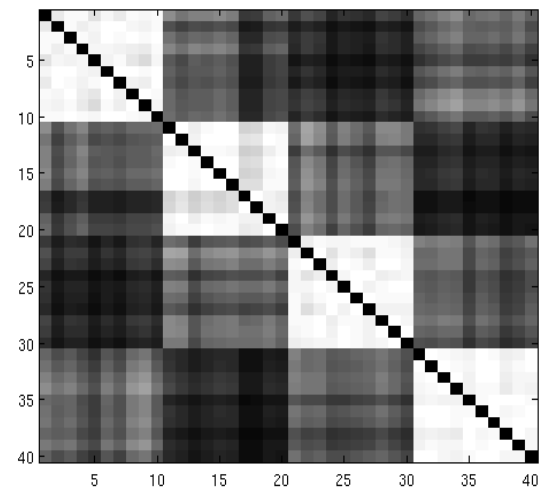
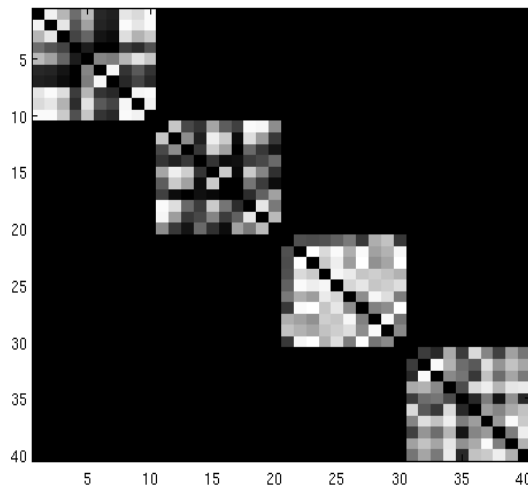
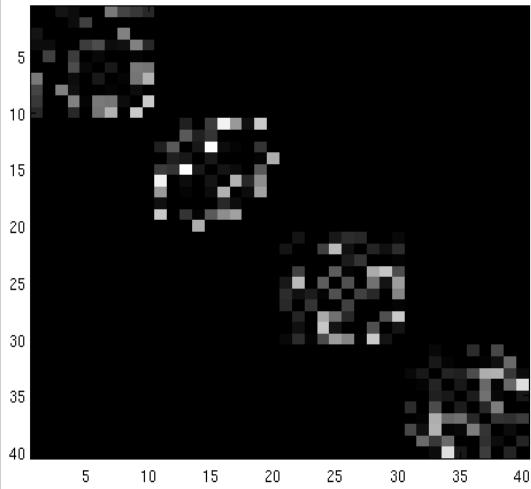
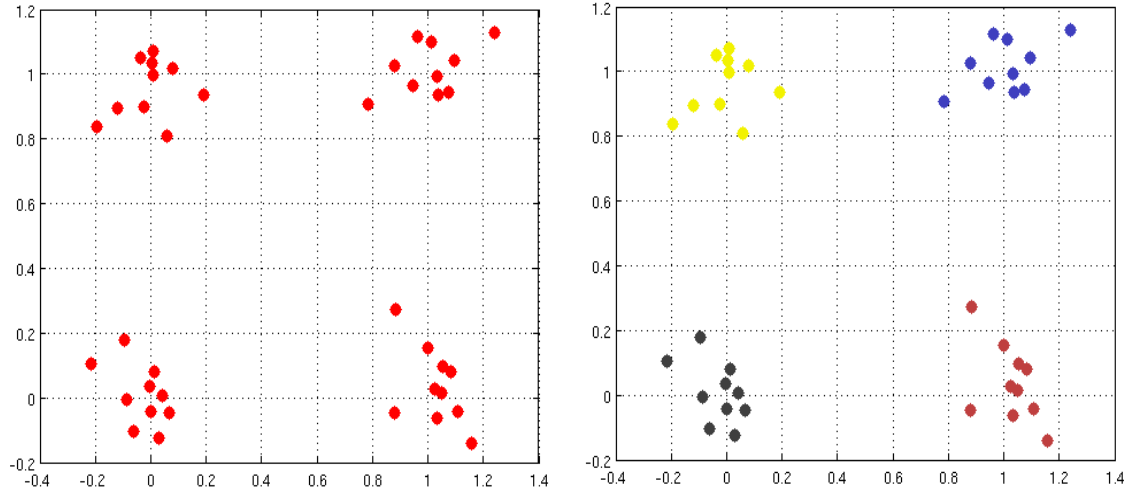
# Spectral Clustering

- A family of algorithms that use eigenvectors of matrices that derive from the data.
- A similarity matrix is needed
- They are called spectral because they use the spectrum of the similarity matrix to reduce the dimension.
- The selection and use of the eigenvectors differs from one algorithm to another.
- The number of clusters is defined by the user ( $k$ )
- Application in areas like :
  - Image segmentation
  - Community detection in graphs

# Affinity Matrix

- Affinity Matrix  $A \in R^{n \times n}$
- Define  $A_{ij} = e^{-\|s_i - s_j\|^2 / 2\sigma^2}$  for  $i \neq j$  else  $A_{ii} = 0$
- The scaling factor ( $\sigma^2$ ) is chosen by the user
- “Closer” points will have higher weight
- The weight is a function of ( $\sigma$ )
- Realistically, we search over the values of  $\sigma^2$  and chose the one that returns the “tightest” clusters

# Affinity Matrix



- [http://www.cs.utah.edu/~jfishbau/advimproc/project6/images/pts\\_dist\\_example.png](http://www.cs.utah.edu/~jfishbau/advimproc/project6/images/pts_dist_example.png)

# Laplacian Matrix

- When the algorithm is applied :
    - on data points we use Matrices like affinity Matrix(A).
    - on Graphs we use the Adjacency Matrix
  - From the affinity Matrix we build the Laplacian Matrix
    - Unnormalized :  $L=D-A$
    - Normalized (symmetric) :  $L = D^{-1/2} A D^{-1/2}$
    - Normalized (random walk) :  $L = D^{-1} A$
- Where D : diagonal matrix where  $D(i,i)=\text{sum}(\text{row}(A,i))$

# Laplacian Matrix

Properties of  $L = D - W$  (unnormalized)

The matrix  $L$  satisfies the following properties:

1. For every vector  $f$  in  $\mathbb{R}^n$  :

$$f' L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$$

2.  $L$  is symmetric and positive semi-definite.

3. The smallest eigenvalue of  $L$  is 0, the corresponding eigenvector is the constant one vector  $\mathbf{1}$ .

4.  $L$  has  $n$  non-negative, real-valued eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$$

*Number of connected components and the spectrum of  $L$*

- multiplicity  $k$  of the eigenvalue 0 of  $L$  equals the number of connected components  $A_1, \dots, A_k$  in the graph.

# Spectral Clustering Algorithms

- **Unnormalized ( $k > 2$ )**

*Input:* Affinity matrix  $S$  ( $n \times n$ ), number of clusters  $k$

- Compute Laplacian  $L$ .
- Compute the first  $k$  eigenvectors  $u_1, \dots, u_k$  of  $L$ .
- Let  $U$  ( $n \times k$ ) the matrix containing the vectors  $u_1, \dots, u_k$  as columns.
- for  $i = 1, \dots, n$ ,
  - let  $y_i$  in  $R_k$  the vector corresponding to the  $i$ -th row of  $U$ .
  - Cluster the points  $(y_i)_{i=1, \dots, n}$  in  $R_k$  with the  $k$ -means algorithm into clusters  $C_1, \dots, C_k$ .

*Output:* Clusters  $A_1, \dots, A_k$  with  $A_i = \{j \mid y_j \text{ belongs to } C_i\}$ .

# Spectral Clustering Algorithms

- Each row in the Eigenvector Matrix (U) corresponds to a point
- The same algorithm as can be used with the normalized Laplacians
- If  $k=2$  we can use only the second eigenvector. In that case we split the clusters by a threshold on the values of the eigenvector.
- In some version of spectral clustering (Shi & Malik) only the 2<sup>nd</sup> eigenvector is used and multiple thresholds are applied

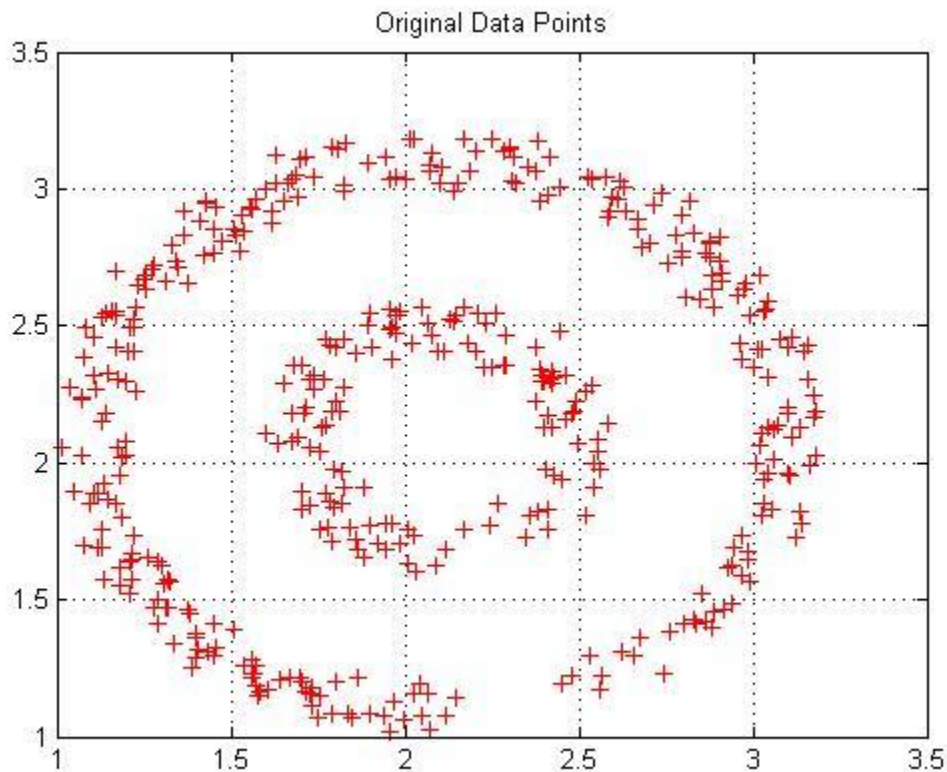
# Why not K-Means?

- The last step in the example algorithm is the application of K-means, why not just apply K-means to the original data?
- The main Reason is the inability of k-means to detect non-convex regions.
- The spectral/eigenvector domain is a lower-dimensional space where the points are easily separable.

# Example 1

## “Good” Scaling Factor (0.1)

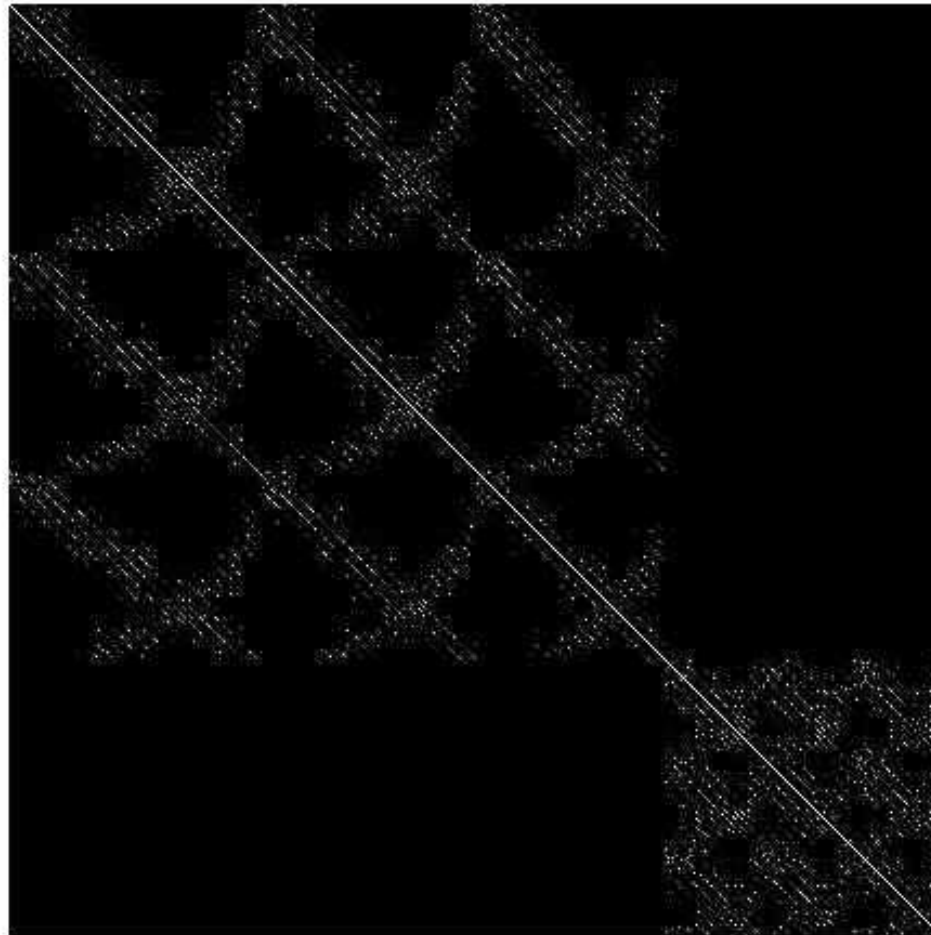
- DATA



# Example 1

## “Good” Scaling Factor (0.1)

Affinity Matrix

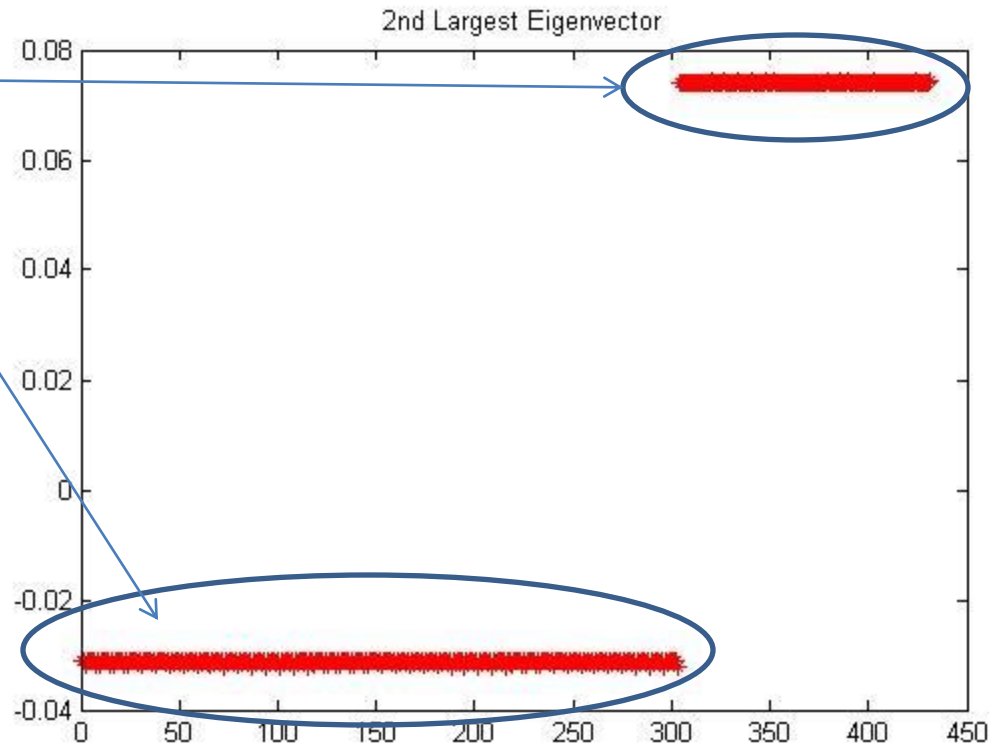


# Example 1

## “Good” Scaling Factor (0.1)

- 2<sup>nd</sup> Largest EigenVector

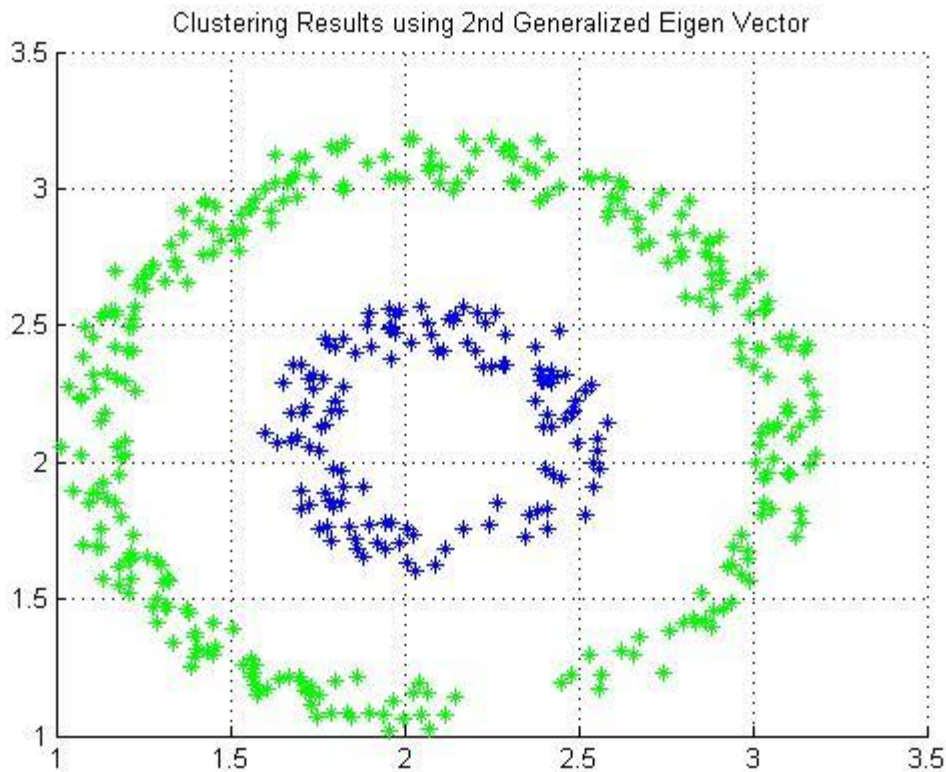
The values of the eigenvector are easily separable into two clusters



# Example 1

## “Good” Scaling Factor (0.1)

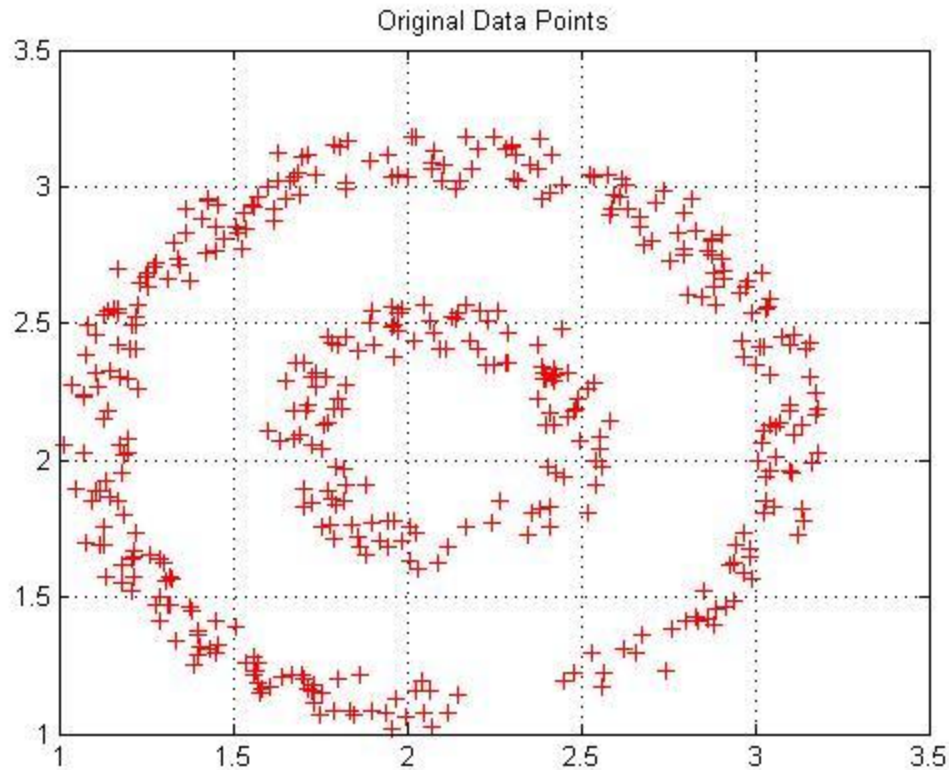
- Clusters



# Example 2

## “Bad” Scaling Factor (1.0)

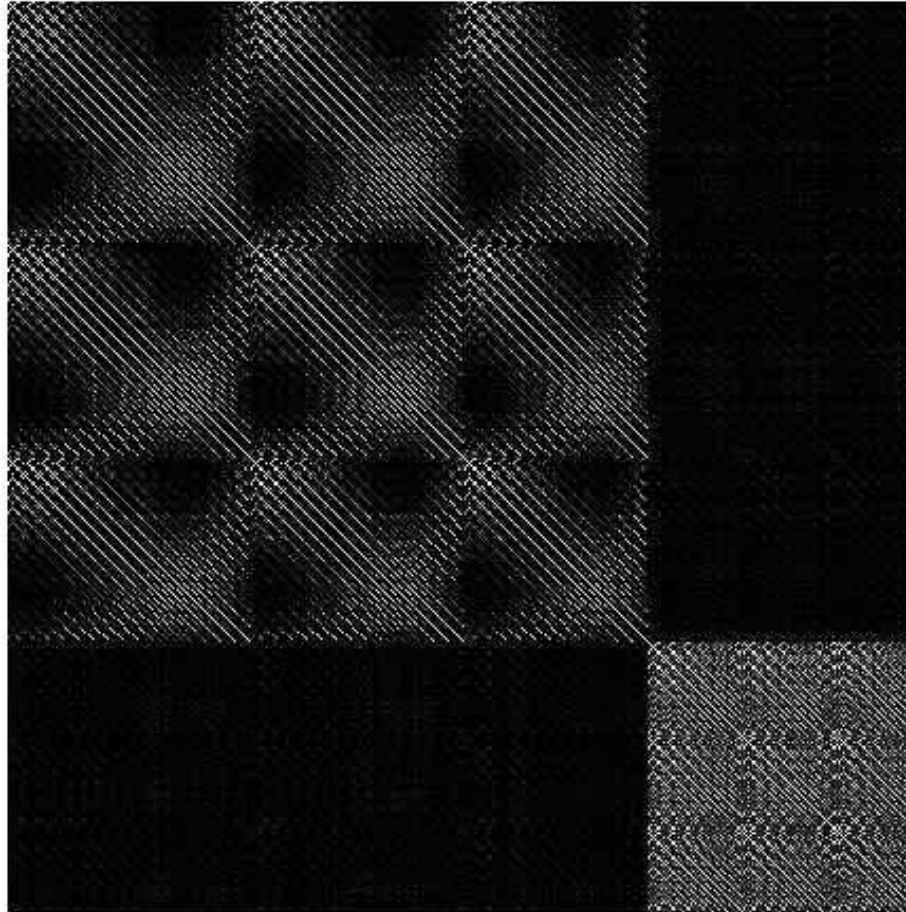
- DATA



# Example 2

## “Bad” Scaling Factor (1.0)

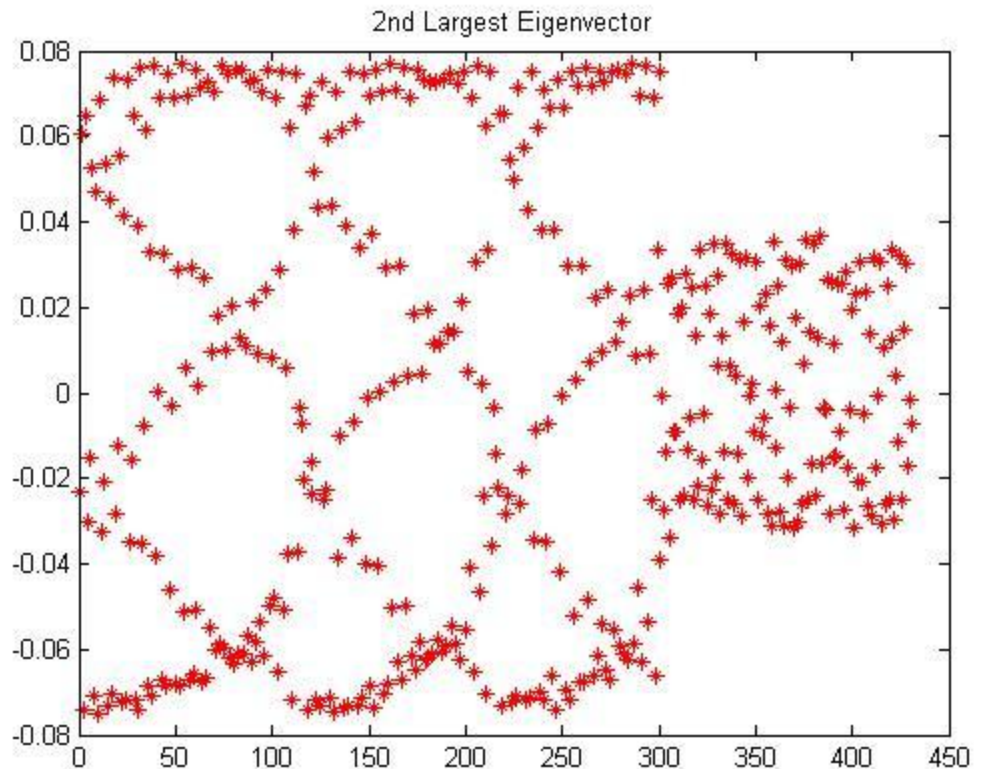
Affinity Matrix



# Example 2

## “Bad” Scaling Factor (1.0)

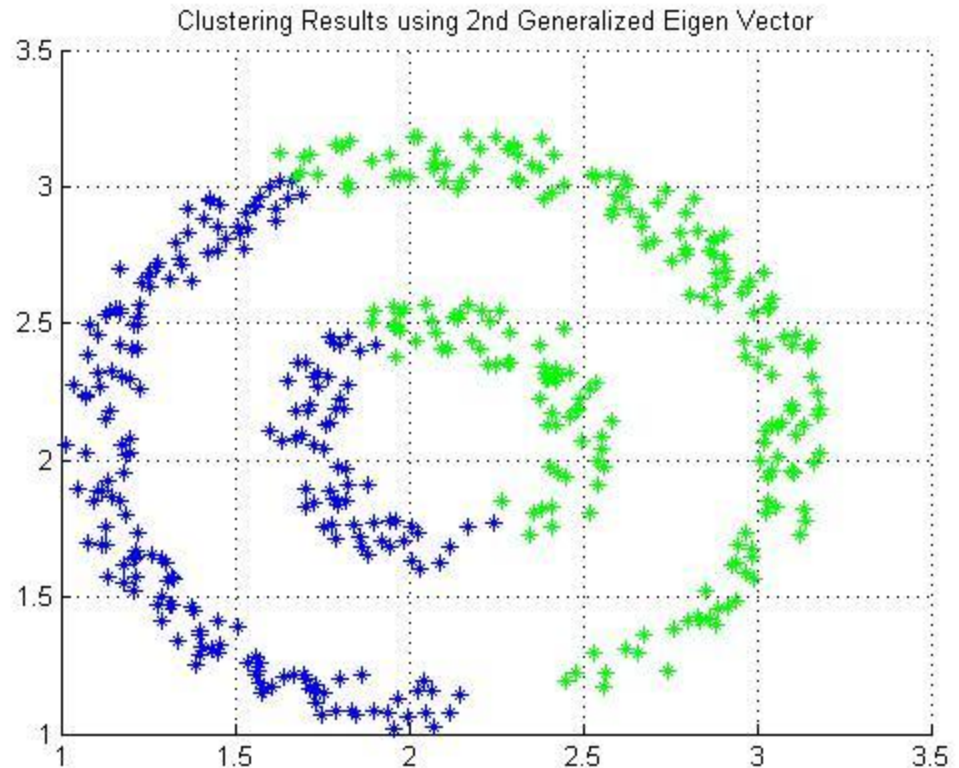
- 2<sup>nd</sup> Largest EigenVector
- It is harder to separated the clusters



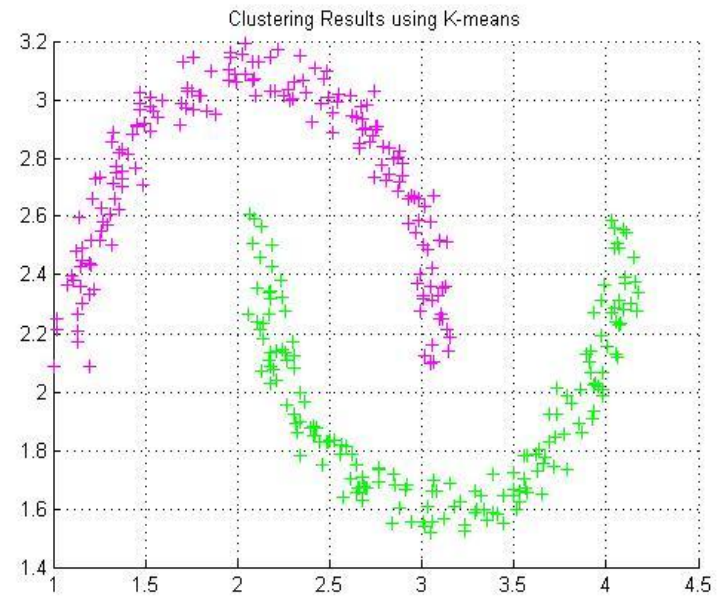
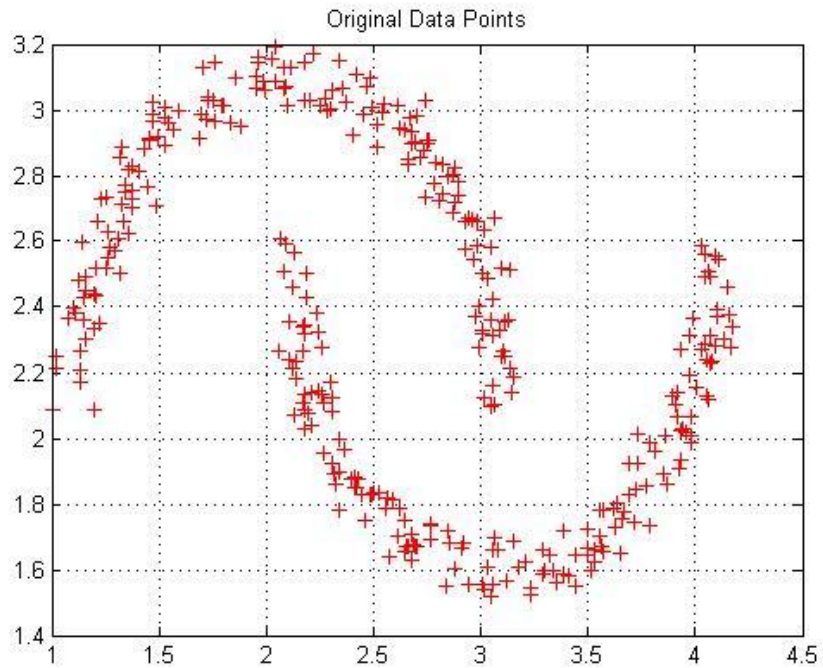
# Example 2

## “Bad” Scaling Factor (1.0)

- Clusters:  
The result is similar to what k-means would give



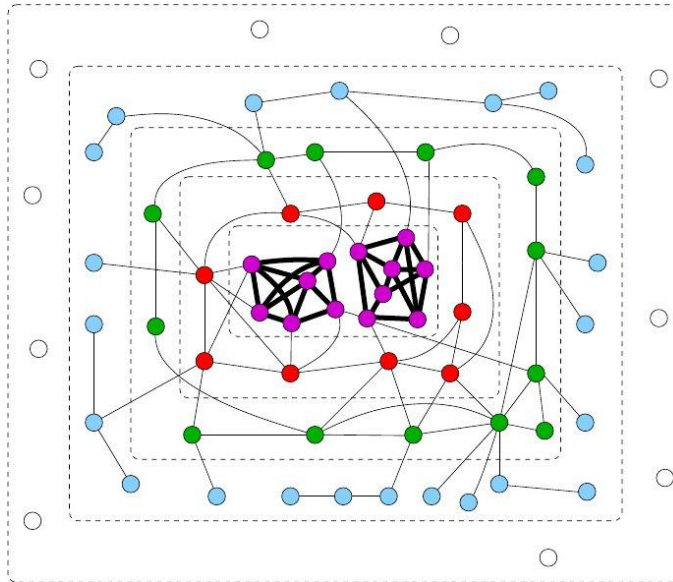
# Example 3



# References

- A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. of NIPS-14, 2001*
- J. Shi and J. Malik. Normalized cuts and image segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.
- S. X. Yu and J. Shi. “Multiclass spectral clustering”, in *International Conference on Computer Vision, 2003*.
- Inderjit S. Dhillon, Yuqiang Guan, Brian Kulis , “Kernel k-means, Spectral Clustering and Normalized Cuts”, in proceedings of the ACM KDD 2004
- “Graph clustering”, Satu Elisa Schaeffer, COMPUTER SCIENCE REVIEW1(2007)27–64
- A Tutorial on Spectral Clustering, Ulrike von Luxburg, March 2007.

# Evaluation of communities with degeneracy



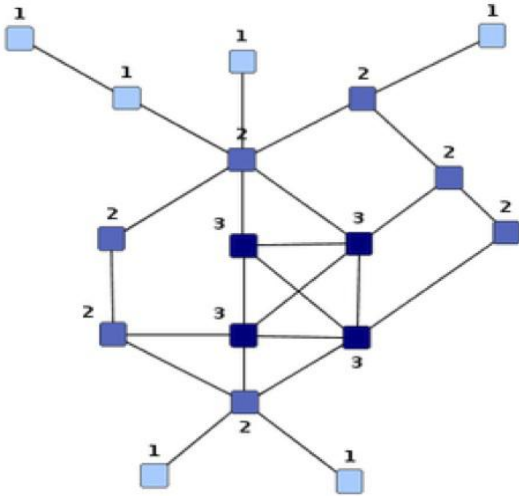
# Graphs are everywhere

- The WWW is a directed graph
- Social Networks & citation graphs constitute inherently Graphs
- Such graphs can be directed (WWW) and or signed (trust networks)
- High dynamics: constantly changing in both “shape” and size”

# Community evaluation

- *Community detection* and *evaluation* in graphs is a cornerstone issue .
- Different metrics/ measurements /methods are used
  - Hub/authorities
  - Modularity
  - Density/Diameter/Link distribution etc....
  - Centrality /Betweenness
  - Clustering coefficient
  - Structural cohesion
- A thorough state of the art review is offered by Fortunato

# K-core



$$G_0 = G$$

$$G_0 : 1\text{-core of } G$$

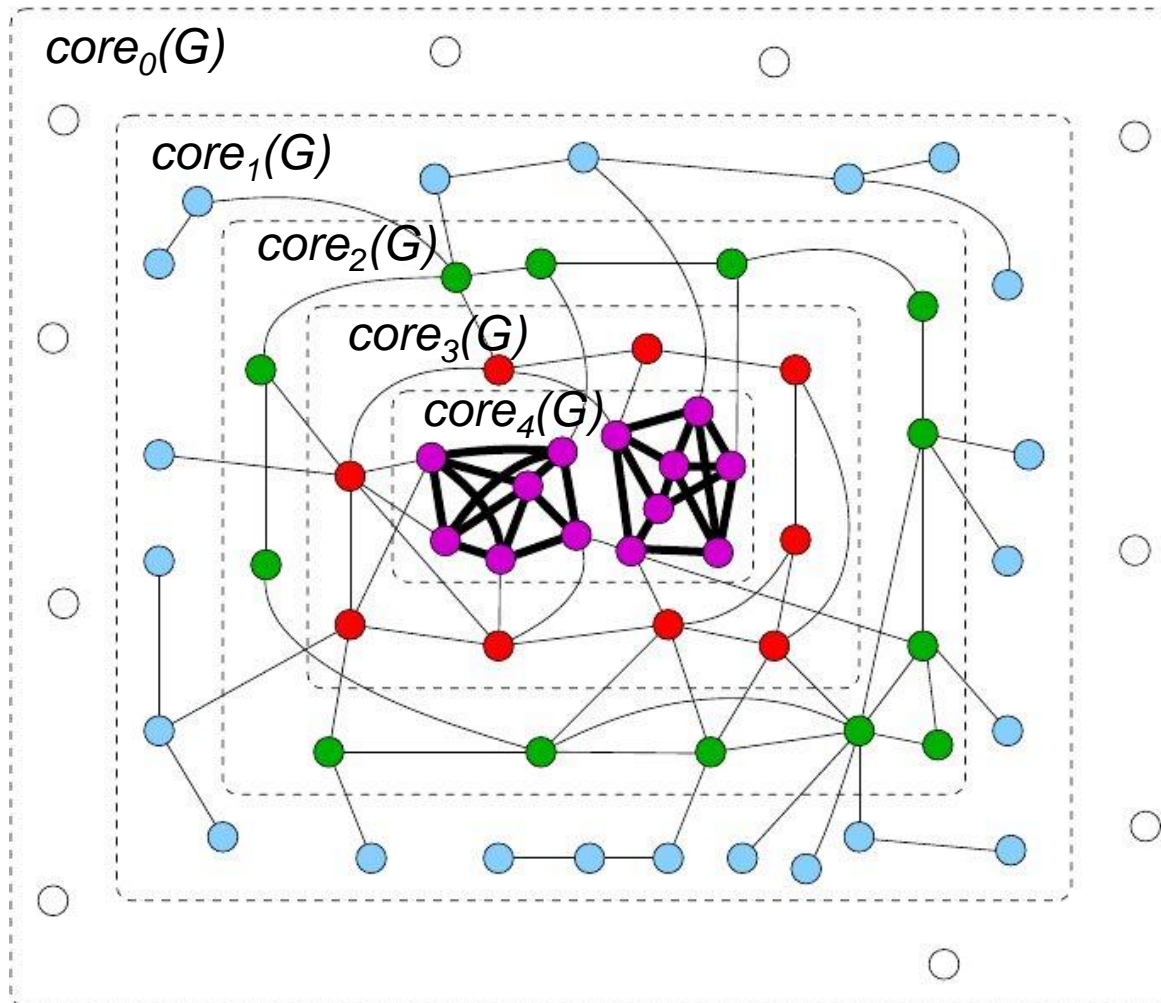
$$G_1 : 2\text{-core of } G$$

$$G_2 : 3\text{-core of } G$$

$$G_0 \supseteq G_1 \supseteq G_2 \supseteq G_3$$

- The degeneracy and the size of the maximum rank core provide a good indication of the cohesiveness of the graph  $G$ .
- *Time complexity:  $O(n.k)$  ( $n = |G|$ )*
- **Fast!** especially in real word data where  $G$  is usually sparse.

# Another example



# DBLP dataset

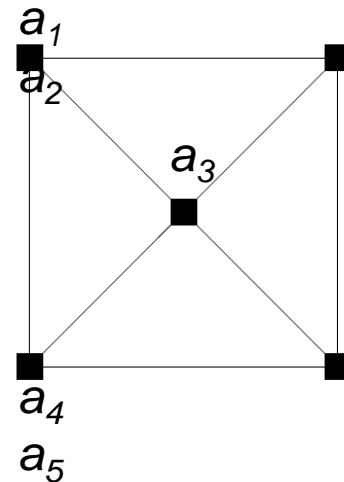
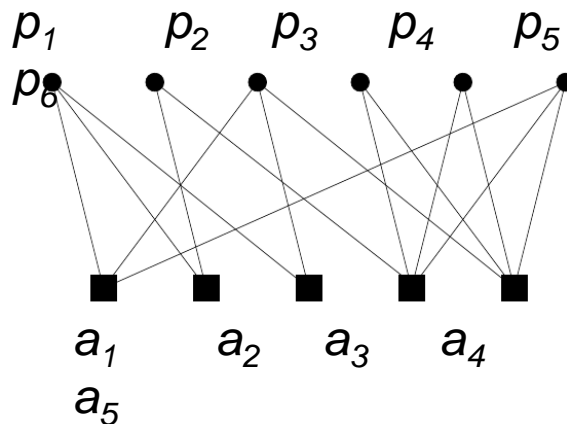
## DBLP: bipartite graph (authors + papers):

- Co-authorship graph (undirected)
- Citations graph (directed)

Co-authorship graph

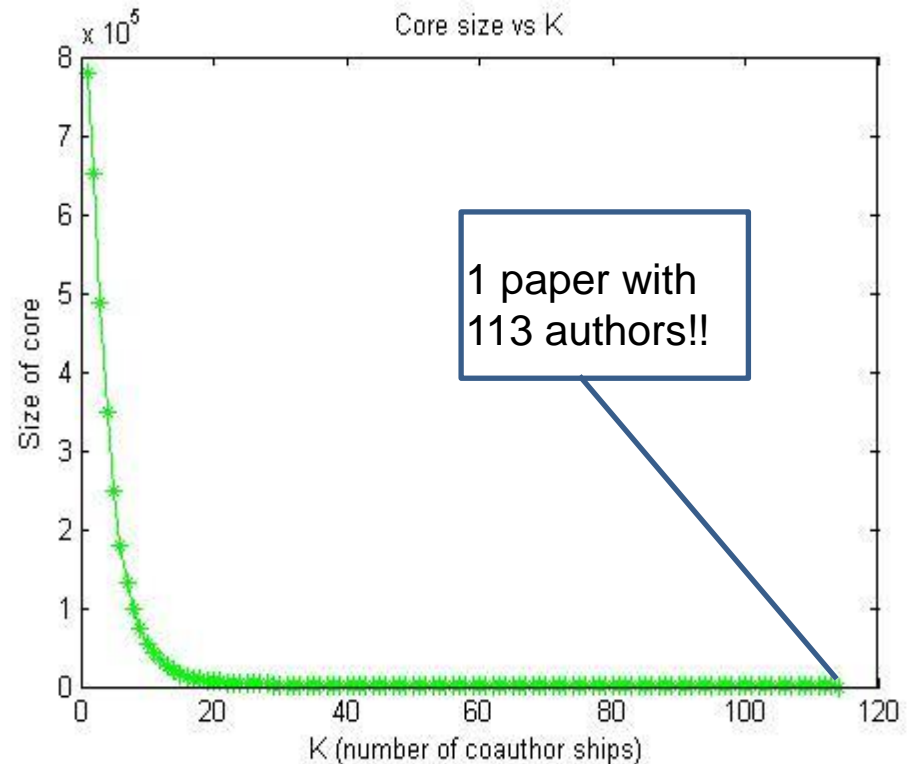
- *Vertices*: Authors

- *Edges*: two authors are joined by an edge if they have some common paper



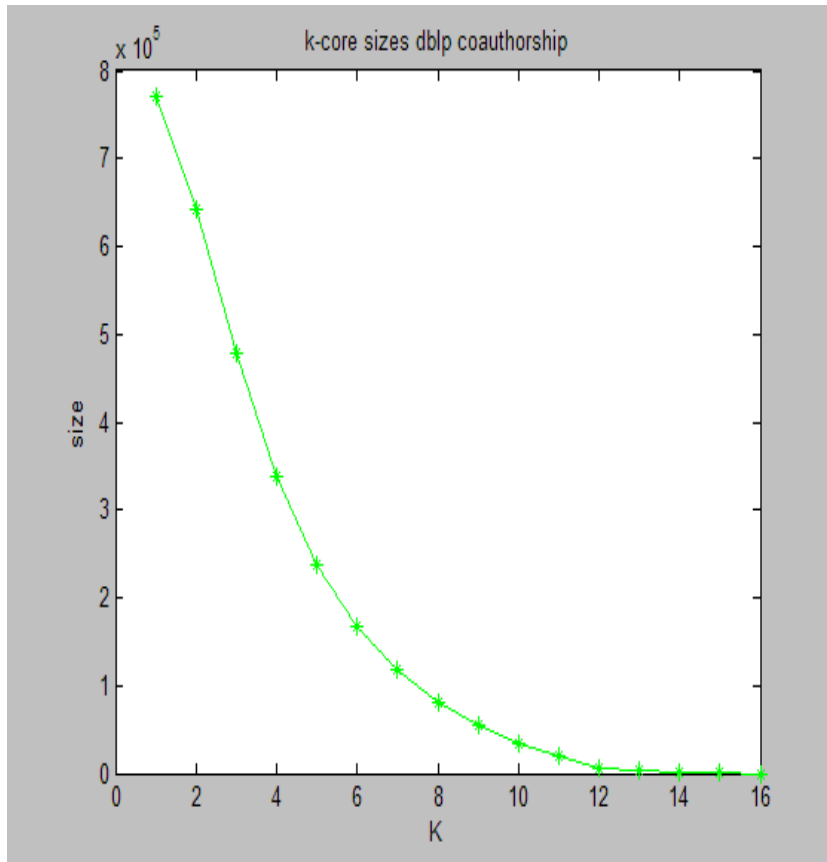
# DBLP – #coauthors distribution

- 825 K authors
- Filtered out 1% of the papers
- max 15 authors/paper



# DBLP co-authorship - filtered graph

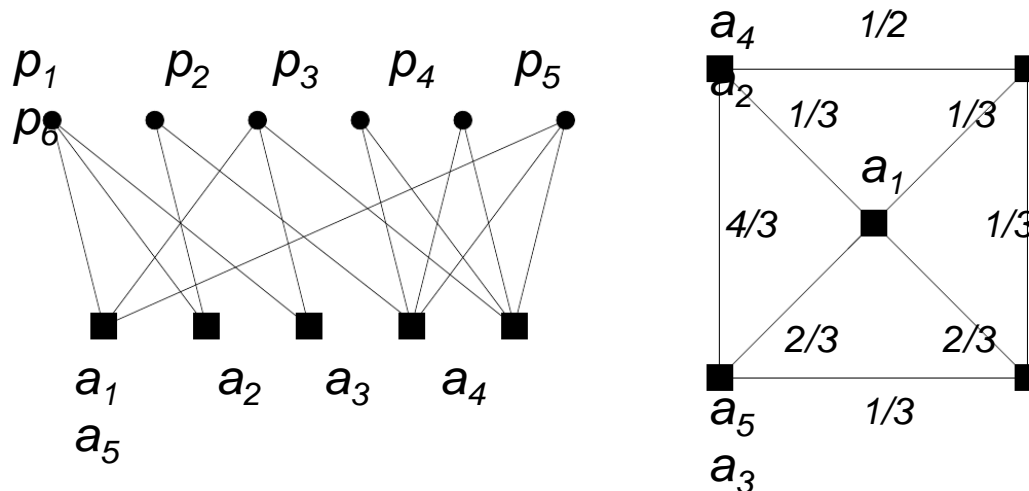
The top rank core is 15 and contains 76 authors



Kurt Mehlhorn	Joseph S. B. Mitchell	Marc J. van Kreveld
Micha Sharir	David Eppstein	Martin L. Demaine
Pankaj K. Agarwal	Erik D. Demaine	Ferran Hurtado
Mark de Berg	Olivier Devillers	Timothy M. Chan
Rolf Klein	Sándor P. Fekete	Oswin Aichholzer
Mark H. Overmars	Henk Meijer	Bettina Speckmann
Herbert Edelsbrunner	Sariel Har-Peled	Jeff Erickson
Stefanie Wuhler	John Hershberger	Therese C. Biedl
Jack Snoeyink	Alon Efrat	Greg Aloupis
Joseph O'Rourke	Stefan Langerman	David Bremner
Subhash Suri	Bernard Chazelle	Anna Lubiw
Otfried Cheong	Joachim	Esther M. Arkin
Hazel Everett	Gudmundsson	Boris Aronov
Sylvain Lazard	Giuseppe Liotta	Vida Dujmovic
Helmut Alt	Sue Whitesides	Suneeta Ramaswami
Emo Welzl	Christian Knauer	Thomas C. Shermer
Günter Rote	Raimund Seidel	David R. Wood
Leonidas J. Guibas	Michiel H. M. Smid	Perouz Taslakian
Chee-Keng Yap	Tetsuo Asano	John Iacono
Danny Krizanc	David Rappaport	Sergio Cabello
Pat Morin	Vera Sacristan	Sébastien Collette
Jorge Urrutia	Hee-Kap Ahn	Belén Palop
Diane L. Souvaine	Prosenjit Bose	Mirela Damian
Ileana Streinu	Michael A. Soss	Jirí Matousek
Dan Halperin	Godfried T.	Otfried Schwarzkopf
Hervé Brönnimann	Toussaint	Richard Pollack

# DBLP co-authorship – Weighted graph

- Collaboration effort: authors participating in papers with many coauthors get biased credit
- i.e. in the unfiltered case:
  - 1 paper with 113 authors creates the most dense co-authorship collaboration structure
  - for most of the authors was the only paper
- Each author of a paper should get a just credit (i.e.  $1/\#$  authors)



# Fractional cores

The following notions are defined for the fractional setting:

- weighted co-authorship graph
- degeneracy  $\delta^*(H_{\text{DBLP}}, w)$
- $k$ -core, for  $k$  in  $\mathbf{Q}$
- densest core author **rank**
- Trim procedure **exactly same complexity!**

# Fractional cores

## Co-authorship edge weight

- For every edge  $e = \{x, x'\}$
- *weighted co-authorship affinity among  $x$  and  $x'$ : collaboration !*

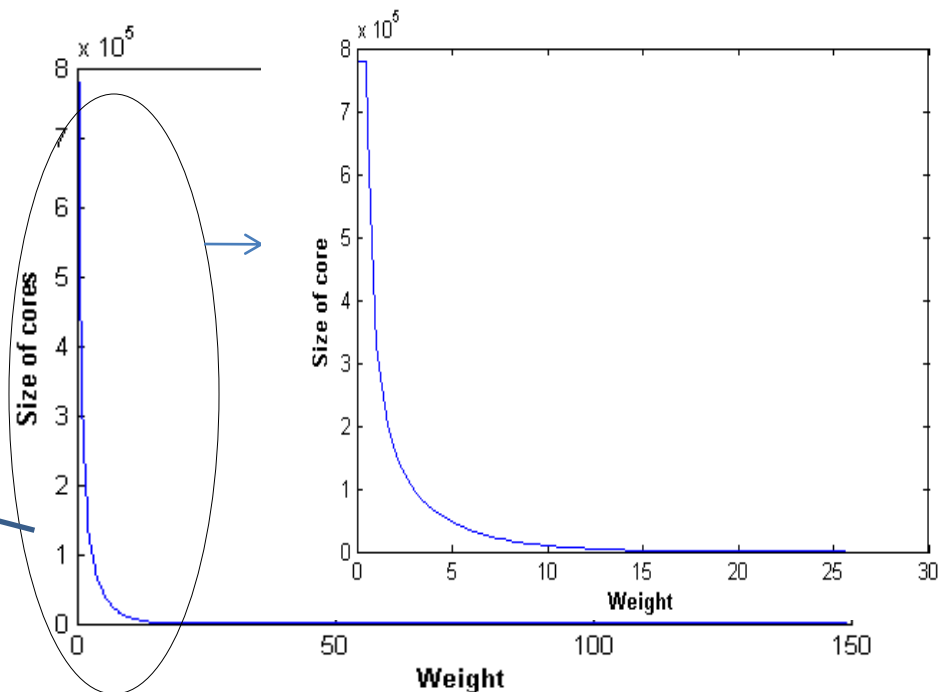
$$w(e) = \sum_{y \in N(x) \cap N(x')} \frac{1}{|N(y)|}, |N(y)| \# \text{ coauthors/ paper}$$

## Vertex fractional degree, $x$ in $(G, w)$

$$\text{deg}_{G,w}(x) = \sum_{e \in E(x)} w(e)$$

represents the total collaboration effort of an author for coauthoring papers

Distribution of the fractional k-core sizes in the DBLP coauthorship graph



# Fractional cores – selected authors

<b>Author</b>	<b>Rank</b>	<b>Fractional Rank</b>
C.H. Papadimitriou	14	20.80
Serge Abiteboul	14	20.5
Christos Faloutsos	14	18.7
Gerhard Weikum	14	16.3
Paul Erdos	14	13.9
Andrew Tanenbaum 12	12	13.0

# Demo

- Author best rank co-authorship community
- Set of authors in k-rank co-authorship core

<http://graphdegeneracy.org/>

# Conclusions

- Graph Mining is fertile research area with prominent industrial applications:
  - Web Search
  - Social networks & community mining
- Contributions
  - New metrics for community evaluation – collaboration/cohesion in terms of edges
  - Extension of k-core structure to
    - directed graphs: d-core, D-core matrix + collaboration indices
  - new aggregate bibliometric indices for collaboration in co-authorship and citation graphs?
  - Experimental evaluation on real world data sets (DBLP, Wikipedia, Epinions)

# Potential Applications of degeneracy results

## Social Networks

- “Which is the set of core members of a community, based on their intensive mutual collaboration”?
- “Is the Epinions trust network mostly positively trustworthy?”

## Scientific citation/co-authorship graphs

- “Which is the densest community of collaboration in the DBLP citation graph in *data mining*” ?
- “Which is the densest collaboration community of Dr. X in the Arxiv citation graph ?”
- “Which is the densest collaboration group in a co-authorship graph in which Dr. X belongs to?”

## Telecoms

- “Which is the most connected component of users in a telecom network based on mutual calls?”

## Biology

- “Which is the most important set of proteins in a protein interaction graph?”

# Conclusions

## Promising directions – Further work

- Extend degeneracy to **signed graphs** (underway)
- Consider additional cohesiveness parameters (connectivity, density) for community evaluation
- Applications of k-cores as preprocessing for **spectral graph clustering**
- Graph degeneracy as means for feature selection in **documents** and applications to retrieval or classification (i.e. index pruning, summarization)

# Relevant publications

- C. Giatsidis, D. Thilikos, M. Vazirgiannis, "Evaluating cooperation in communities with the k-core structure", in the proceedings of the 2011 IEEE - International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Taiwan
- Christos Giatsidis, Dimitrios M. Thilikos and Michalis Vazirgiannis, "D-cores: measuring cohesion and collaboration of directed graphs", the 11th IEEE International Conference on Data Mining, ICDM 2011, Canada (*18% selectivity*)
- Online demo (k-cores, f-cores on DBLP)  
<http://www.graphdegeneracy.org/>

# THANK YOU !!



Nikiforos Lytras, "Boat with Sail" (Tinos) 1923-26.