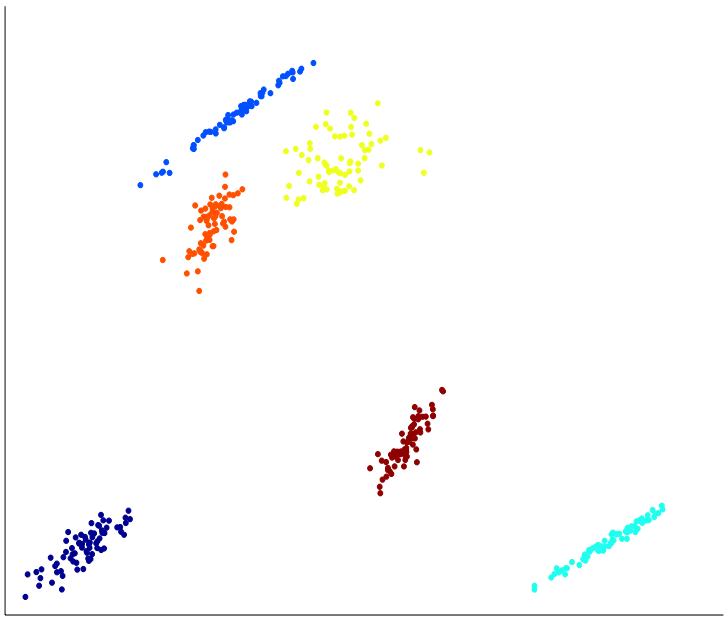
# A Tutorial on Spectral Clustering

Arik Azran
Department of Engineering
University of Cambridge

# Good clustering – we know it when we see it



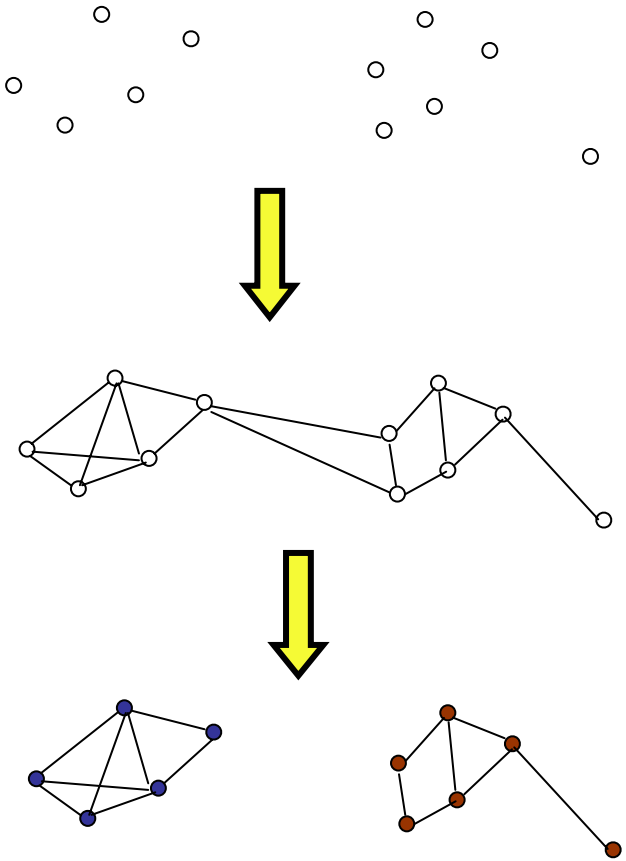
A mixture (of Gaussians) is a good model for this data set

But clusters are not always about Euclidean distance and parametric models are not always appropriate

# What is and what is not in the talk

**First** - graph representation of data (largely, application dependent)

**Then** - graph partitioning



**In this talk** – mainly how to find a good partitioning of a given graph using spectral properties of that graph

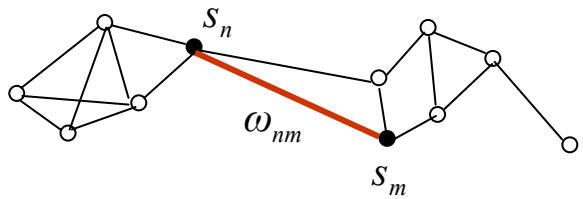
# Plan

- **Graph partitioning**
- Algorithms
- Possible explanations
- Self tuning algorithm
- Practicalities and comments

# Some Graph Terminology

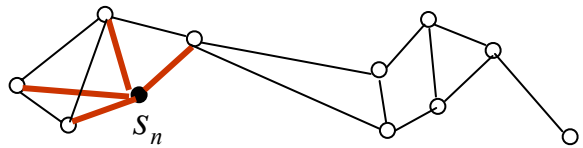
pairwise affinity (similarity)

$$\omega_{nm} = e^{-\frac{\|s_n - s_m\|}{\sigma^2}}$$



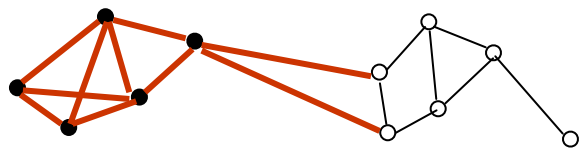

node volume (degree)

$$D_n = \sum_{m=1}^{N} \omega_{nm}$$



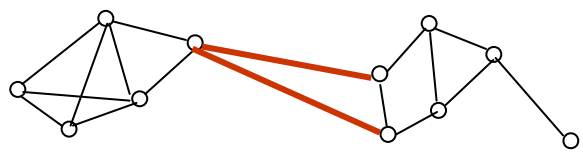

volume of a set (cluster)

$$Vol(C) = \sum_{n \in C} D_n$$

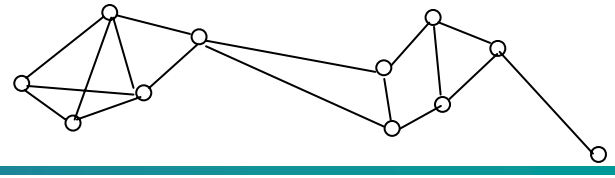


cut between 2 sets

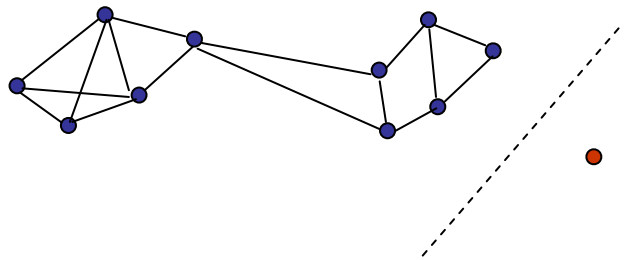
$$Cut(C_1, C_2) = \sum_{n \in C_1} \sum_{m \in C_2} \omega_{nm}$$



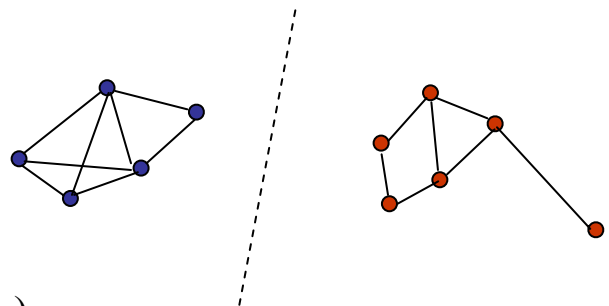
# Graph Cuts



minimal bipartition cut $\min \; Cut(C_1, C_2)$



minimal bipartition normalized cut



$$\min \; \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_2, C_1)}{Vol(C_2)} = \min \; \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

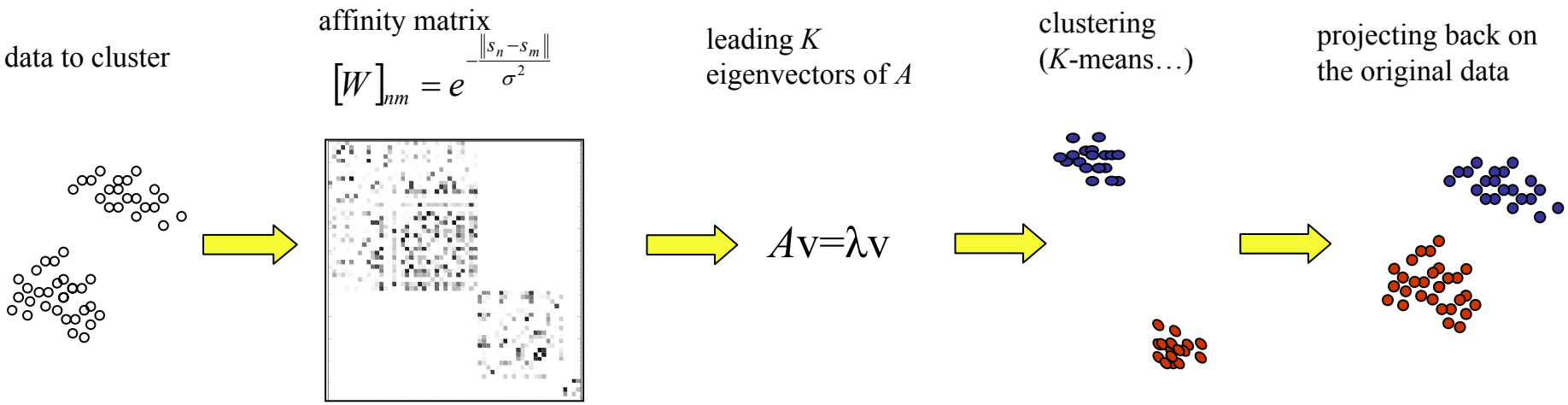
**Problem**: finding an optimal graph (normalized) cut is NP-hard

**Approximation**: spectral graph partitioning

# Plan

- Graph partitioning
- **Algorithms**
- Possible explanations
- Self tuning algorithm
- Practicalities and comments

# Spectral clustering - overview




Main difference between algorithms is the definition of $A = \mathrm{func}(W)$

## On Spectral Clustering: Analysis and an algorithm

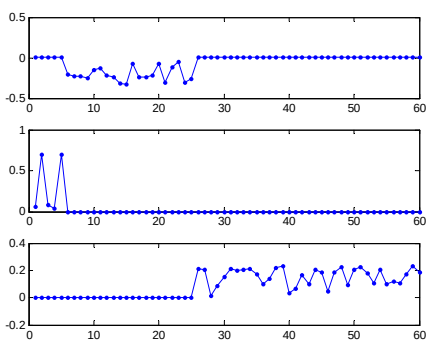

**Andrew Y. Ng**
CS Division
U.C. Berkeley
*ang@cs.berkeley.edu*

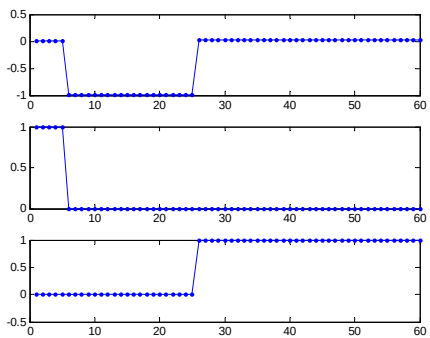
**Michael I. Jordan**
CS Div. & Dept. of Stat.
U.C. Berkeley
*jordan@cs.berkeley.edu*

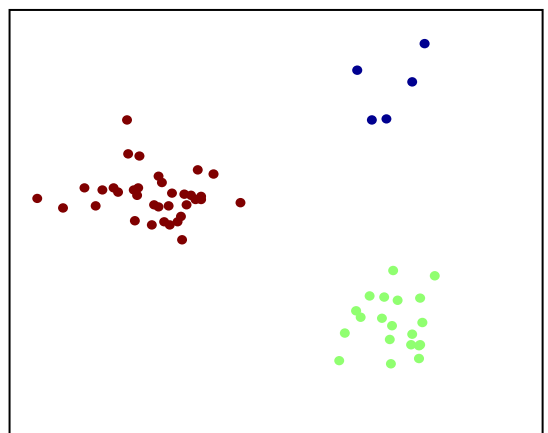
**Yair Weiss**
School of CS & Engr.
The Hebrew Univ.
*yweiss@cs.huji.ac.il*


1. set $\omega_{nn} = 0$ ; $D_n = \sum_{m=1}^{N} \omega_{nm}$ ; $D = \mathrm{diag}(D_1, D_2, \ldots, D_N)$
2. compute $A = D^{-0.5} W D^{-0.5}$
3. compute the K leading eigenvectors of A; set $V = [v_1, v_2, \ldots, v_K]$
4. set $U_{nk} = \dfrac{v_n(k)}{\sum_{k'=1}^{K} v_n(k')}$
5. view each row of $U$ as a point in $\mathrm{R}^{\mathrm{K}}$; cluster them into K clusters using K-means

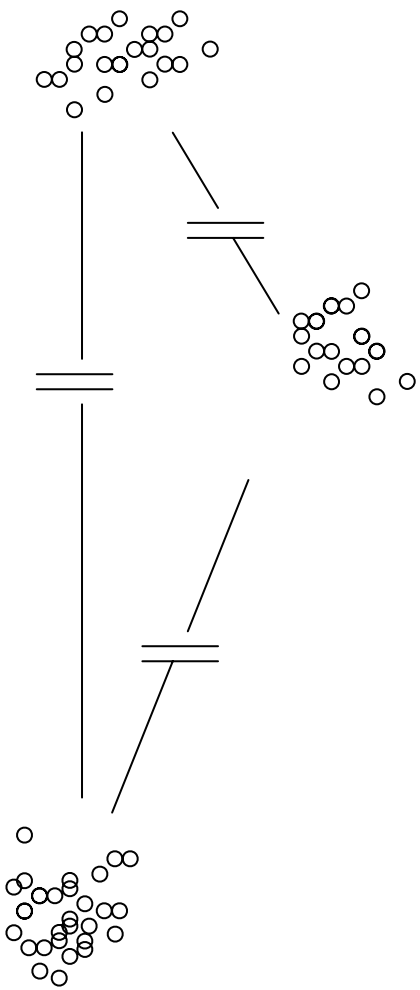
$W; A$ $\quad V = [v_1, v_2, v_3]$ $\quad U = [u_1, u_2, u_3]$







# Informal discussion: the ‘ideal’ case



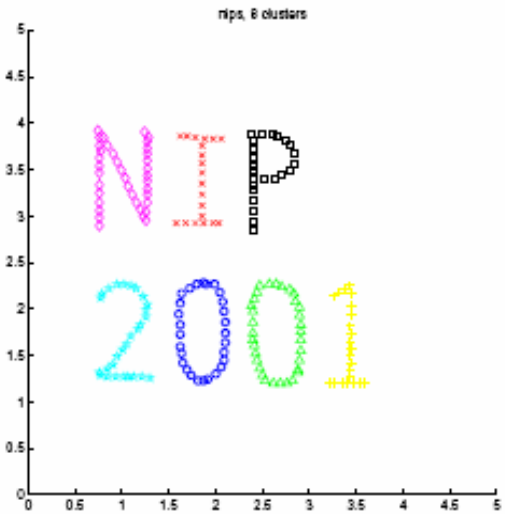
$$W \cong \begin{bmatrix} 1_{N_1 \times N_1} & 0_{N_1 \times N_2} & 0_{N_1 \times N_3} \\ 0_{N_2 \times N_1} & 1_{N_2 \times N_2} & 0_{N_2 \times N_3} \\ 0_{N_3 \times N_1} & 0_{N_3 \times N_2} & 1_{N_3 \times N_3} \end{bmatrix} \qquad A \cong \begin{bmatrix} \frac{1}{N_1} 1_{N_1 \times N_1} & 0_{N_1 \times N_2} & 0_{N_1 \times N_3} \\ 0_{N_2 \times N_1} & \frac{1}{N_2} 1_{N_2 \times N_2} & 0_{N_2 \times N_3} \\ 0_{N_3 \times N_1} & 0_{N_3 \times N_2} & \frac{1}{N_3} 1_{N_3 \times N_3} \end{bmatrix}$$

$$V \cong \left[ \frac{1}{\sqrt{N_1}} \begin{bmatrix} 1_{N_1 \times 1} \\ 0_{N_2 \times 1} \\ 0_{N_3 \times 1} \end{bmatrix}, \frac{1}{\sqrt{N_2}} \begin{bmatrix} 0_{N_1 \times 1} \\ 1_{N_2 \times 1} \\ 0_{N_3 \times 1} \end{bmatrix}, \frac{1}{\sqrt{N_3}} \begin{bmatrix} 0_{N_1 \times 1} \\ 0_{N_2 \times 1} \\ 1_{N_3 \times 1} \end{bmatrix} \right]$$

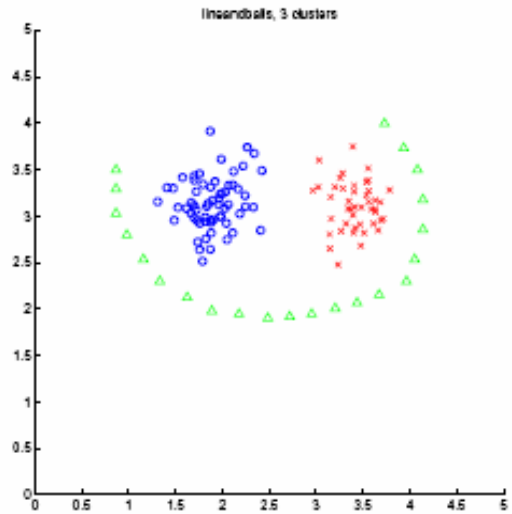
$$U \cong \left[ \begin{bmatrix} 1_{N_1 \times 1} \\ 0_{N_2 \times 1} \\ 0_{N_3 \times 1} \end{bmatrix}, \begin{bmatrix} 0_{N_1 \times 1} \\ 1_{N_2 \times 1} \\ 0_{N_3 \times 1} \end{bmatrix}, \begin{bmatrix} 0_{N_1 \times 1} \\ 0_{N_2 \times 1} \\ 1_{N_3 \times 1} \end{bmatrix} \right]$$

- eigenvectors are orthogonal

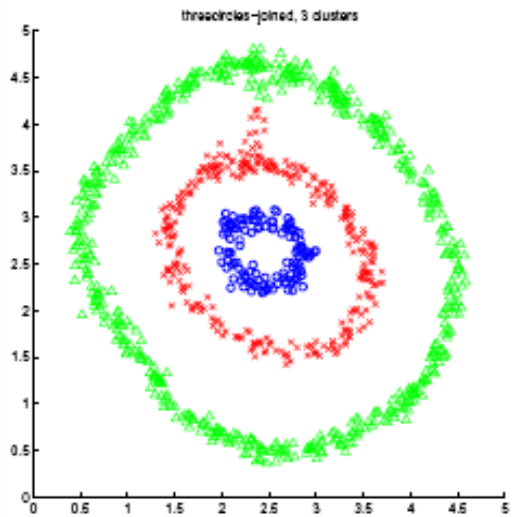
- clustering rows of $U$ correspond to clustering points in the ‘feature’ space


nips, 8 clusters
NIP
2001

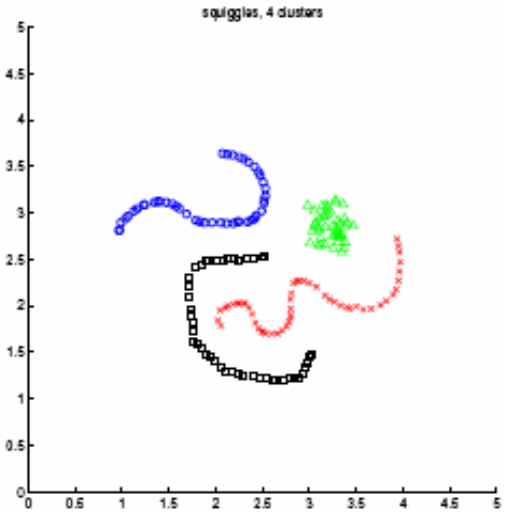
(a)


fiveandballs, 3 clusters

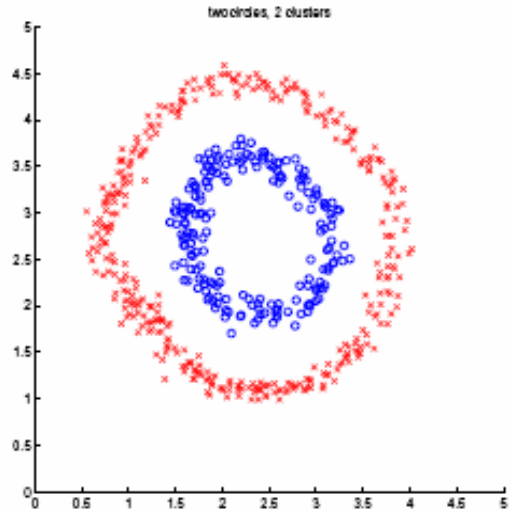
(b)


threecircles-joined, 3 clusters

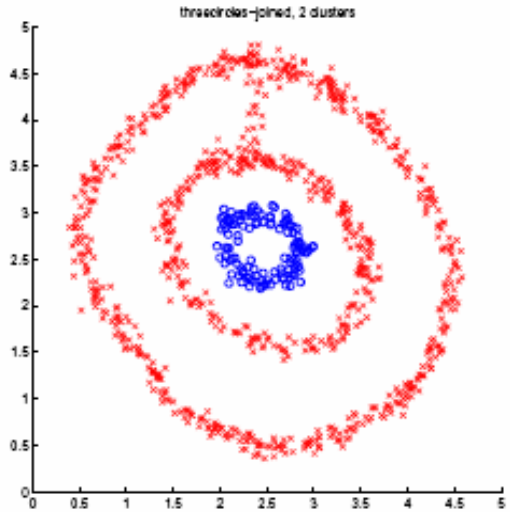
(g)


squiggles, 4 clusters

(d)


twocircles, 2 clusters

(e)

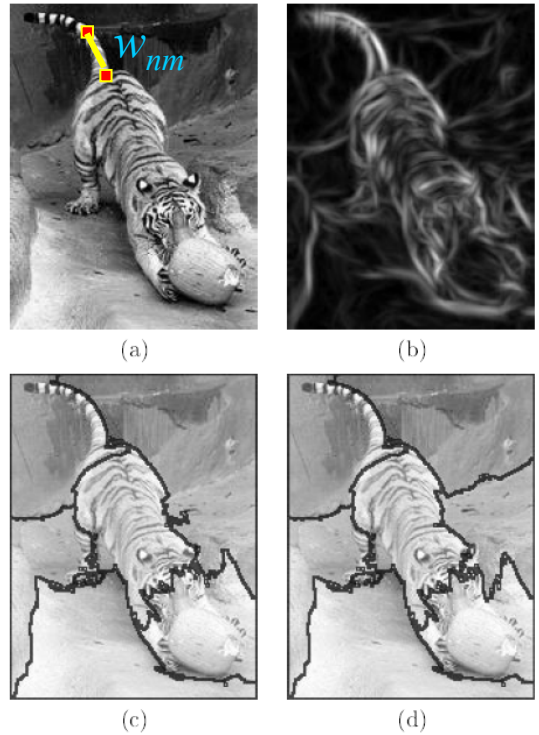

threecircles-joined, 2 clusters

(f)

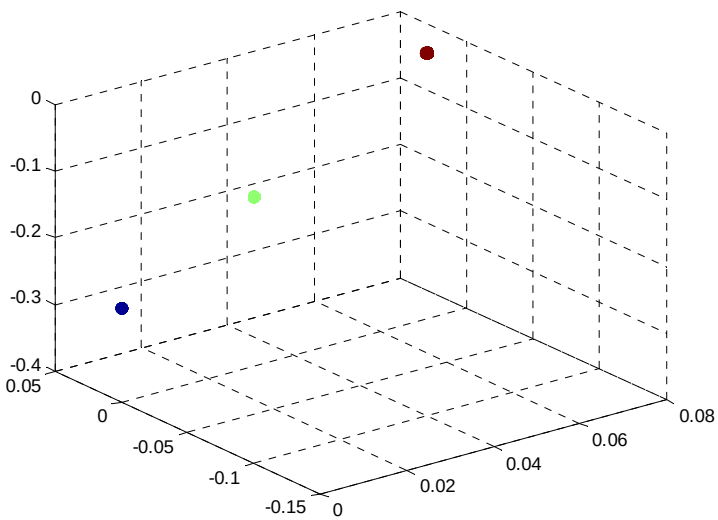
## A Random Walks View of Spectral Segmentation

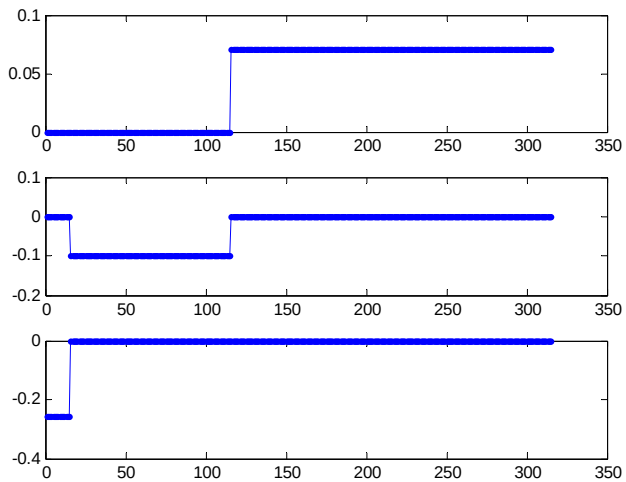
**Marina Meilă**
University of Washington
mmp@stat.washington.edu

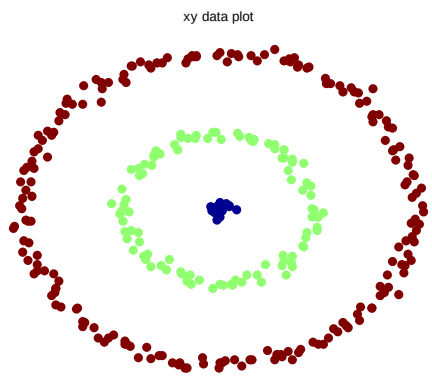
**Jianbo Shi**
Carnegie Mellon University
jshi@cs.cmu.edu




1. set $P = D^{-1}W$
2. compute the K leading eigenvectors of P; set $V = [v_1, v_2, \ldots, v_K]$
3. view each row of $V$ as a point in $R^K$ ; cluster them into K clusters using K-means



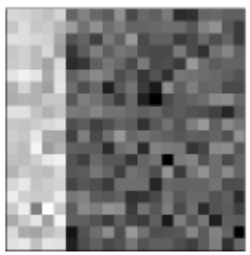



# Normalized Cuts and Image Segmentation

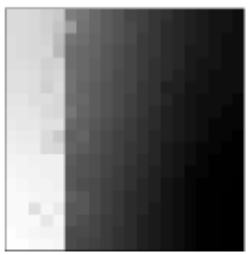
Jianbo Shi and Jitendra Malik, *Member, IEEE*

1. set $L = D - W$
2. compute the K eigenvectors of $Lv = \lambda Dv$ associated with the smallest eigenvalues
3. set $V = [v_1, v_2, \ldots, v_K]$
4. view each row of $V$ as a point in $R^K$ ; cluster them into K clusters using K-means

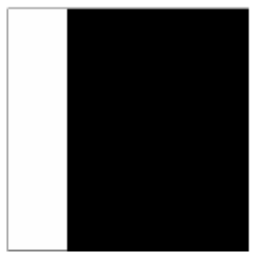
noisy 'step' image



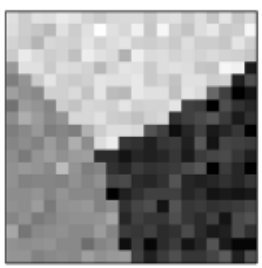
eigenvector of second smallest eigenvalue



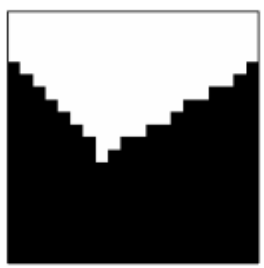
resulting partitioning

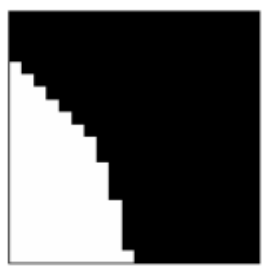


noisy image



resulting three segments





# Connections

| | Affinity Matrix | Eigenproblem to solve | Chosen eigenvectors | Processing of eigenvectors |
|---|---|---|---|---|
| Ng Jordan Weiss | $D^{-0.5}WD^{-0.5}$ | $D^{-0.5}WD^{-0.5}u = \beta u$ | $U = [u_1, u_2, \ldots, u_K]$ (largest eigenvalues) | row normalization |
| Meila Shi | $D^{-1}W$ | $D^{-1}Wv = \lambda v$ | $V = [v_1, v_2, \ldots, v_K]$ (largest eigenvalues) | none |
| Shi Malik | $D - W$ | $(D - W)r = \mu Dr$ | $R = [r_N, r_{N-1}, \ldots, r_{N-1+K}]$ (smallest eigenvalues) | none |

$$D^{-0.5}WD^{-0.5}u = \beta u \quad \Rightarrow \quad D^{-1}WD^{-0.5}u = \beta D^{-0.5}u \quad \Rightarrow \quad \lambda = \beta;\ v = D^{-0.5}u$$

$$(D - W)r = \mu Dr \quad \Rightarrow \quad D^{-1}(D - W)r = \mu r \quad \Rightarrow \quad D^{-1}Wr = (1 - \mu)r \quad \Rightarrow \quad \lambda = 1 - \mu;\ v = r$$

| | NJW | MS | SM |
|---|---|---|---|
| NJW | | $\beta = \lambda;\ u = D^{0.5}v$ | $\beta = 1 - \mu;\ u = D^{0.5}r$ |
| MS | $\lambda = \beta;\ v = D^{-0.5}u$ | | $\lambda = 1 - \mu;\ v = r$ |
| SM | $\mu = 1 - \beta;\ r = D^{-0.5}u$ | $\mu = 1 - \lambda;\ r = v$ | |

# Plan

- Graph partitioning
- Algorithms
- **Possible explanations**
- Self tuning algorithm
- Practicalities and comments

# Possible Explanations

1. Shi and Malik showed that an approximate solution to the minimum **normalized cut** can be found using the Graph Laplacian

$$\min \left( \frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

2. Graph Laplacian and **smooth functions**

3. Random walks, **piecewise constant eigenvectors** and minimum **multiway normalized cut**

# Graph Laplacian and Smooth Functions

**Definition**: $L = D - W$

**Key property**: $v' L v = \sum_{n=1}^{N} \sum_{m=1}^{N} (v(n) - v(m))^2 w_{nm}$

**Implication**: if $Lv = \lambda v$ and $v'v = 1$ then $\lambda = \sum_{n=1}^{N} \sum_{m=1}^{N} (v(n) - v(m))^2 w_{nm}$ and eigenvectors with low eigenvalues are smooth functions on the graph defined by W

Normalized graph Laplacian:

**Left-Right normalization**: $L_{LR} = D^{-0.5} L D^{-0.5} = D^{-0.5}(D - W)D^{-0.5} = I - D^{-0.5} W D^{-0.5}$

$$v' L_{LR} v = \sum_{n=1}^{N} \sum_{m=1}^{N} \left( \frac{v(n)}{\sqrt{D_n}} - \frac{v(m)}{\sqrt{D_m}} \right)^2 w_{nm}$$

**Left normalization**: $L_L = D^{-1} L = D^{-1}(D - W) = I - D^{-1} W$

# Random Walks

**Definition**: $P = D^{-1}W$ can be viewed as a first order Markov transition matrix

$$P_{nm} = \frac{w_{nm}}{\sum_{j=1}^{N} w_{nj}} = \text{probability of transitioning from node n to node m in a single step}$$

**Key properties**: if $Pv_n = \lambda_n v_n$ then

1. $\lambda_1 = 1$ and $v_1 = [1,1,\ldots,1]^T$
2. $\lambda_n, v_n$ are real and $|\lambda_n| \le 1 \quad \forall \quad n = 2,3,\ldots,N$
3. smoothness $v_n(m) = \lambda_n \sum_{j=1}^{N} v_n(j) P_{mj}$

**Multiway Normalized Cut**: find a partitioning such that the probability of transitioning between clusters is minimized

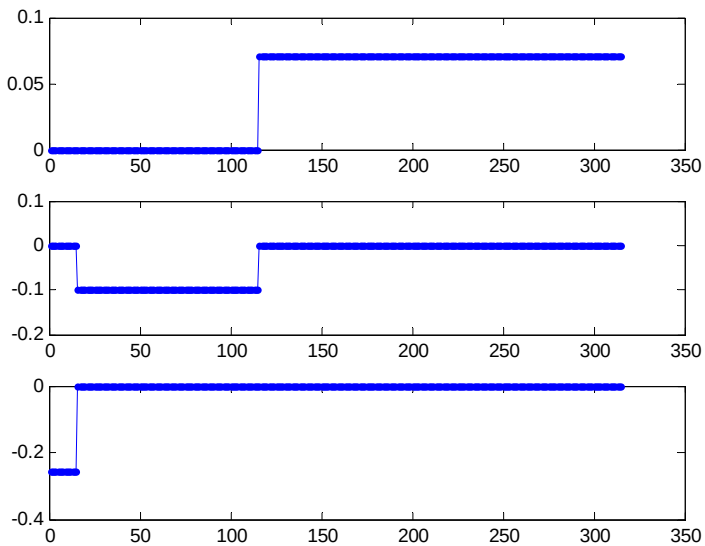
$$MNCut(C) = \sum_{k=1}^{K} \left(1 - \Pr(C_k \to C_k \mid C_k)\right) = \sum_{k=1}^{K} \left(1 - \frac{\sum_{n,m \in C_k} w_{nm}}{\sum_{n \in C_k} \sum_{m=1}^{N} w_{nm}}\right)$$

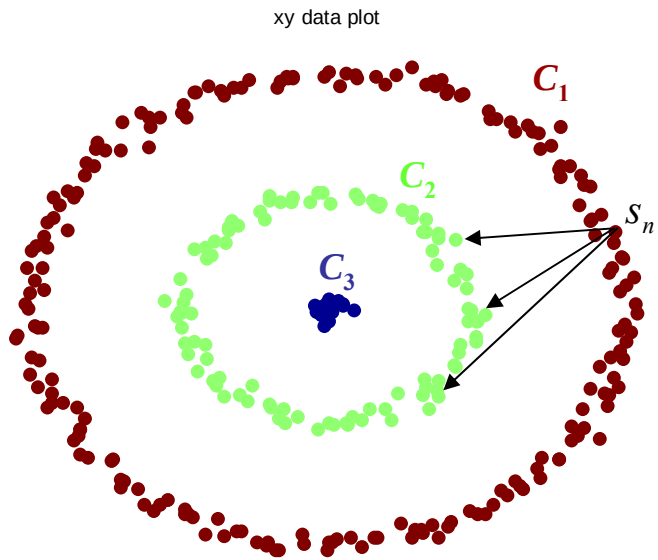
# The Multicut Lemma (Meila, 02)

If there exists a K-way partitioning $C = \{C_1, C_2, \ldots, C_K\}$ and a set of scalars $\{\widetilde{P}_{kk'}\}_{k,k'=1}^{K}$ such that

$$\Pr\big((n \in C_k) \rightarrow C_{k'}\big) = \sum_{m \in C_{k'}} P_{nm} = \widetilde{P}_{kk'} \qquad \forall n \in C_k,\ k \neq k'$$

then P has K Piecewise Constant Eigenvectors and the Meila-Shi algorithm is guarenteed to find the minimizer of the MNCut



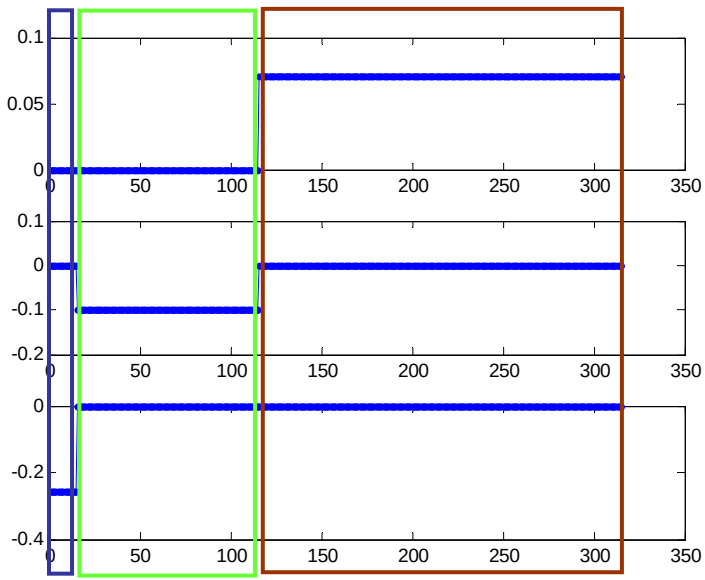



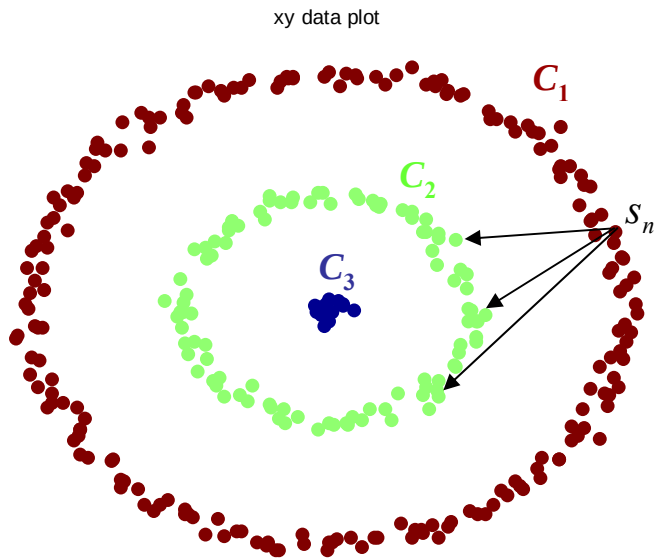
# The Multicut Lemma (Meila, 02)

If there exists a K-way partitioning $C = \{C_1, C_2, \ldots, C_K\}$ and a set of scalars $\{\widetilde{P}_{kk'}\}_{k,k'=1}^{K}$ such that

$$\Pr\big((n \in C_k) \rightarrow C_{k'}\big) = \sum_{m \in C_{k'}} P_{nm} = \widetilde{P}_{kk'} \qquad \forall n \in C_k,\ k \neq k'$$

then P has K Piecewise Constant Eigenvectors and the Meila-Shi algorithm is guarenteed to find the minimizer of the MNCut



# Plan

- Graph partitioning
- Algorithms
- Possible explanations
- **Self tuning algorithm**
- Practicalities and comments

# About the principal components of a random walk

Consider letting $N$ particles explore the data set according to $P$. The distribution for the location of these particles after taking $t$ steps is summarized by $P^t$, which satisfies the following.

*1. for any $t = 1, 2 \ldots$ the matrix $P^t$ is given by*

$$P^t = \sum_{n=1}^{N} \lambda_n^t \frac{v_n v_n^\top D}{v_n^\top D v_n}.$$

*2. the set of matrices $\left\{ \frac{v_n v_n^\top D}{v_n^\top D v_n} \right\}_{n=1}^{N}$ with weights $\{\lambda_n^t\}_{n=1}^{N}$ form an idempotent-orthogonal basis of $P^t$, meaning that*

$$\frac{v_n v_n^\top D}{v_n^\top D v_n} \cdot \frac{v_m v_m^\top D}{v_m^\top D v_m} = \frac{v_n v_n^\top D}{v_n^\top D v_n} \delta_{nm}$$

# About the principal components of a random walk

In PCA the eigenvalues indicate the data variance in the directions of the principal components, here the power series of the eigenvalues indicate how relevant are the associated basis matrices in synthesizing $P^t$

*1. for any $t = 1, 2 \ldots$ the matrix $P^t$ is given by*

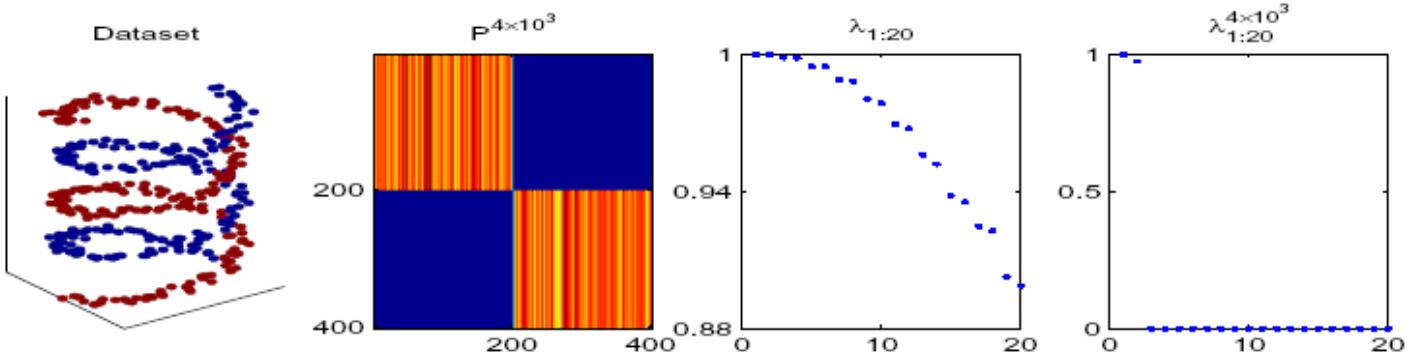
$$P^t = \sum_{n=1}^{N} \lambda_n^t \frac{v_n v_n^\top D}{v_n^\top D v_n}.$$

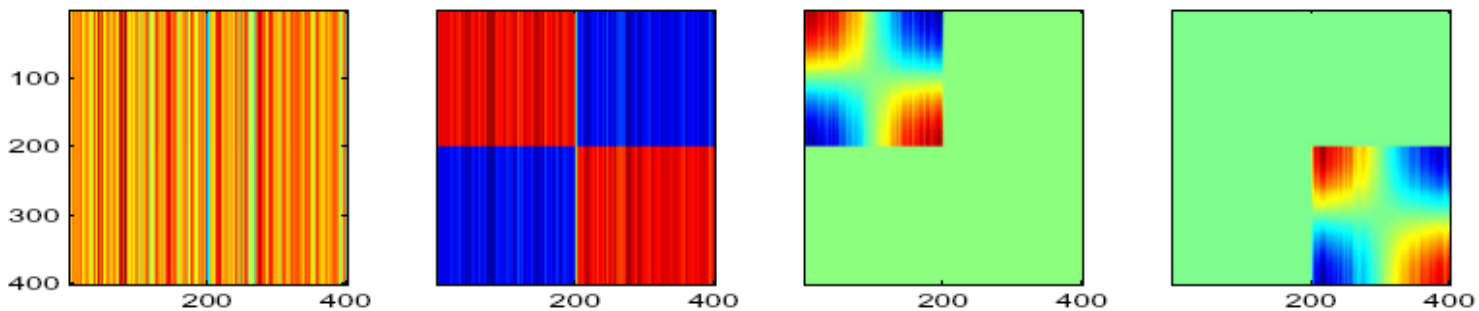
*2. the set of matrices $\left\{ \frac{v_n v_n^\top D}{v_n^\top D v_n} \right\}_{n=1}^{N}$ with weights $\{\lambda_n^t\}_{n=1}^{N}$ form an idempotent-orthogonal basis of $P^t$, meaning that*

$$\frac{v_n v_n^\top D}{v_n^\top D v_n} \cdot \frac{v_m v_m^\top D}{v_m^\top D v_m} = \frac{v_n v_n^\top D}{v_n^\top D v_n} \delta_{nm}$$

In PCA the principal components are orthogonal, here the basis matrices are idempotent-orthogonal

# Example – 2 spirals




$$\frac{v_1 v_1^\top D}{v_1^\top D v_1} \qquad \frac{v_2 v_2^\top D}{v_2^\top D v_2}$$

Idempotent-Orthogonal basis:

$$\frac{v_1 v_1^\top D}{v_1^\top D v_1}\frac{v_1 v_1^\top D}{v_1^\top D v_1} = \frac{v_1 v_1^\top D}{v_1^\top D v_1}$$

$$\frac{v_1 v_1^\top D}{v_1^\top D v_1}\frac{v_2 v_2^\top D}{v_2^\top D v_2} = 0$$

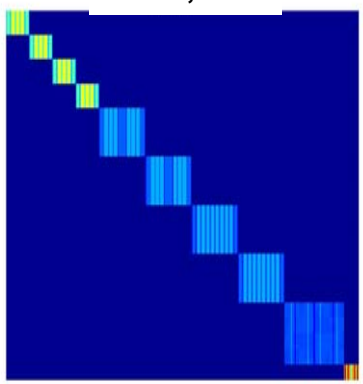
Low rank approximation:

$$\exists t \text{ such that } P^t \approx \frac{v_1 v_1^\top D}{v_1^\top D v_1} + \frac{v_2 v_2^\top D}{v_2^\top D v_2}$$

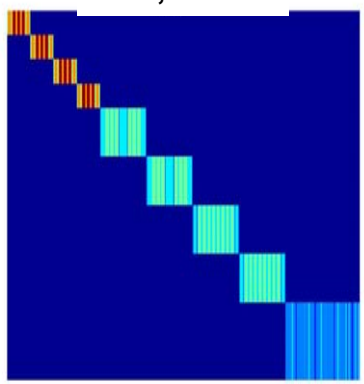
# Example – clusters in increasing scales

**Algorithm**: given $K$, find $t$ and then cluster the rows of $P^t$

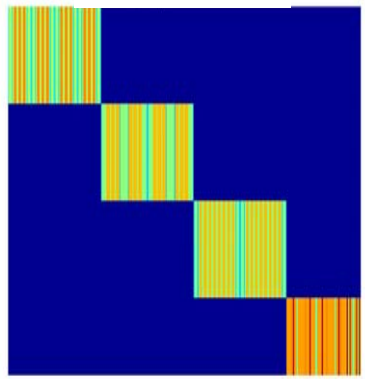
K=10; $P^{342}$

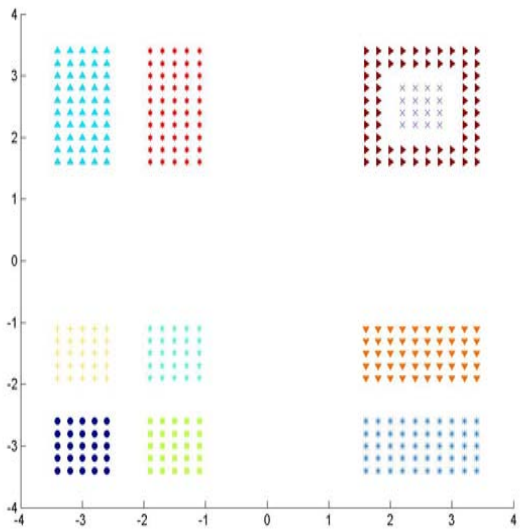


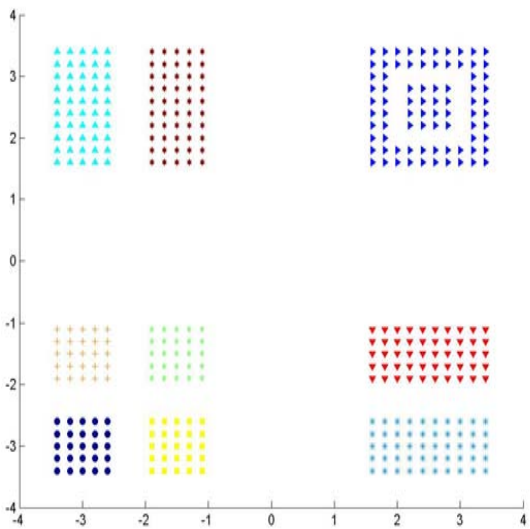
K=9; $P^{210482}$

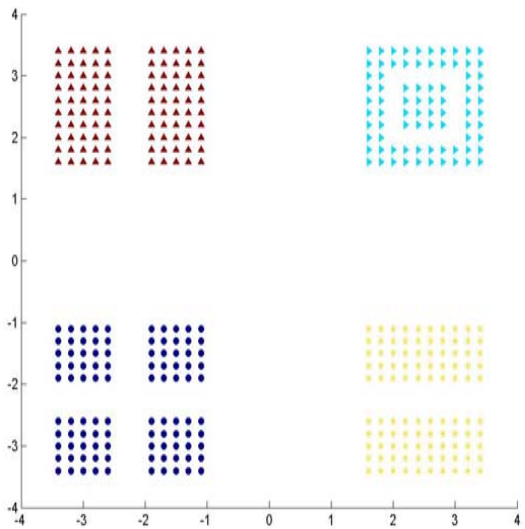


K=4; $P^{4.77e11}$









# Learning *t* given *K*

If the data has *K* well separated clusters of any shape, then there exist *t* such that

$$P^t \approx \sum_{n=1}^{K} \lambda_n^t T_n \approx \sum_{n=1}^{K} 1 \cdot \frac{v_n v_n^\top D}{v_n^\top D v_n}$$

$\Rightarrow$ the first *K* eigenvectors contain most of the information that is relevant to the data exploration

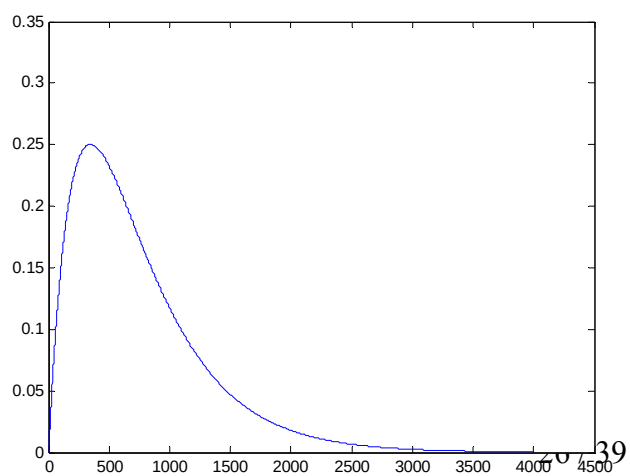
$\Rightarrow$ the *K*'th eigengap is maximized: $\lambda_K^t - \lambda_{K+1}^t \to 1$

So, a value of *t* that reveals *K* clusters can be found by

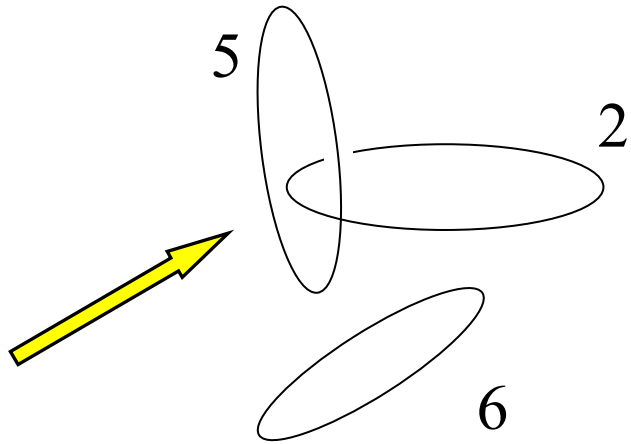
$$t_K = \underset{t}{\operatorname{argmax}} \left( \lambda_K^t - \lambda_{K+1}^t \right)$$

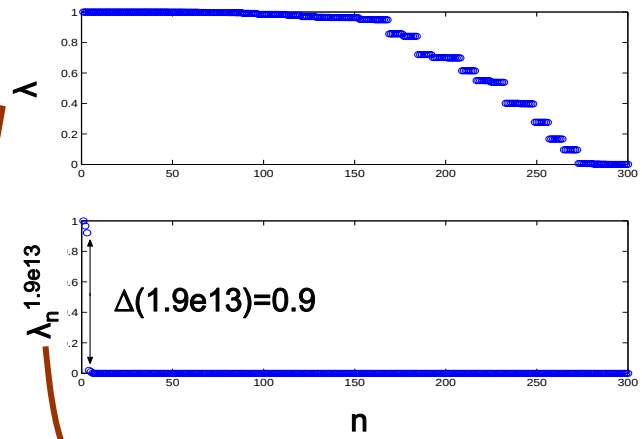
This can be found analytically and given by

$$t_K = \left[ \frac{\ln\left( \frac{\ln(|\lambda_{K+1}|)}{\ln(|\lambda_K|)} \right)}{\frac{\ln(|\lambda_K|)}{\ln(|\lambda_{K+1}|)}} \right]_{\text{even}}$$



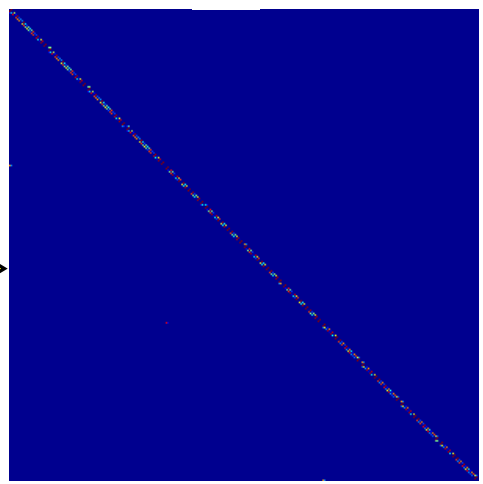
# Example – 300 randomly rotated digits



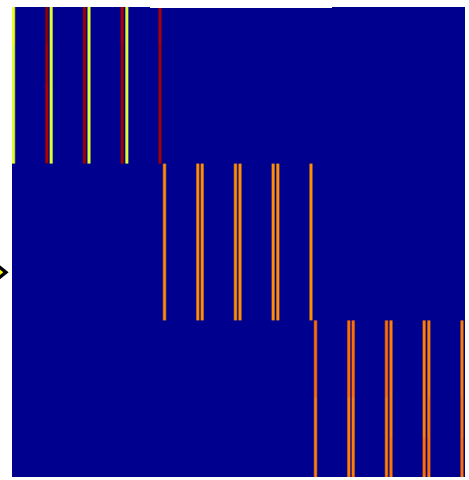




P

P^1.9e13



$$t_3 = \left[ \frac{\ln\left(\frac{\ln(|\lambda_4|)}{\ln(|\lambda_3|)}\right)}{\frac{\ln(|\lambda_3|)}{\ln(|\lambda_4|)}} \right] = 1.9 \cdot 10^{13}$$

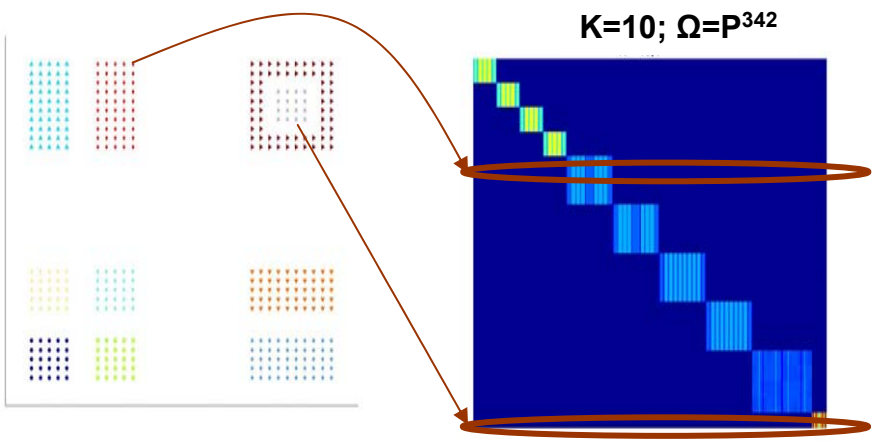


# The *K*-prototypes algorithm

Clustering in the feature space: minimize the loss function

$$J(Q,\mathcal{I}) = \sum_{k=1}^{K}\sum_{m\in\mathcal{I}_k} \mathrm{KL}(\Omega_m||Q_k) = \sum_{k=1}^{K}\sum_{m\in\mathcal{I}_k}\sum_{n=1}^{N} \Omega_{mn}\ln\frac{\Omega_{mn}}{Q_{kn}}$$

where $Q$ is a $K$x$N$ matrix with $Q_k$ as its $k^{\text{th}}$ row, and $\Omega_m$ the $m^{\text{th}}$ row of $P^t$




**Input**: $t$'th order transition matrix $\Omega$, number of cluster $K$
**Output**: A partitioning $\mathcal{I}$, a matrix of prototypes $Q$
**Algorithm**:
0. Initialize: $Q^{(old)}$
1. **Assign points**: for $k = 1, 2, \ldots, K$ set

$$\mathcal{I}_k^{(new)} = \left\{m : k = \mathrm{argmin}_k \,\mathrm{KL}\left(\Omega_m||Q_k^{(old)}\right)\right\}$$

2. **Update prototypes**: for $k = 1, 2, \ldots, K$ set

$$Q_k^{(new)} = \frac{1}{|\mathcal{I}_k^{(new)}|}\sum_{m\in\mathcal{I}_k^{(new)}}\Omega_m$$

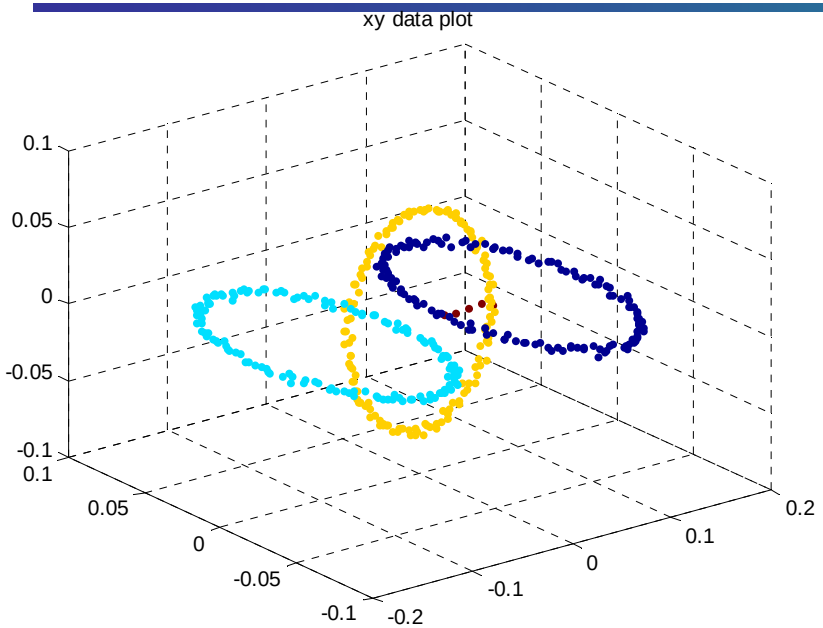
3. Set $Q^{(old)} = Q^{(new)}$.
4. If stop condition is satisfied, stop. Otherwise go to 1.

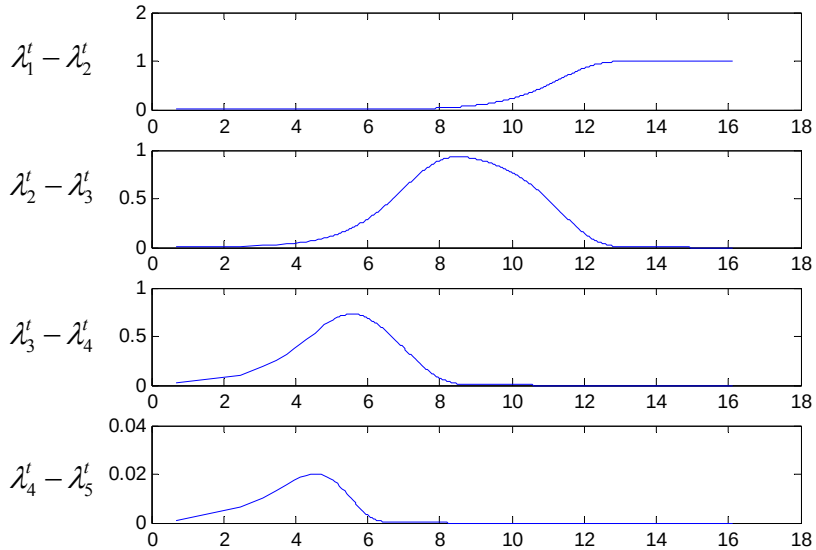
# Learning pairs of $t$ & $K$

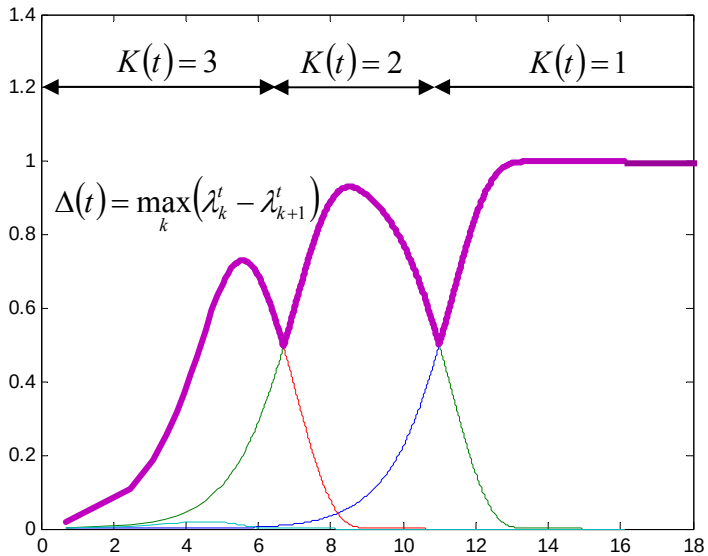
So far the algorithm, given the number of clusters $K$, finds a number of steps $t_K$ such that particles mix well within 'their' cluster, but have little probability of transitioning to other clusters

$t$ increase $\Rightarrow$ $K$ decrease: we can learn good pairs of $(K, t_K)$ by scanning over $t$ and looking for local maximum of the maximal eigengap

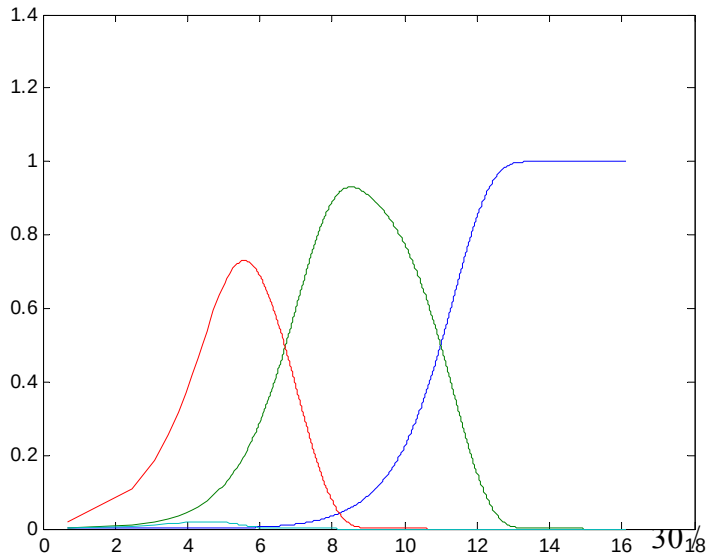
# Learning pairs of $t$ & $K$




$$K(t)=\arg\max_k\left(\lambda_k^t-\lambda_{k+1}^t\right)$$



# Algorithm for learning pairs of $t$ & $K$

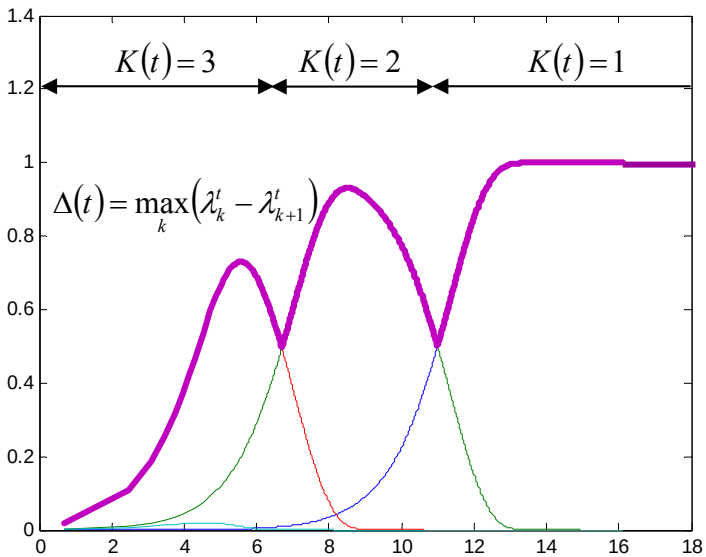
1. find all local maxima of

$$\Delta(t) = \max_k \left( \lambda_k^t - \lambda_{k+1}^t \right)$$

2. for each of these, set the number of clusters to be

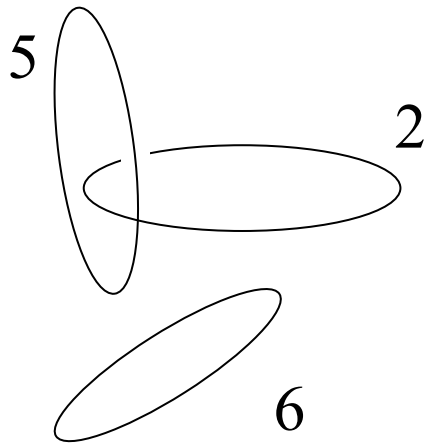
$$K = \arg\max_k \left( \lambda_k^{t_K} - \lambda_{k+1}^{t_K} \right)$$

where $t_K$ is the number of steps where the $K^{th}$ local maxima is achieved and cluster the data using the pair $(K, t_K)$

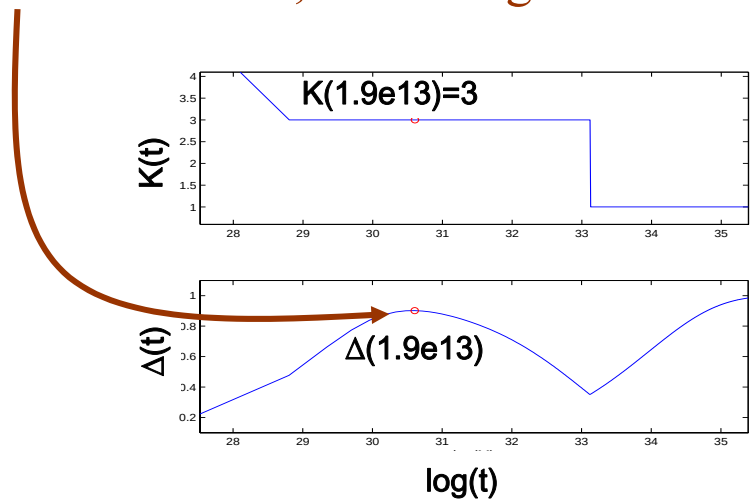


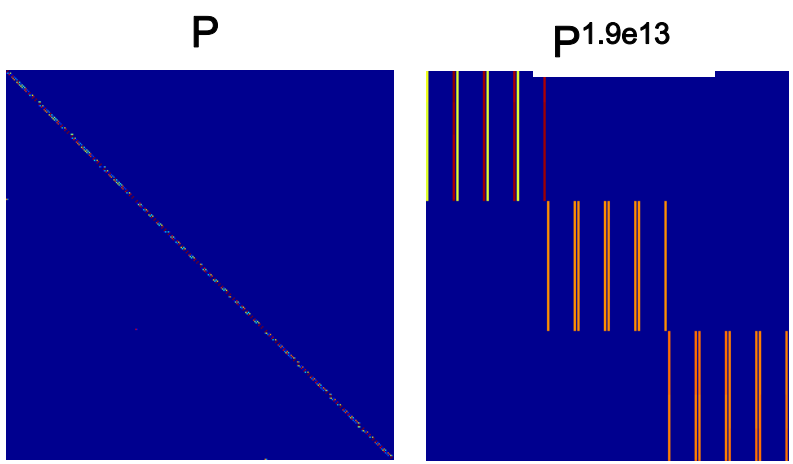

$K(t) = \arg\max_k \left( \lambda_k^t - \lambda_{k+1}^t \right)$

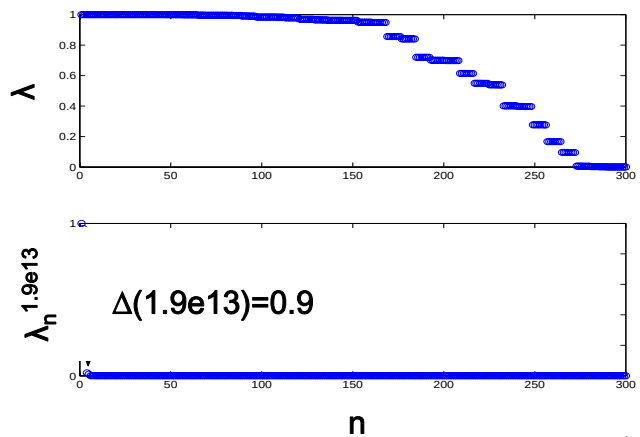
# Example – 300 randomly rotated digits (2)



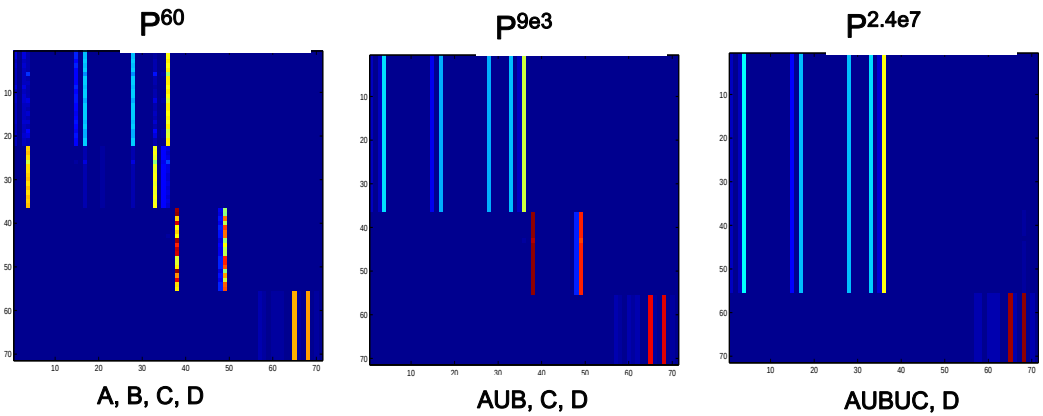

a single local maximum, indicating that $K$=3

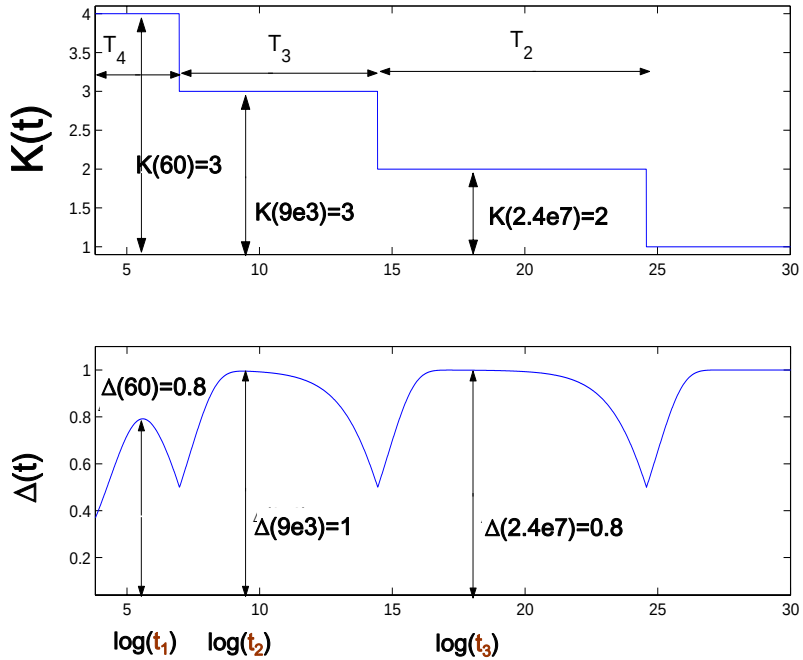


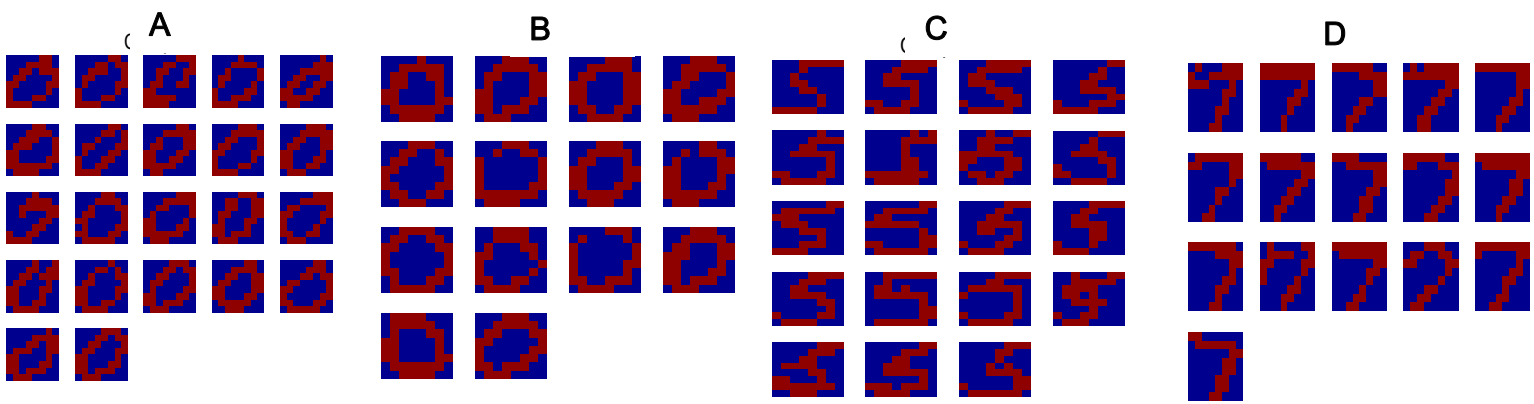



# Example - learning pairs of *t* & *K*

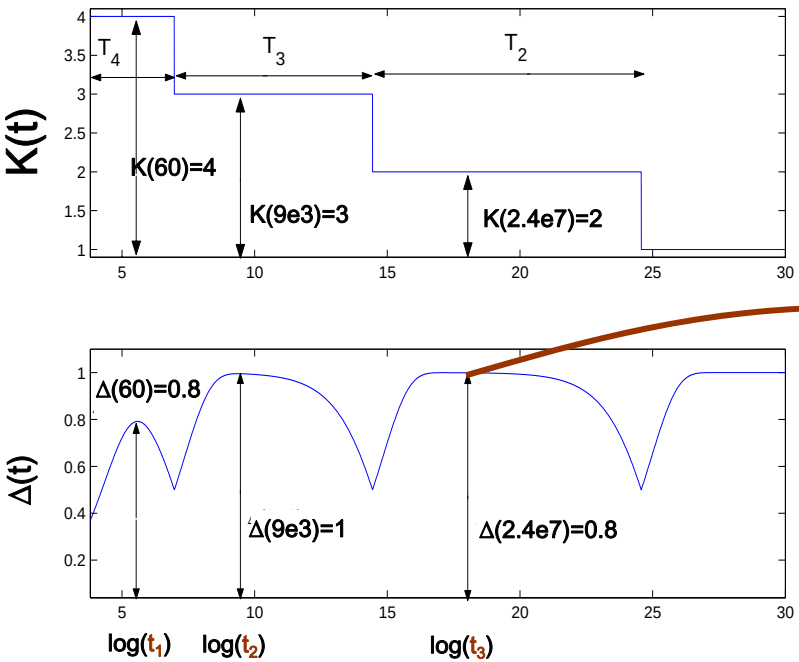


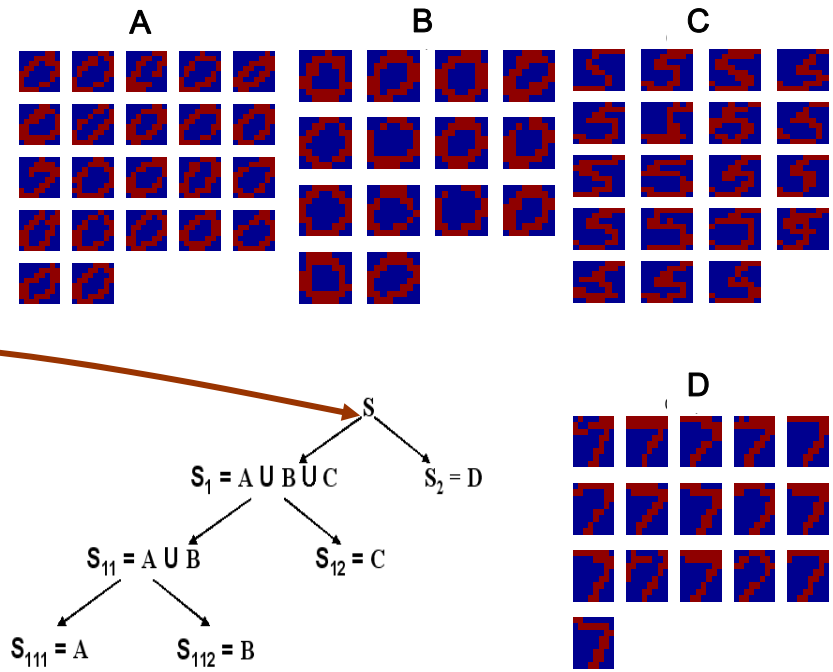



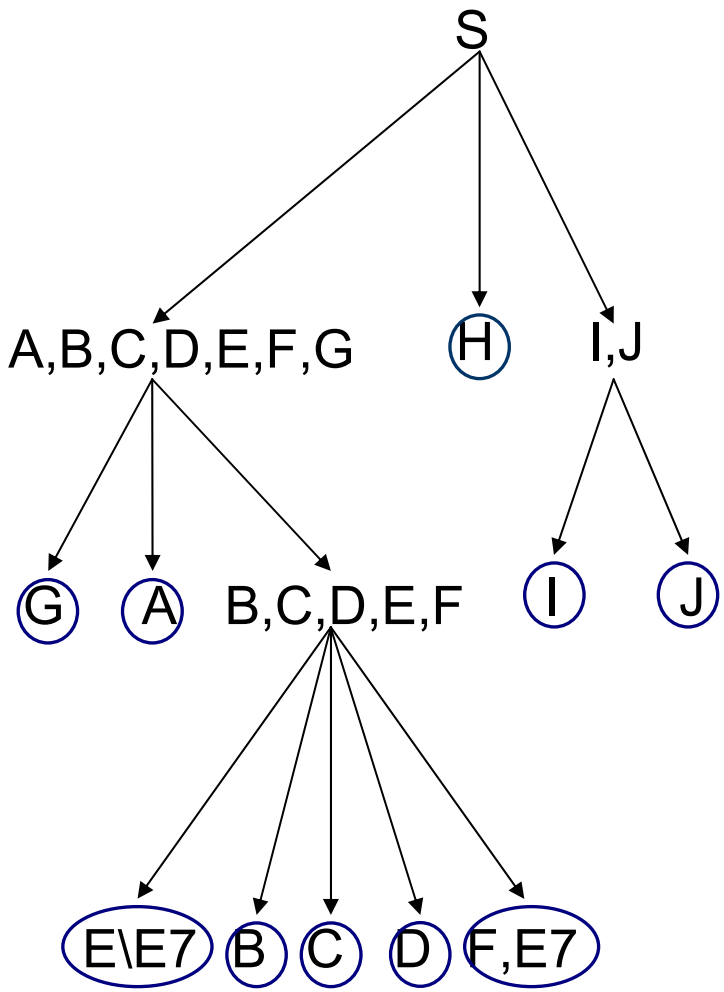
# Hierarchical clustering (1)

Instead of several partitionings in different scales, find a single tree where each split corresponds to partitioning in different scale:



# Hierarchical clustering (2)

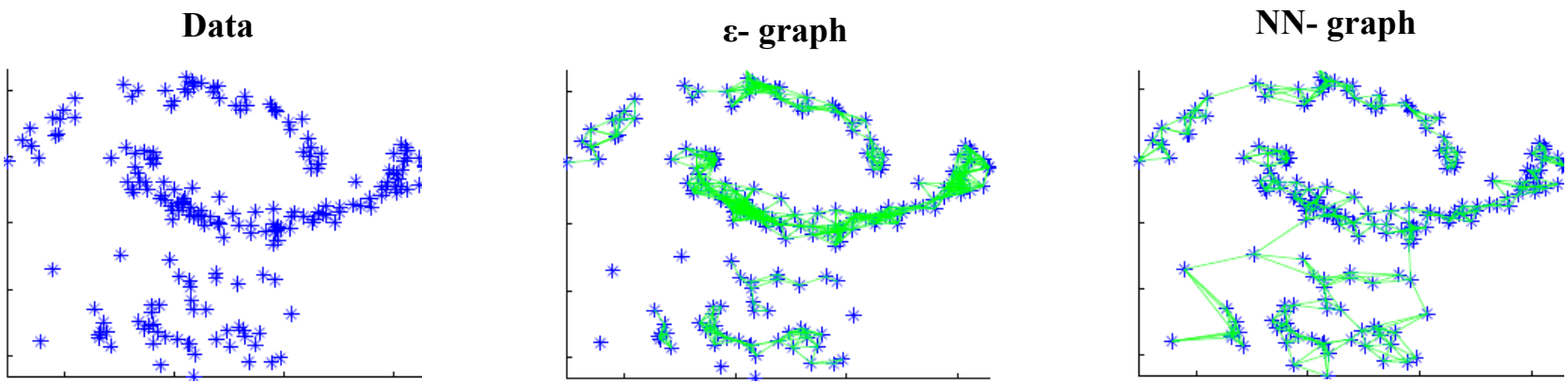


# Plan

- Graph partitioning
- Algorithms
- Possible explanations
- Self tuning algorithm
- **Practicalities and comments**

# Some implementation issues

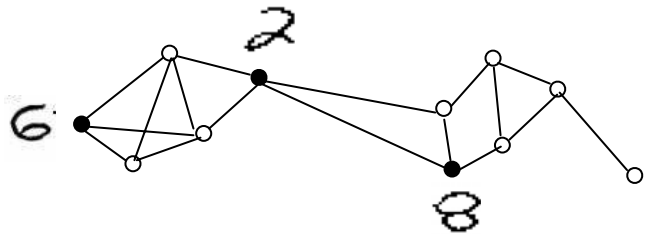
- Initializing the length scale
  - global: $\sigma = \mathrm{O}\left(\min \left\| s_n - s_m \right\| \mid m \neq n\right)$
  - local: $\sigma_n = \mathrm{O}\left(\min \left\| s_n - s_m \right\| \mid w_{nm} > 0\right)$
- Sparsifying $W$: threshold all entries that are very small, or only keep a small number of neighbors for each point




- Similarity of a point and itself: should outliers be considered singleton clusters?
- Which matrix/algorithm to use?
- How to find a good graph representation of data? Need to use domain expertise.

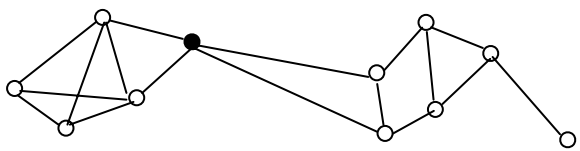
# Connection to other problems

**Semisupervised Learning**: given a small number of labels, estimate the missing labels




**Active Learning**: which missing label will would be most beneficial to know?

**Ranking**: order all data items accordance to their `relevance' to a given (one or more) items of interest in data,



**Question**: can we find a unified framework to tackle all these problems?

# Short bibliography


1. J. Shi and J. Malik. Normalized Cuts and Image Segmentation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, 2000.
2. Marina Meila and Jianbo Shi. A random walks view of spectral segmentation. In Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS), 2001.
3. A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems (NIPS),14, 2001.
4. M. Meila. The Multicut Lemma. In UW Statistics Technical Report 417, 2001.
5. A. Azran and Z. Ghahramani. A new Approach to Data Driven Clustering. In International Conference on Machine Learning (ICML),11, 2006.
6. A. Azran and Z. Ghahramani. Spectral Methods for Automatic Multiscale Data Clustering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.
7. Ulrike Von Luxburg. A Tutorial on Spectral Clustering. Max Planck Institute for Biological Cybernetics Technical Report No. TR-149.
8. L. Zelnik-Manor and P. Perona. Self-Tuning Spectral Clustering. NIPS 2004.