Analysis of Social Media

MLD 10-802, LTI 11-772

William Cohen

10-2-2012

Announcements

- Next two papers summaries are due today
- Next Tuesday (10/9): initial project proposal
- Following Tuesday (10/16): final project proposal
- Mid-term project report (11/13)
- Final project: conference-paper like report due (12/10), and a presentation

Non-Survey Project Proposal

- On Tues 10/9 answer most of these questions:
  - What's the team?
    - Groups of 2-3 are recommended
    - Ask if your proposed group is 1 or 4
  - What's the data you'll work with?
    - Mid-term report is due week of 11/13
    - You should report then on the dataset and baseline results
  - What's the task or tasks?
  - How will you evaluate? (qualitative or quantitative?)
  - Is there a baseline method you hope to beat?
  - What are the key technical challenges, and what do you hope to learn?

Non-Survey Project Proposal

- On Tues 10/16 answer **all** of these questions:
  - What's the team?
    - Groups of 2-3 are recommended
    - Ask if your proposed group is 1 or 3
  - What's the data you'll work with?
    - Mid-term report is due week of 11/13
    - You should report then on the dataset and baseline results
  - What's the task or tasks?
  - How will you evaluate?
  - Is there a baseline method you hope to beat?
  - **Also address any feedback we give you on round 1**

Non-Survey Project Guidelines

- Sample projects are posted on the wiki
- Some examples:
  - reproduce the experiments in a paper, or do some more (compare X's method to Y's)
  - apply some off-the-shelf method to a new dataset (maybe solve some new task)
  - re-implement a published method and test on a dataset
  - write a survey of a subfield (new option)

Non-Survey Project Guidelines

- Usually there's at least one *dataset*, at least one *task* (with an evaluation metric), at at least one *method* you plan to use
  - These are things to talk about
  - It's ok to not know all the answers yet
- Collecting your own data is discouraged
  - it takes more time than you think
- There's *always* related work
  - similar/same task
  - same dataset
  - similar/same method
  - think about "k-NN with fixed k, not fixed distance"

Lit Survey Project Guidelines

- Final project is
  - lots of wikified paper summarize + study plans
  - set of dimensions along which papers in this area can be compared, and results of that comparison
  - a well-written survey of the key papers in the area expressed in these terms
- Initial (final) proposal has most (all) of:
  - Team
  - Working definition of the area, and some representative papers
  - Tentative section headings for the final writeup

Sample “dimensions”

Model	Links	Documents
LDA	-	words
Link LDA	-	words + entities
Relational Topic model	document-document	words + document ids
Pairwise Link-LDA, Link-PLSA-LDA	document-document	words + cited document ids
Copycat, Citation Influence models	document-document	words + cited document ids
Latent Topic Hypertext model	document-document	words + cited document ids
Author Recipient Topic model	-	docs + authors + recipients
Author Topic model	-	docs + authors
Topic Link LDA	document-document	words + authors
MMSB	entity-entity	-
Sparse block model (Parkkinen et al.)	entity-entity	-
Nubbi	entity-entity	words near entities or entity-pairs
Group topic model	entity-entity	words about the entity-entity event
Block-LDA	entity-entity	words + entities

Table 1: Related work

This is the rough idea but might be modified - eg maybe papers fall into 2-3 clusters in your subarea…..

Sample “dimensions”

Link LDA and many other extensions to LDA model documents that are annotated with metadata. In a parallel area of research, various different approaches to modeling links between documents have been explored. For instance, Pairwise-Link-LDA[6] combines MMSB with LDA by modeling documents using LDA and generating links between them using MMSB. The Relational Topic Model [7] generates links between documents based on their topic distributions. The Copycat and Citation Influence models [8] also model links between citing and cited documents by extending LDA and eliminating independence between documents. The Latent Topic Hypertext Model (LTHM) [9] presents a generative process for documents that can be linked to each other from specific words in the citing document. The model proposed in this paper, Block-LDA, is different from this class of models in that they model links

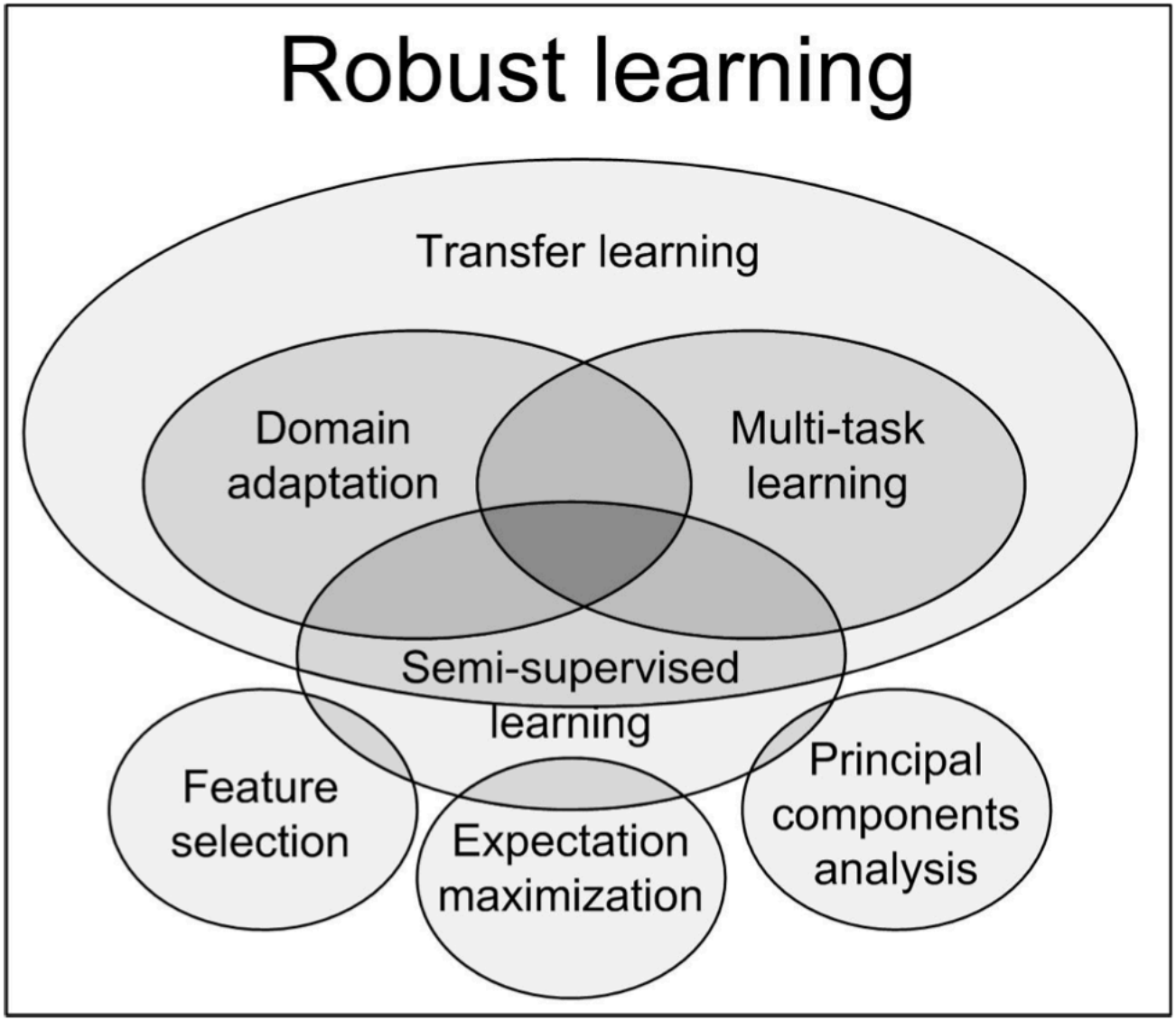


Figure 2.1: Venn diagram representation of the subspace of robust learning settings. Domain adaptation and multi-task learning are represented as subsets of transfer learning, which is itself a subset of all robust learning techniques. These techniques can also intersect with semi-supervised methods. A sampling of non-transfer robust learning techniques (such as sparse

Table 2.1: Learning settings are summarized by the type of auxiliary and test data used. For all settings we assume $(X_{train}^{source}, Y_{train}^{source})$ is available at training time, while Y_{test} is unknown. Settings for which we have run experiments are marked in bold (c.f. Table 2.4). Some settings are omitted where they do not correspond to a known natural example.

Natural name for learning setting	Auxiliary data		Test data	
	Domain	Labels	Domain	X_{test}
Inductive learning	-	-	$\mathcal{D}^{source}$	unseen
Semi-supervised inductive learning	$\mathcal{D}^{source}$	unseen	$\mathcal{D}^{source}$	unseen
Transductive learning	-	-	$\mathcal{D}^{source}$	seen
Transfer learning	-	-	$\mathcal{D}^{target}$	unseen
Inductive transfer learning	$\mathcal{D}^{target}$	seen	$\mathcal{D}^{target}$	unseen
Semi-supervised inductive transfer learning	$\mathcal{D}^{source}$	unseen	$\mathcal{D}^{target}$	unseen
Transductive transfer learning	-	-	$\mathcal{D}^{target}$	seen
Supervised Transductive transfer learning	$\mathcal{D}^{target}$	seen	$\mathcal{D}^{target}$	seen
Relaxed Transductive transfer learning[1]	-	-	$\mathcal{D}^{target}$	seen
Semi-supervised transductive transfer learning	$\mathcal{D}^{source}$	unseen	$\mathcal{D}^{target}$	seen

[1] A relaxation of transductive transfer learning in which proportions of labels in the target data is known at training time.

Lit Survey Project Guidelines

- Midterm report
  - set of dimensions along which papers in this area can be compared, and results of that comparison
  - complete bibliography of the area
- Expectations
  - Working in a team is good
  - Every team member should submit two wikified summaries per week

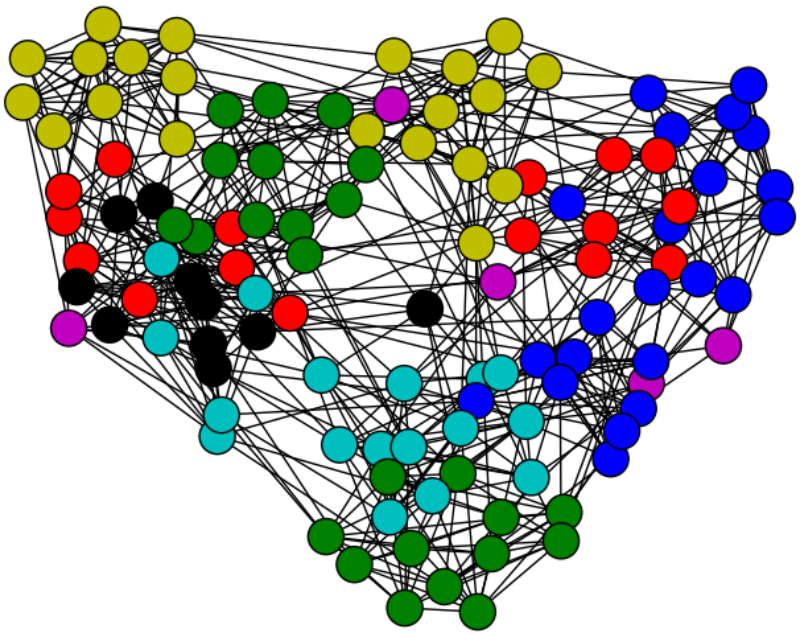
Spectral clustering methods

Motivation

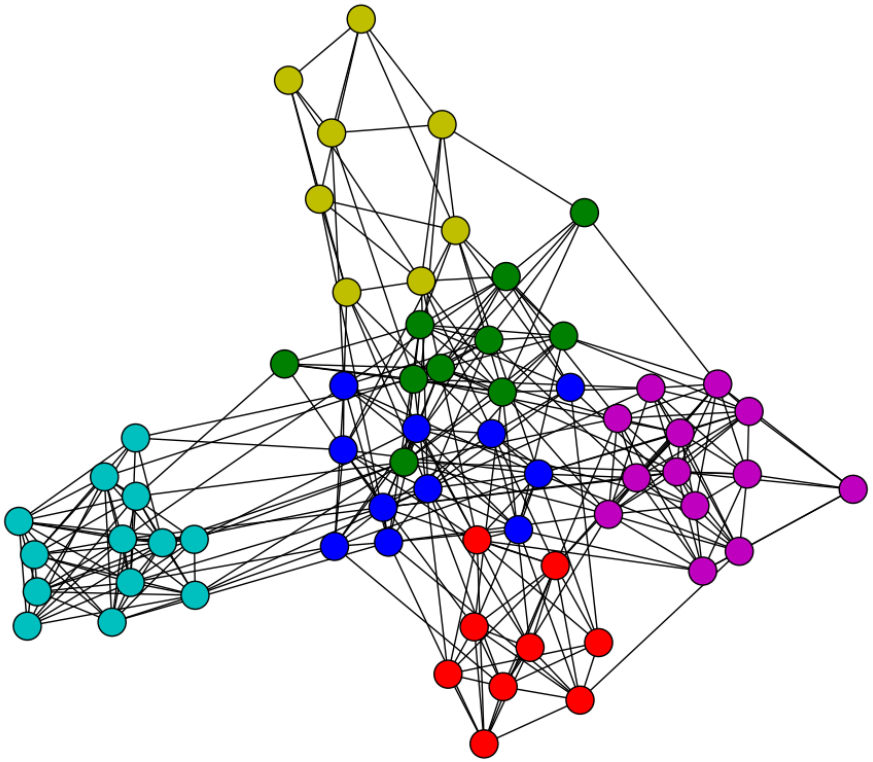
- Social graphs seem to have
  - some aspects of randomness
    - small diameter, giant connected components,..
  - some structure
    - homophily
    - small diameter
    - scale-free degree distribution
    - …
- Stochastic models for how graphs are constructed
  - Erdos-Renyi “random”, preferential-attachment, …

Stochastic Block Models

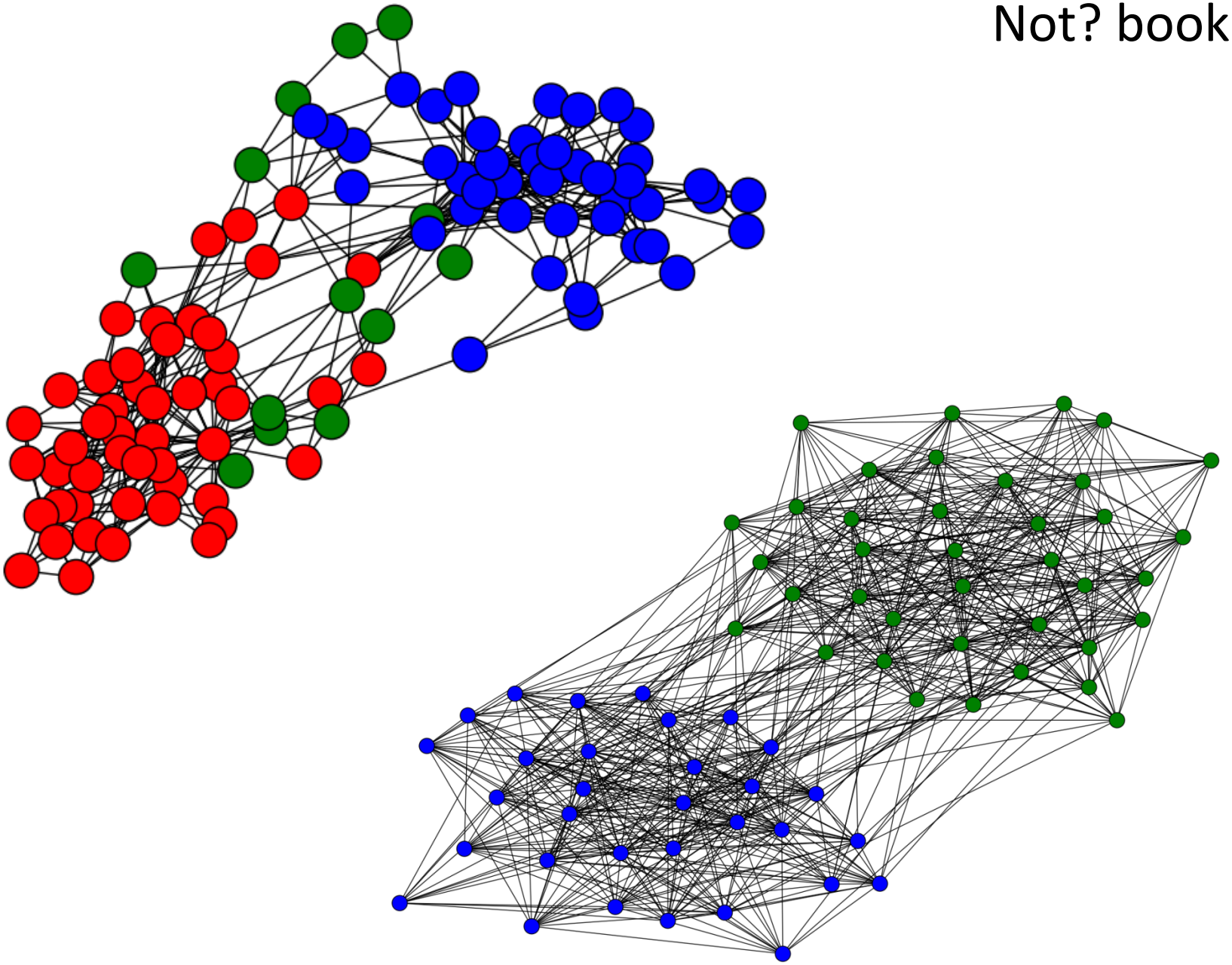
- "Stochastic block model", aka "Block-stochastic matrix":
  - Draw n_i nodes in block i
  - With probability p_{ij}, connect pairs (u,v) where u is in block i, v is in block j
  - Special, simple case: $p_{ii}=q_i$, and $pij=s$ for all $i \neq j$



Not? football



Not? books

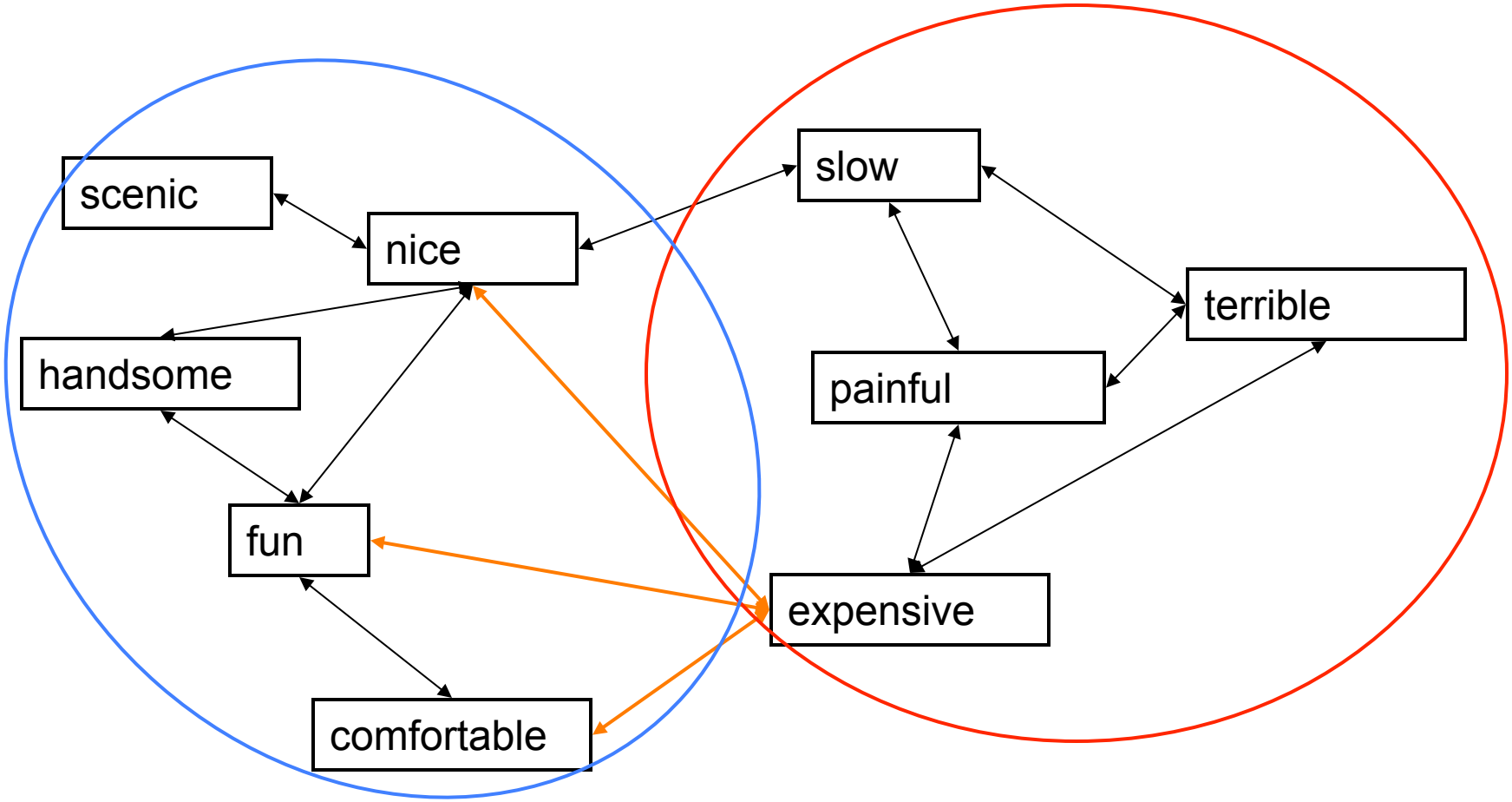


Stochastic Block Models

- "Stochastic block model", aka "Block-stochastic matrix":
  - Draw n_i nodes in block i
  - With probability p_{ij}, connect pairs (u,v) where u is in block i, v is in block j
  - Special, simple case: $p_{ii}=q_i$, and $p_{ij}=s$ for all $i \neq j$
- Inferring a model
  - determine block $i(u)$ for each node u
  - p_{ij}'s are now easy to estimate
  - how?

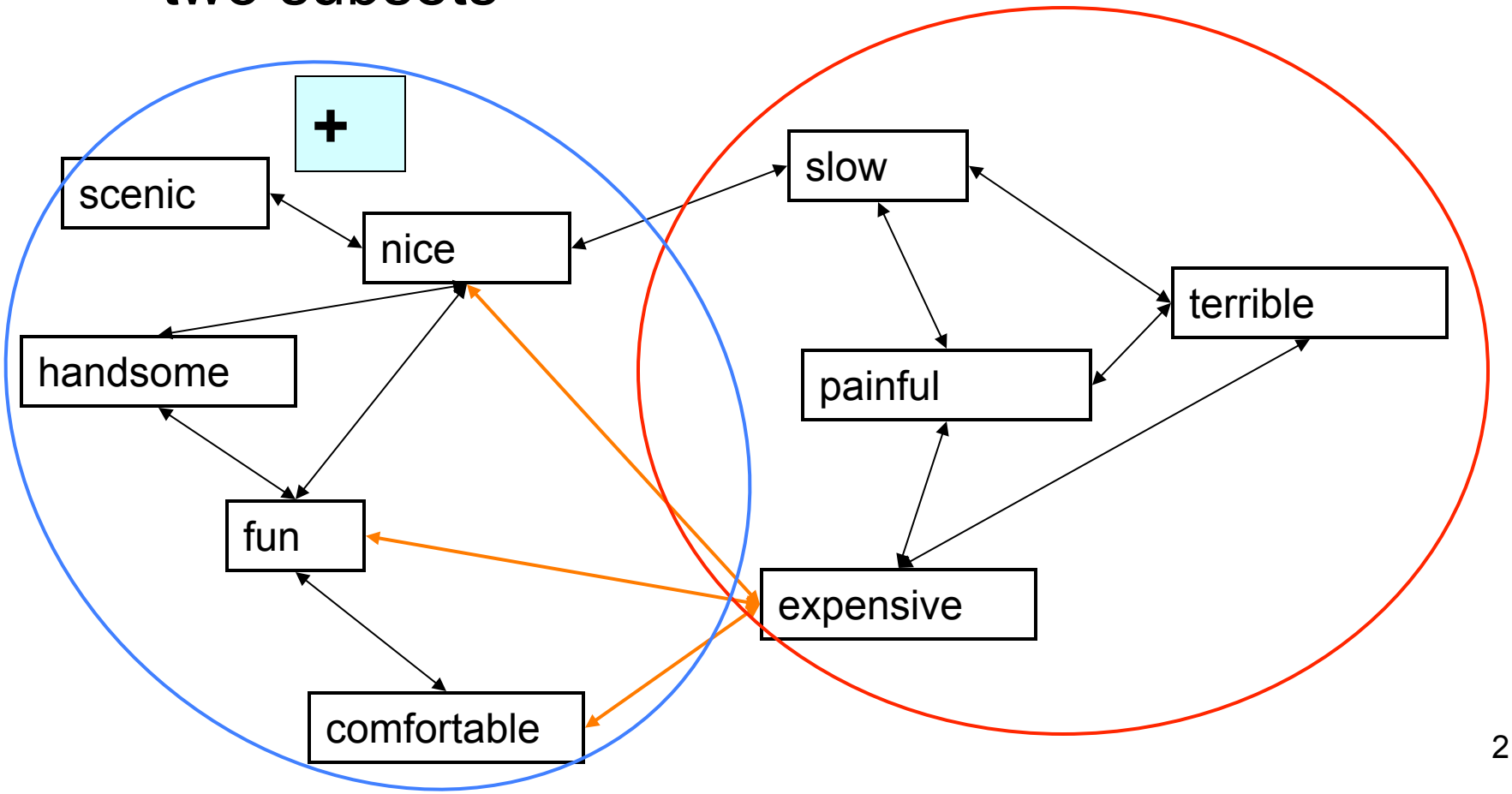
We’ve seen this problem before…

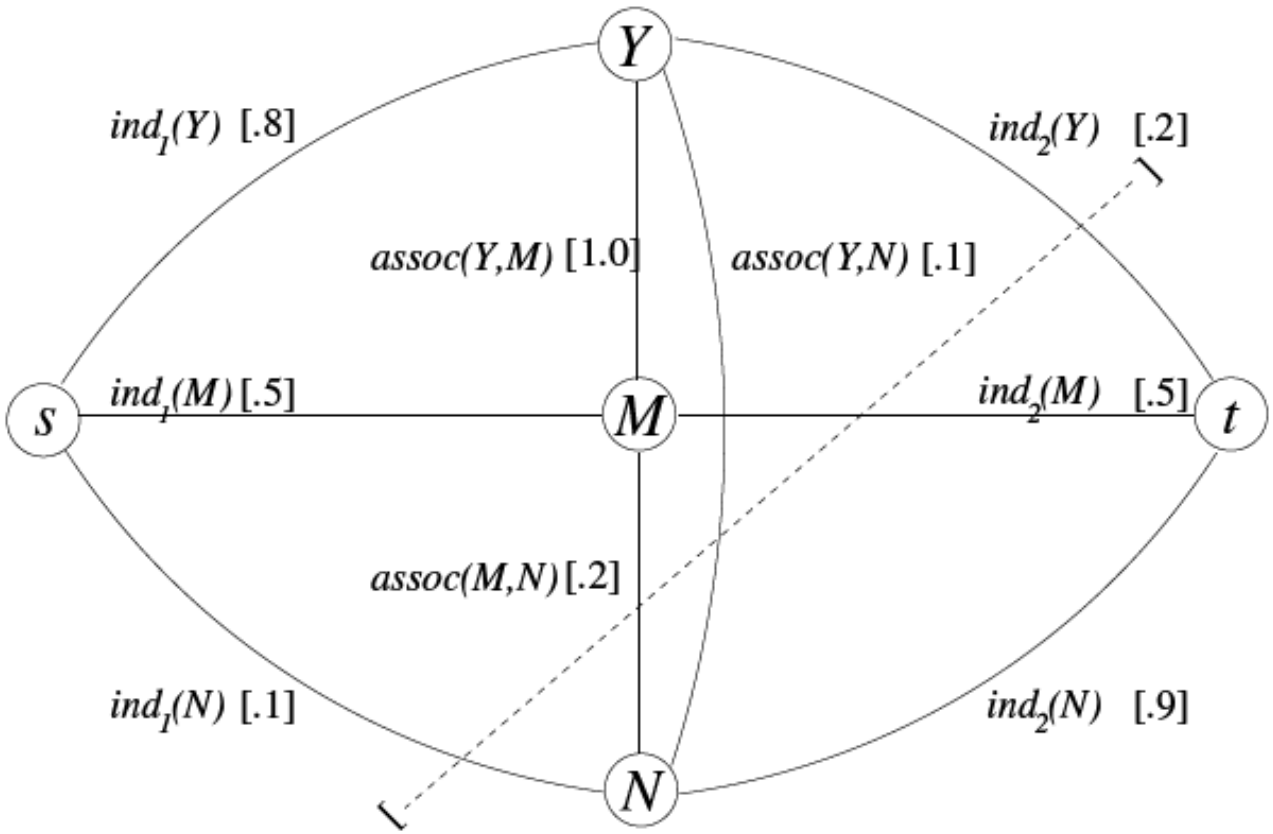
<aside>


scenic
nice
handsome
fun
comfortable
slow
terrible
painful
expensive

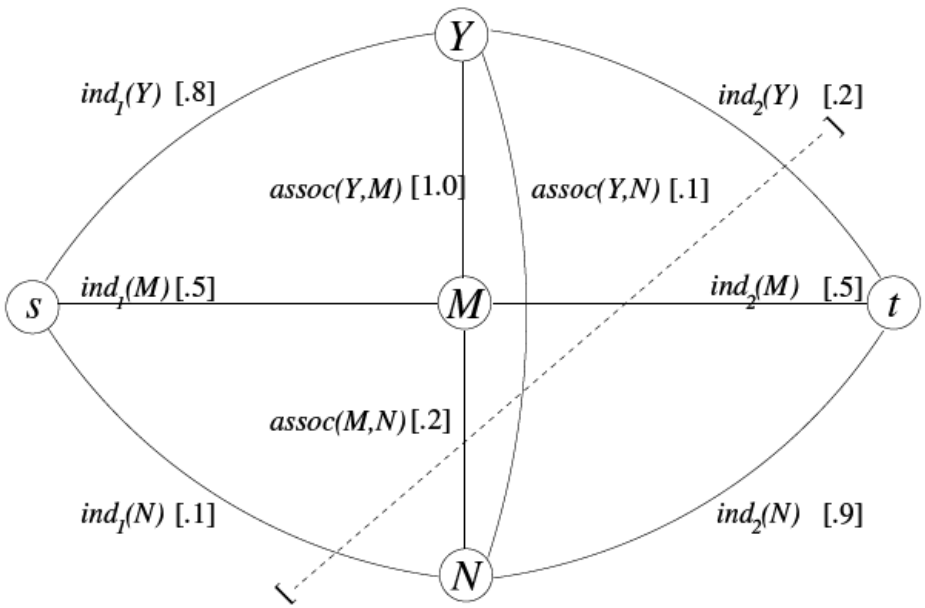
Hatzivassiloglou & McKeown 1997

4. A clustering algorithm partitions the adjectives into two subsets




Y
ind1(Y) [.8]
ind2(Y) [.2]
assoc(Y,M) [1.0]
assoc(Y,N) [.1]
s
ind1(M) [.5]
M
ind2(M) [.5]
t
assoc(M,N) [.2]
ind1(N) [.1]
ind2(N) [.9]
N

Pang & Lee EMNLP 2004



C_1	Individual penalties	Association penalties	Cost
{Y,M}	$.2 + .5 + .1$	$.1 + .2$	1.1
(none)	$.8 + .5 + .1$	0	1.4
{Y,M,N}	$.2 + .5 + .9$	0	1.6
{Y}	$.2 + .5 + .1$	$1.0 + .1$	1.9
{N}	$.8 + .5 + .9$	$.1 + .2$	2.5
{M}	$.8 + .5 + .1$	$1.0 + .2$	2.6
{Y,N}	$.2 + .5 + .9$	$1.0 + .2$	2.8
{M,N}	$.8 + .5 + .9$	$1.0 + .1$	3.3

Figure 2: Graph for classifying three items. Brackets enclose example values; here, the individual scores happen to be probabilities. Based on *individual* scores alone, we would put Y ("yes") in C_1, N ("no") in C_2, and be undecided about M ("maybe"). But the *association* scores favor cuts that put Y and M in the same class, as shown in the table. Thus, the minimum cut, indicated by the dashed line, places M together with Y in C_1.

</aside>

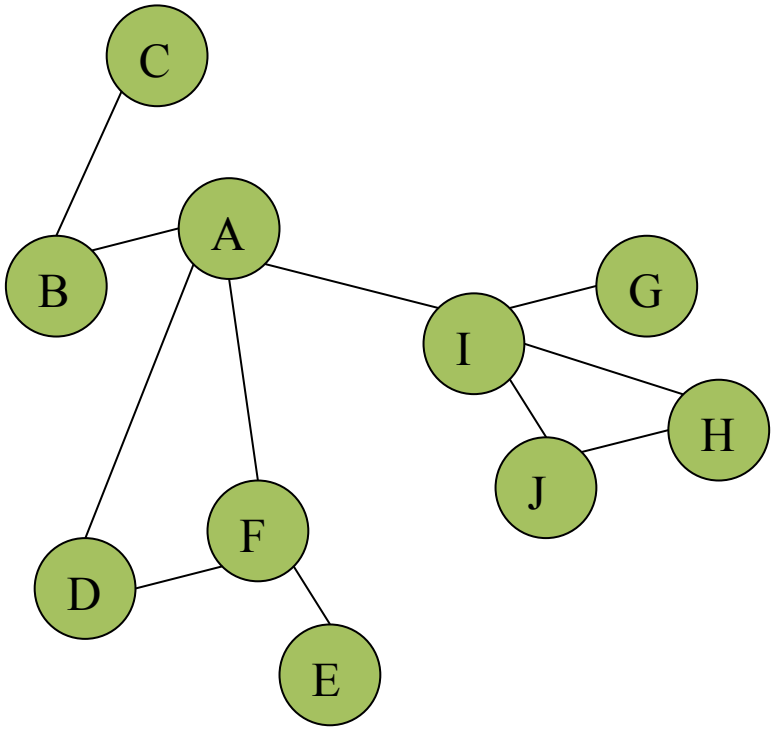
Stochastic Block Models

- "Stochastic block model"
- Inferring a model
  - determine block *i(u)* for each node *u*
  - p_{ij}'s are now easy to estimate
  - how?
    1. probabilistic modeling (later)
    2. spectral clustering
       - assumes i**ntra-block** connectivity higher than **inter-block** connectivity

Spectral clustering

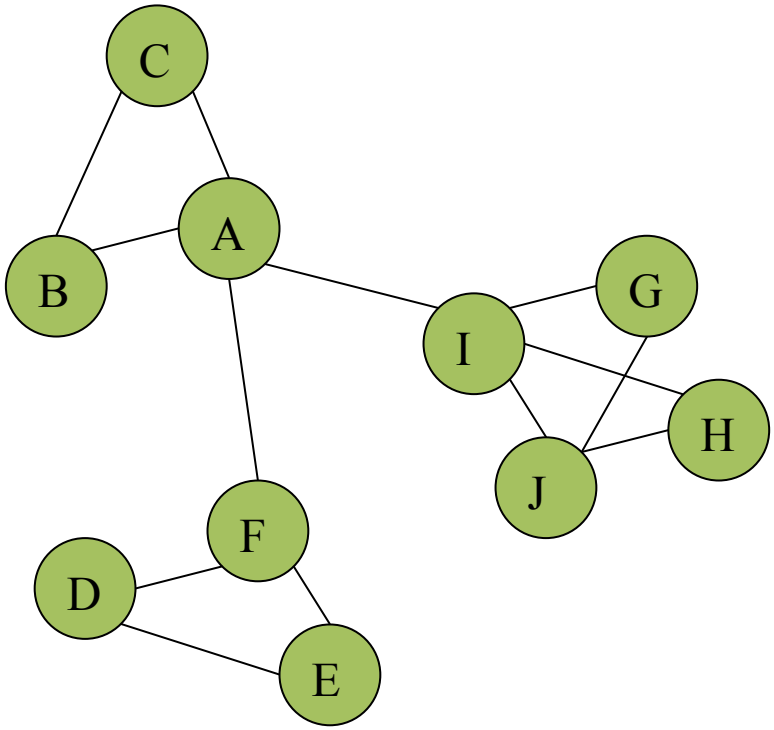
Spectral Clustering: Graph = Matrix

	A	B	C	D	E	F	G	H	I	J
A		1		1		1				
B	1		1							
C		1								
D	1					1				
E						1				
F	1			1	1					
G									1	
H							1		1	1
I							1	1		1
J								1	1	



Spectral Clustering: Graph = Matrix
Transitively Closed Components = “Blocks”

	A	B	C	D	E	F	G	H	I	J
A	_	1	1			1				
B	1	_	1							
C	1	1	_							
D				_	1	1				
E				1	_	1				
F	1			1	1	_				
G							_		1	1
H								_	1	1
I							1	1	_	1
J							1	1	1	_



Of course we can’t see the “blocks” unless the nodes are sorted by cluster…

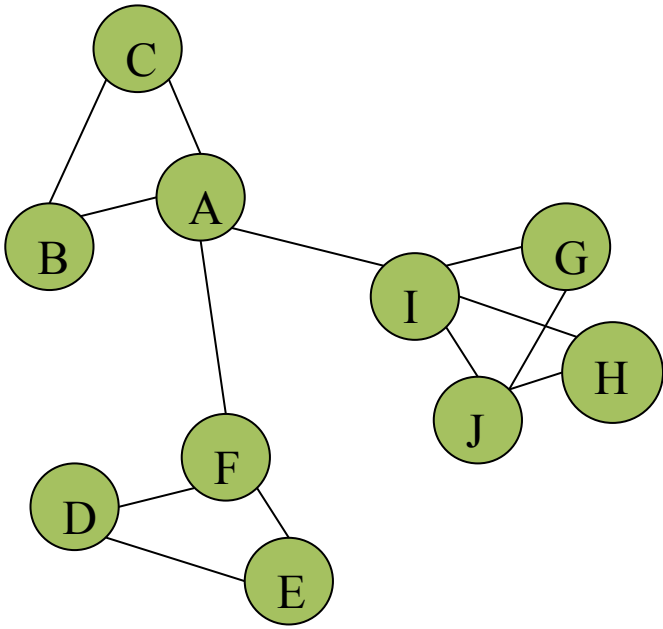
Spectral Clustering: Graph = Matrix
Vector = Node → Weight

M

	A	B	C	D	E	F	G	H	I	J
A	_	1	1			1				
B	1	_	1							
C	1	1	_							
D				_	1	1				
E				1	_	1				
F	1			1	1	_				
G							_		1	1
H								_	1	1
I							1	1	_	1
J							1	1	1	_

v

	A
A	3
B	2
C	3
D	
E	
F	
G	
H	
I	
J	



M

Spectral Clustering: Graph = Matrix

$M*v_1 = v_2$ "propogates weights from neighbors"

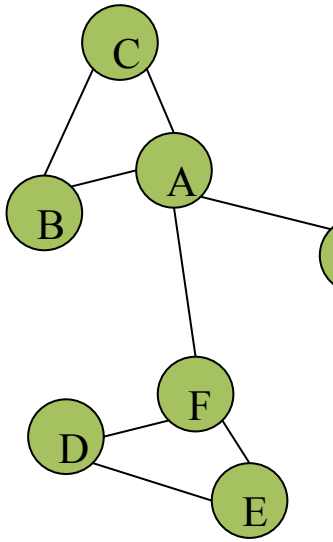
M * v_1 = v_2

	A	B	C	D	E	F	G	H	I	J
A	_	1	1			1				
B	1	_	1							
C	1	1	_							
D				_	1	1				
E				1	_	1				
F				1	1	_				
G							_		1	1
H								_	1	1
I							1	1	_	1
J							1	1	1	_

M

A	3
B	2
C	3
D	
E	
F	
G	
H	
I	
J	

A	2*1+3*1+0*1
B	3*1+3*1
C	3*1+2*1
D	
E	
F	
G	
H	
I	
J	



Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

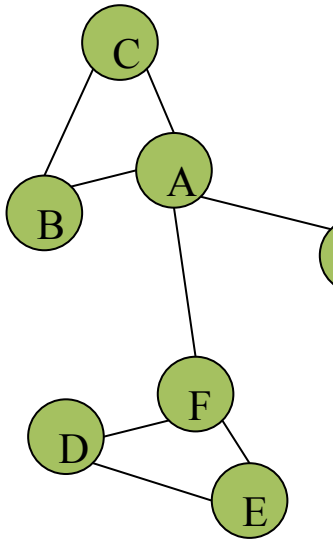
W: normalized so columns sum to 1

W * v_1 = v_2

	A	B	C	D	E	F	G	H	I	J
A	_	.5	.5			.3				
B	.3	_	.5							
C	.3	.5	_							
D				_	.5	.3				
E				.5	_	.3				
F	.3			.5	.5	_				
G							_		.3	.3
H								_	.3	.3
I							.5	.5	_	.3
J							.5	.5	.3	_

A	3
B	2
C	3
D	
E	
F	
G	
H	
I	
J	

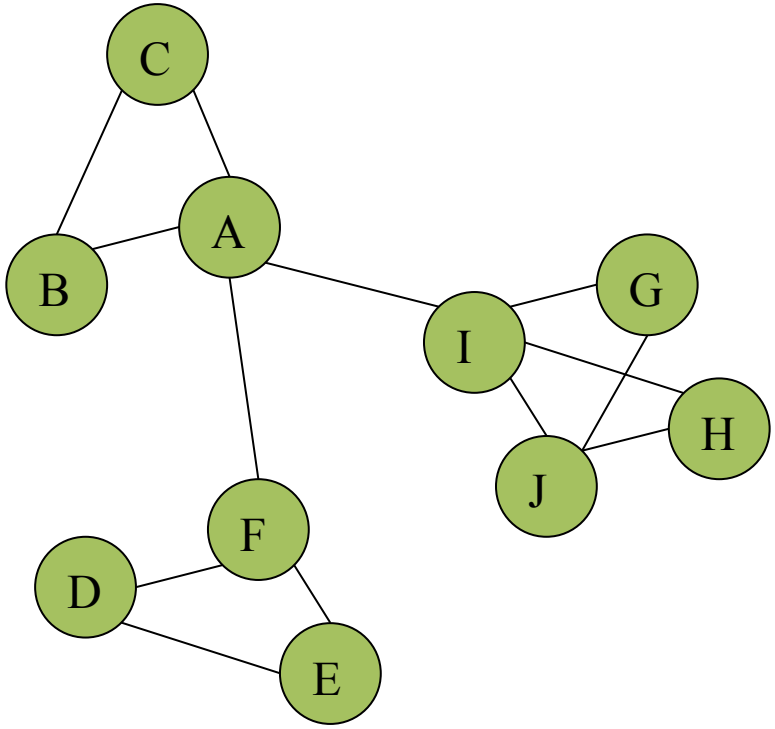
A	2*.5+3*.5+0*.3
B	3*.3+3*.5
C	3*.33+2*.5
D	
E	
F	
G	
H	
I	
J	



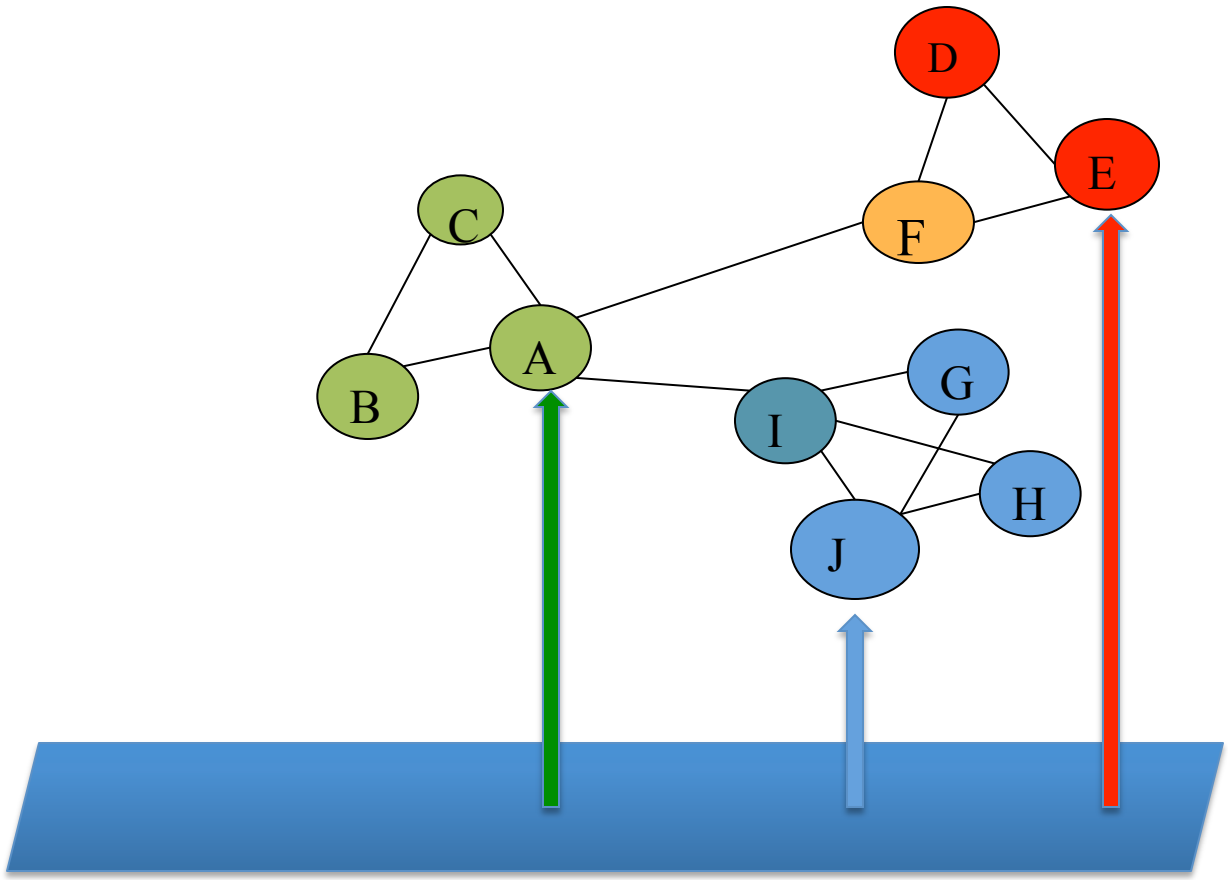
How Matrix Notation Helps

- Suppose every node has a value (IQ, income,..) *y(i)*
  - Each node *i* has value y_i …
    - and neighbors *N(i),* degree d_i
  - If *i,j* connected then *j* exerts a force $-K*[y_i-y_j]$ on *i*
  - Total*:* $$F_i = \sum_{j \in N(i)} \left(-K(y_i - y_j) \right) = -K \left(d_i y_i - \sum_{j \in N(i)} y_j \right)$$
  - Matrix notation: **F** = -K(D-A)**y**
    - D is *degree matrix:* $D(i,i)=d_i$ and 0 for $i \neq j$
    - *A* is *adjacency matrix: A(i,j)=1* if *i,j* connected and 0 else
  - Interesting (?) goal: set **y** so (D-A)**y** = *c****y**

Blocks and Forces



Blocks and Forces



How Matrix Notation Helps

- Suppose every node has a value (IQ, income,..) *y(i)*
  - Matrix notation: **F** = -K(D-A)**y**
    - D is *degree matrix:* $D(i,i)=d_i$ and 0 for $i \neq j$
    - *A* is *adjacency matrix: A(i,j)=1* if *i,j* connected and 0 else
  - Interesting (?) goal: set **y** so (D-A)**y** = *c****y**
    - D-A is the *graph Laplacian* and comes up a lot
  - Picture: neighbors pull *i* up or down, but net force doesn't change *relative* positions of nodes

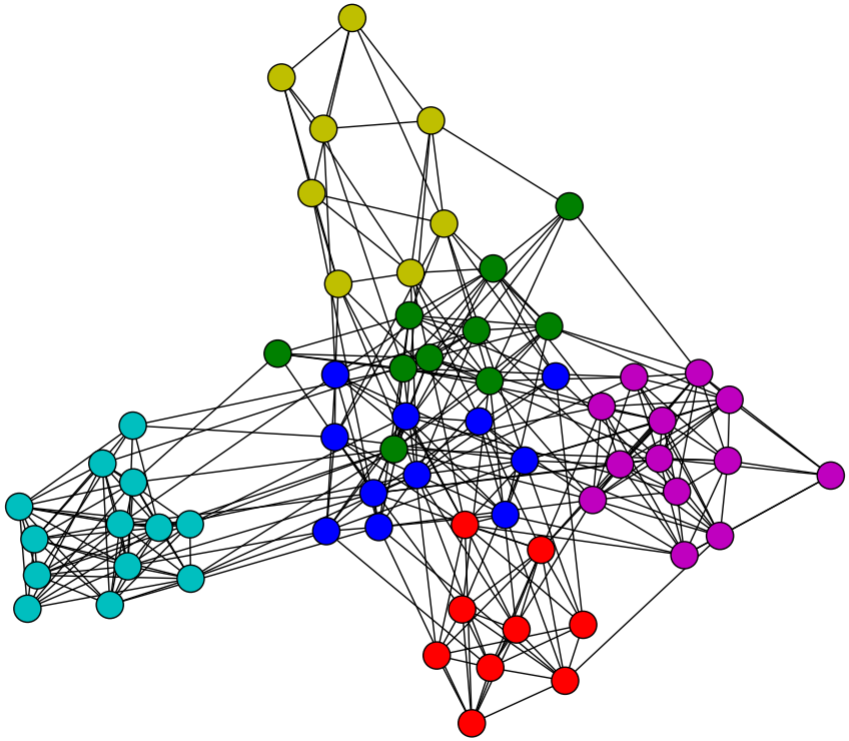
Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

- smallest eigenvecs of D-A are largest eigenvecs of A
- smallest eigenvecs of I-W are largest eigenvecs of W

Q: How do I pick **v** to be an eigenvector for a block-stochastic matrix?

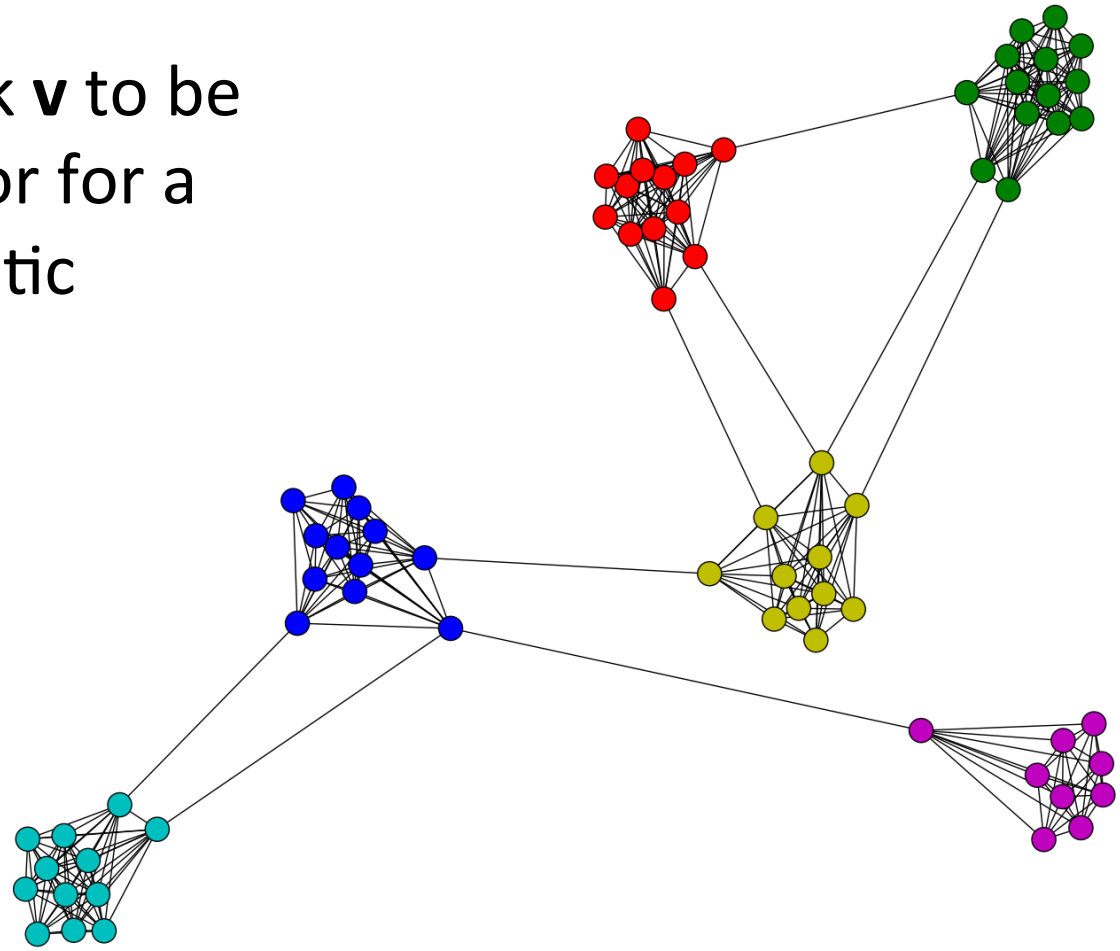


Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

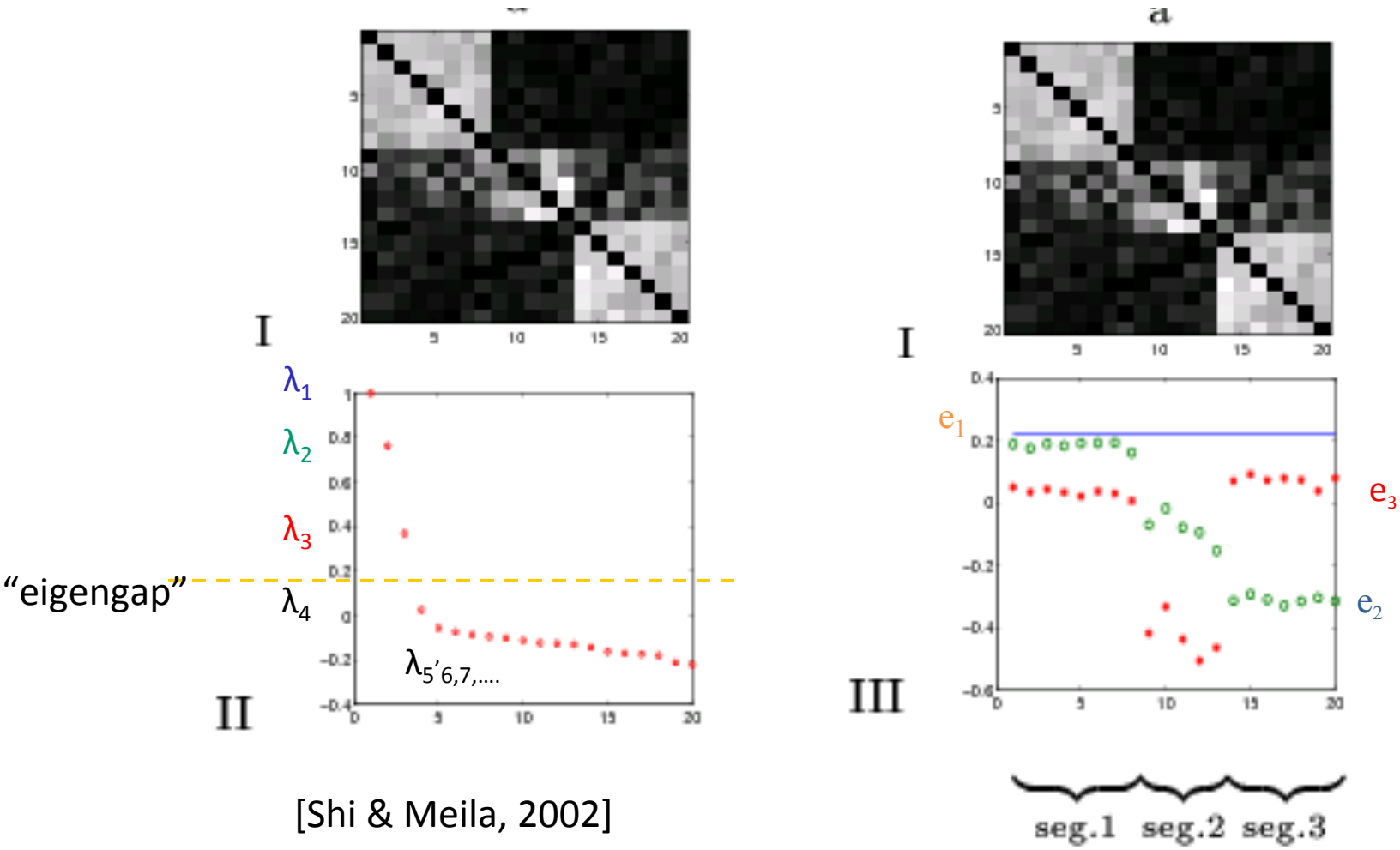
How do I pick **v** to be an eigenvector for a block-stochastic matrix?



Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

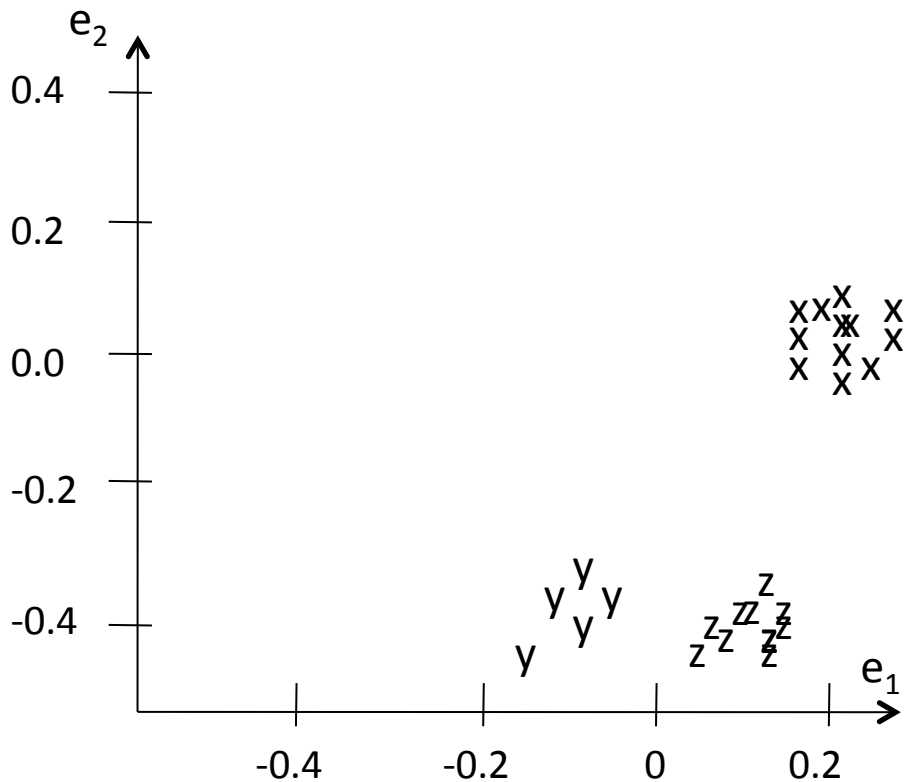
$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

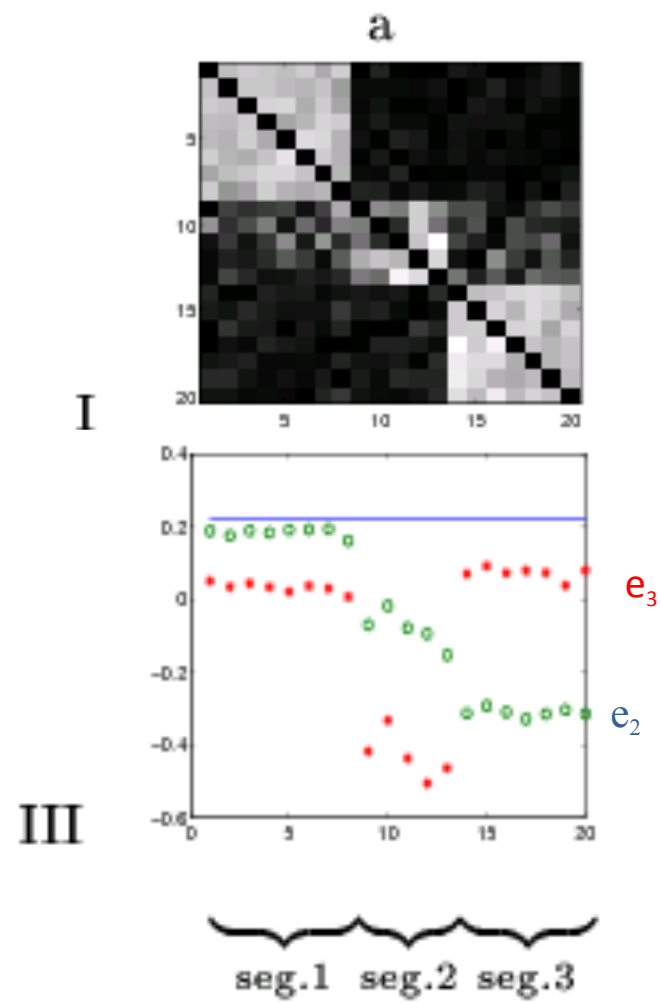


[Shi & Meila, 2002]

Spectral Clustering: Graph = Matrix

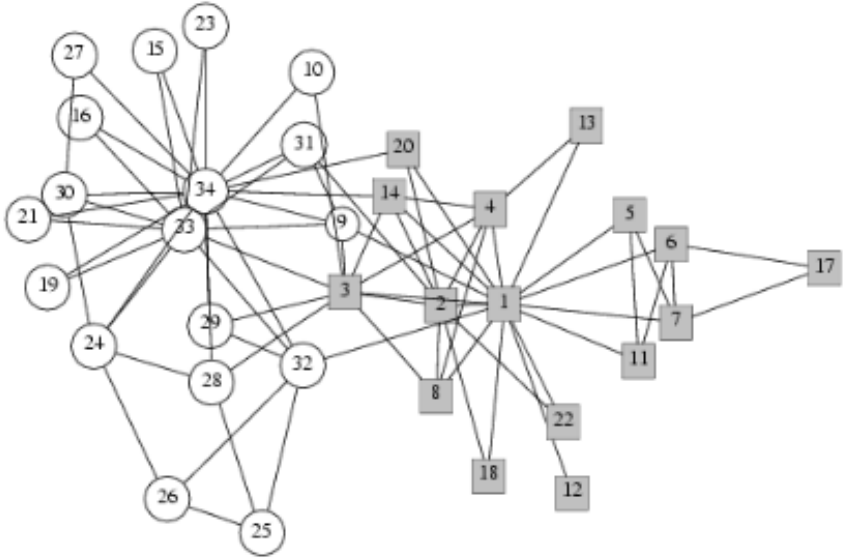
$W*v_1 = v_2$ "propagates weights from neighbors"

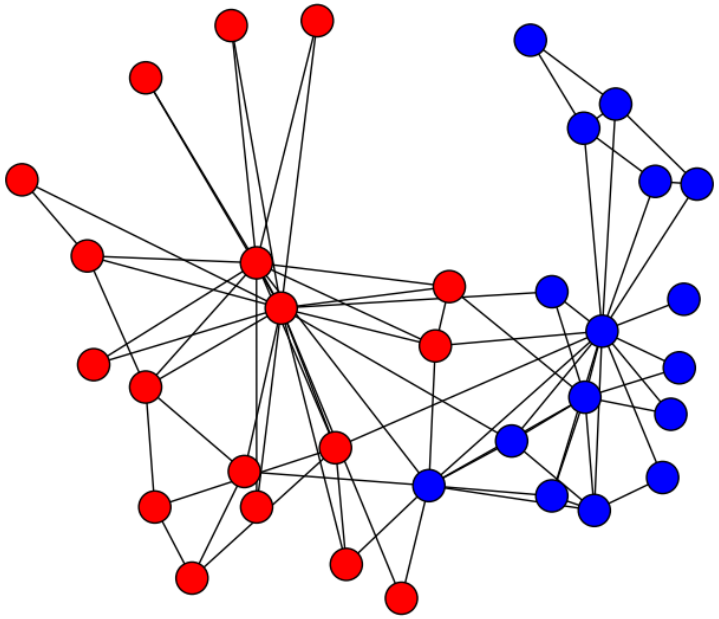


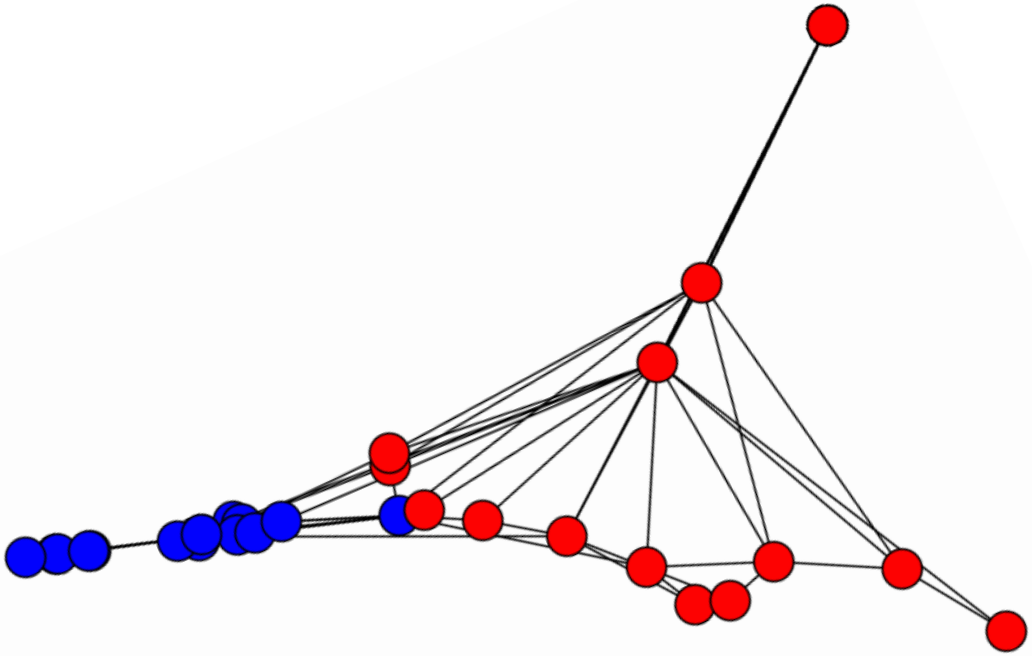


[Shi & Meila, 2002]

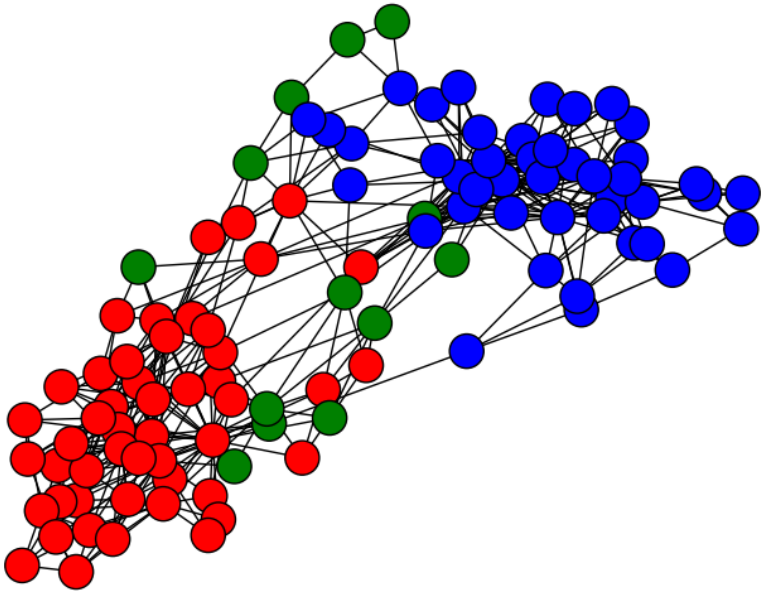
M

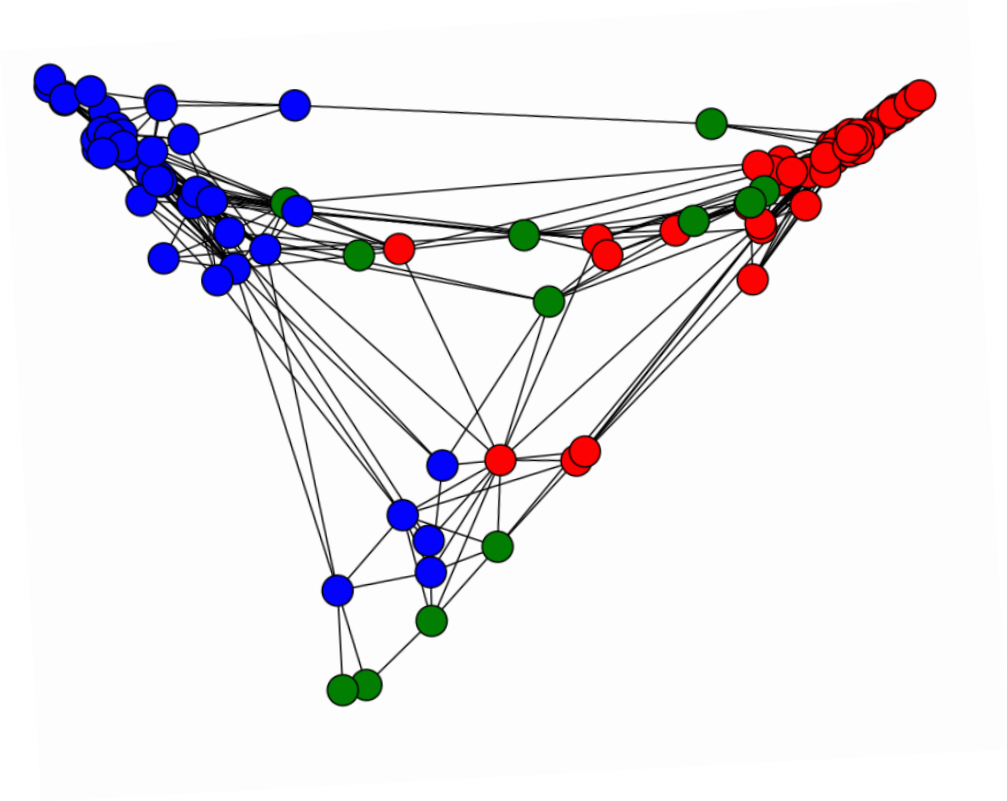

27
15
23
10
16
31
20
13
30
34
14
4
21
33
19
5
6
17
19
3
2
1
7
29
24
32
11
28
8
22
18
12
26
25



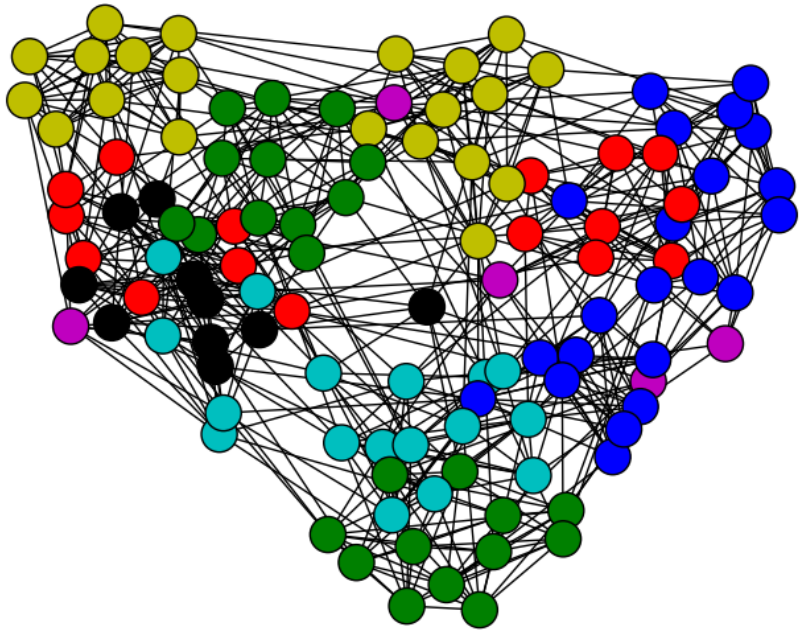


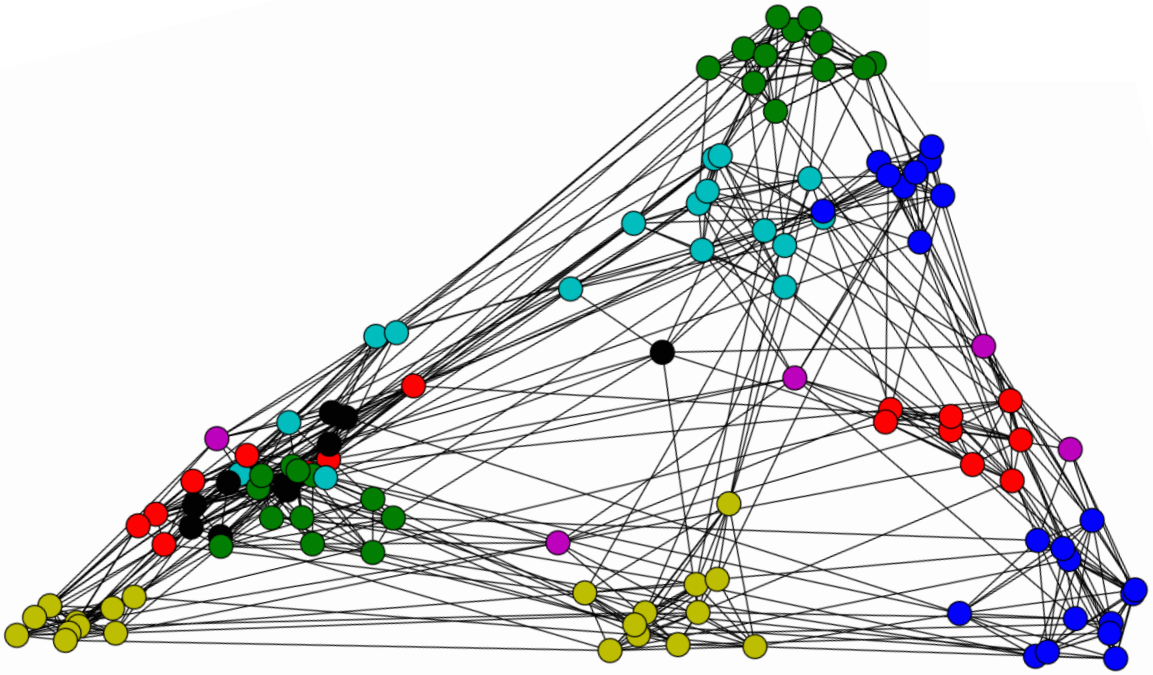
Books



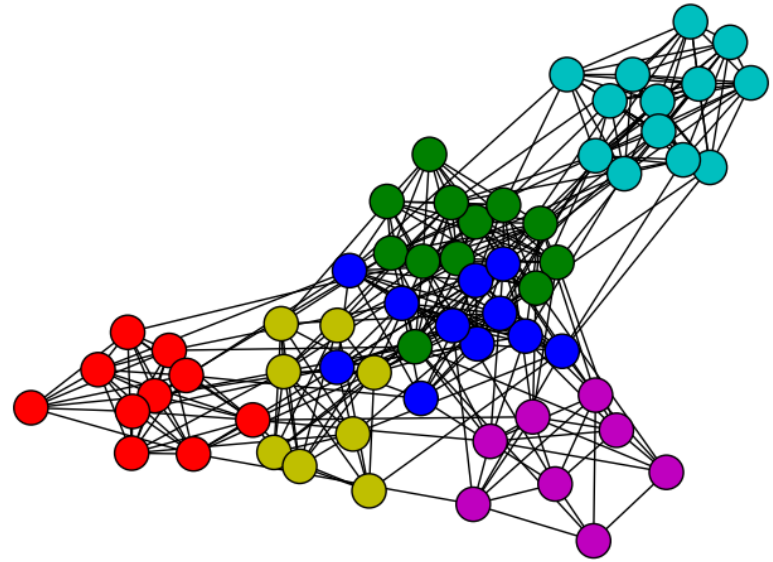


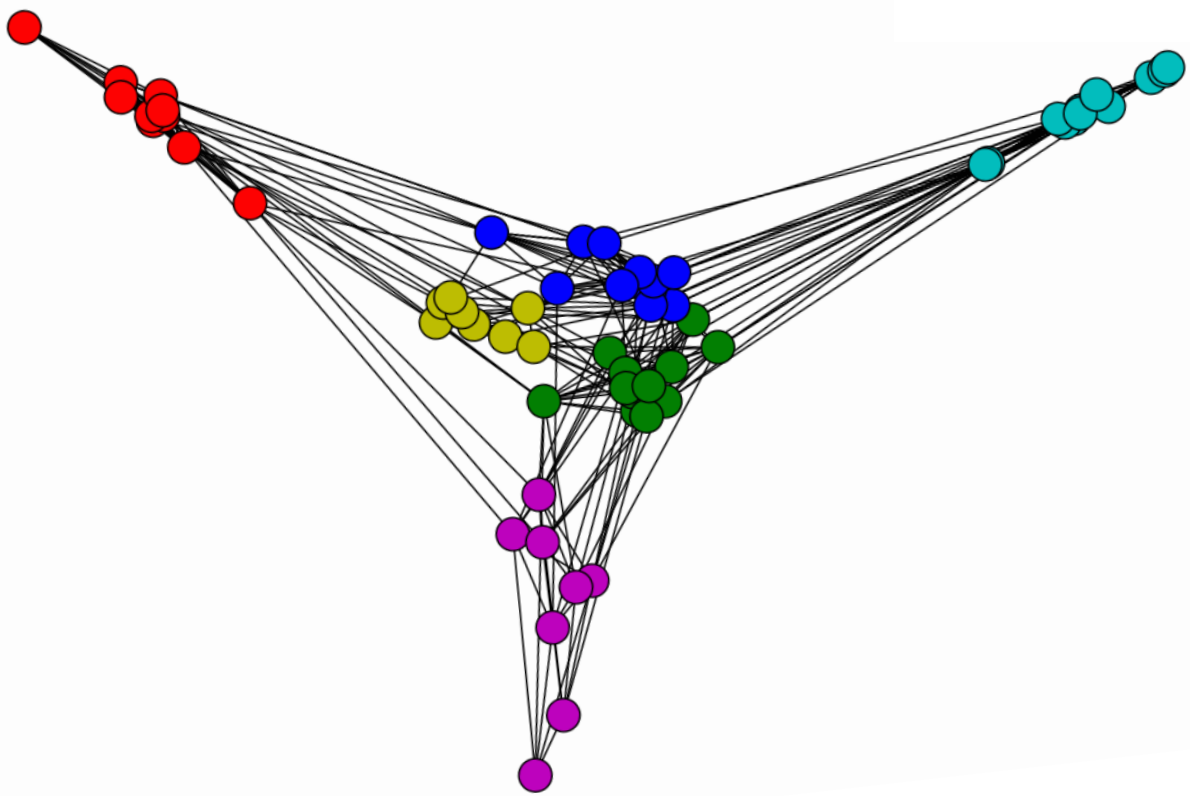
Football



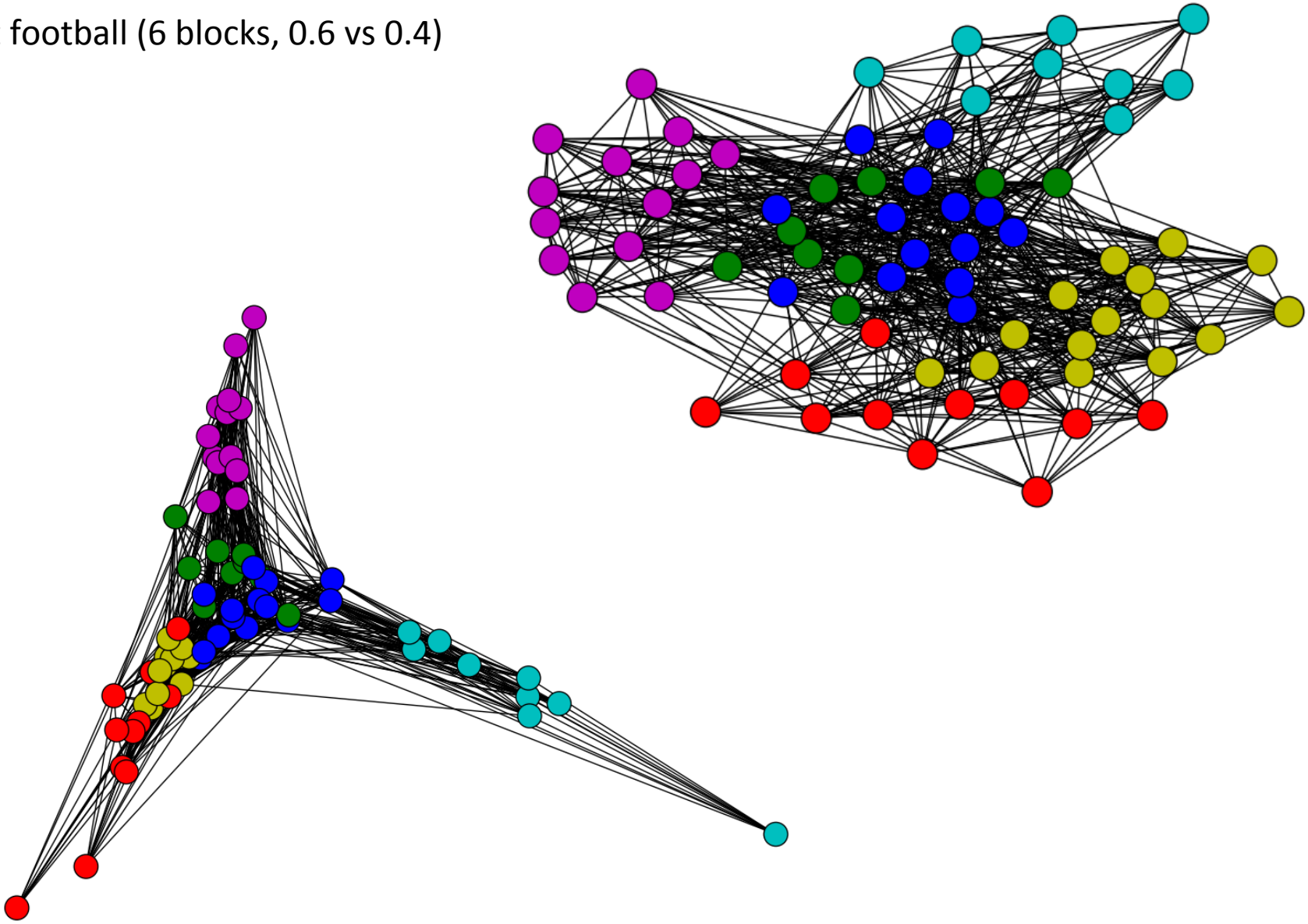


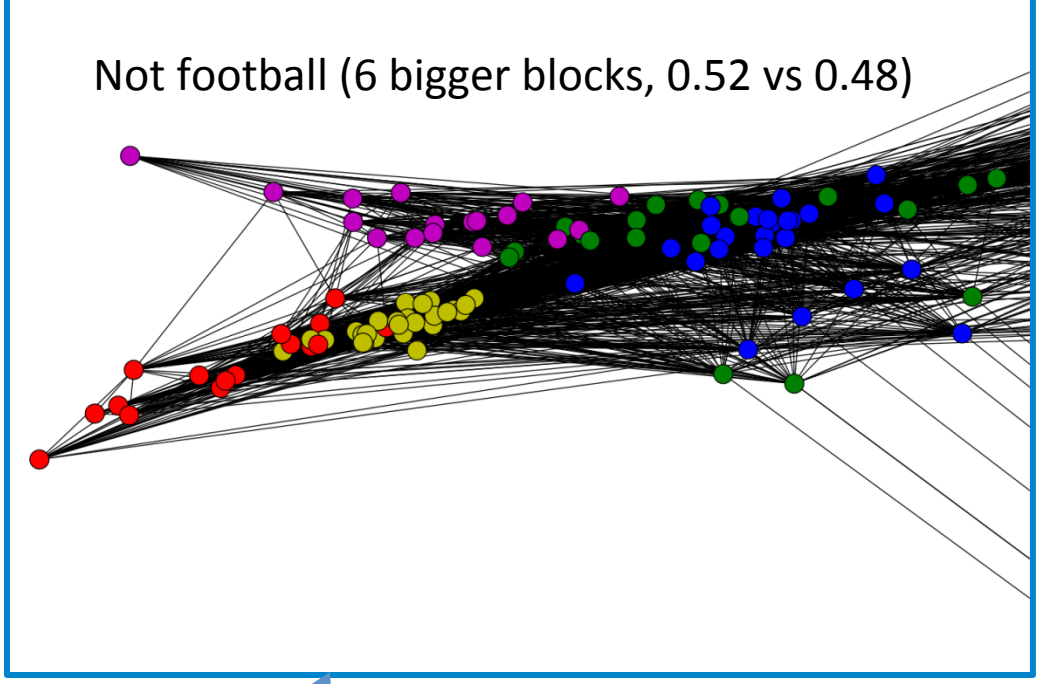
Not football (6 blocks, 0.8 vs 0.1)

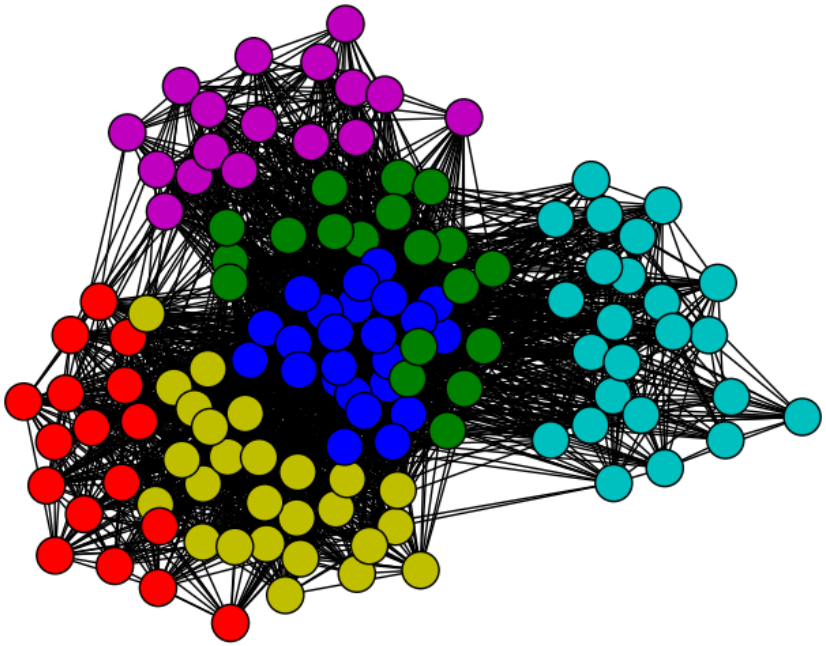


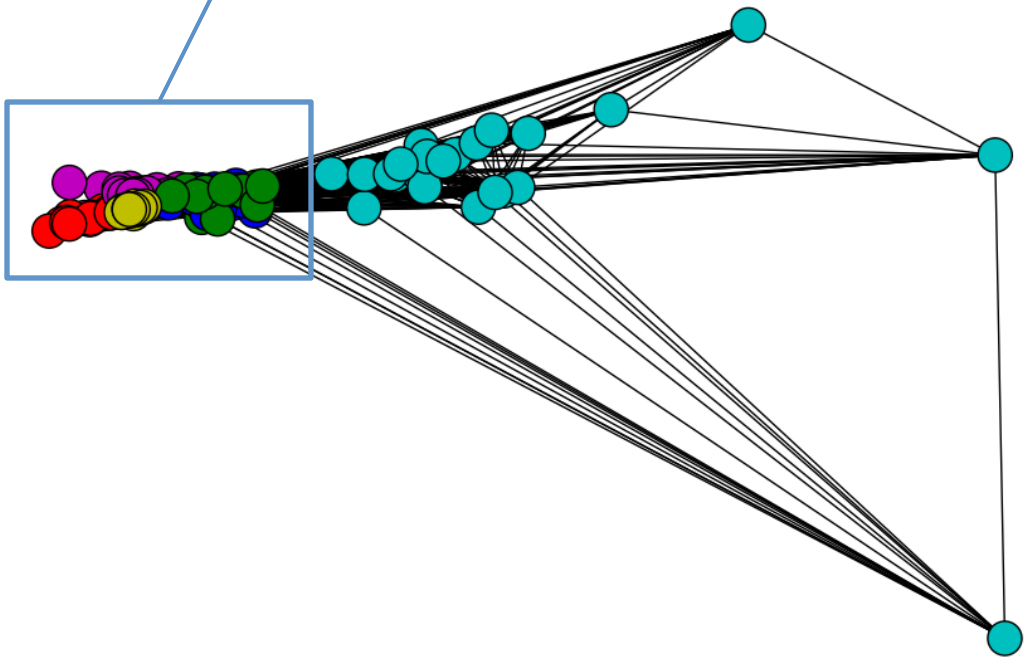


Not football (6 blocks, 0.6 vs 0.4)




Not football (6 bigger blocks, 0.52 vs 0.48)





Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

- smallest eigenvecs of D-A are largest eigenvecs of A
- smallest eigenvecs of I-W are largest eigenvecs of W

Suppose each *y(i)=+1* or *-1*:

- Then *y* is a cluster indicator that splits the nodes into two
- what is $\mathbf{y}^T(D-A)\mathbf{y}$?

$$\mathbf{y}^T(D-A)\mathbf{y} = \mathbf{y}^T D\mathbf{y} - \mathbf{y}^T A\mathbf{y} = \sum_i d_i y_i^{\,2} - \sum_{i,j} a_{i,j} y_i y_j$$

$$= \frac{1}{2}\left[2\sum_i d_i y_i^{\,2} - 2\sum_{i,j} a_{i,j} y_i y_j\right]$$

$$= \frac{1}{2}\left[\sum_i \left(\sum_j a_{ij}\right) y_i^{\,2} + \sum_j \left(\sum_i a_{ij}\right) y_j^{\,2} - 2\sum_{i,j} a_{i,j} y_i y_j\right]$$

$$= \frac{1}{2}\left[\sum_{i,j} a_{ij} y_i^{\,2} + \sum_{i,j} a_{ij} y_j^{\,2} - 2\sum_{i,j} a_{i,j} y_i y_j\right]$$

$$= \frac{1}{2}\left[\sum_{i,j} a_{i,j}(y_i - y_j)^2\right] \qquad = \text{size of CUT}(\mathbf{y})$$

$$\mathbf{y}^T(I-W)\mathbf{y} = \text{size of NCUT}(\mathbf{y})$$

NCUT: roughly minimize ratio of transitions between classes vs transitions within classes

Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

- smallest eigenvecs of D-A are largest eigenvecs of A
- smallest eigenvecs of I-W are largest eigenvecs of W

Suppose each *y(i)=+1* or *-1*:

- Then *y* is a cluster indicator that cuts the nodes into two
- what is $\mathbf{y}^T(D-A)\mathbf{y}$? <u>The cost of the graph cut defined by **y**</u>
- what is $\mathbf{y}^T(I-W)\mathbf{y}$? <u>Also a cost of a graph cut defined by **y**</u>
- How do minimize it?
  - Turns out: to minimize $\mathbf{y}^T X \mathbf{y} / (\mathbf{y}^T\mathbf{y})$ find *smallest* eigenvector of X
  - But: this will not be +1/-1, so it's a "relaxed" solution

Some more terms

- If A is an adjacency matrix (maybe weighted) and D is a (diagonal) matrix giving the degree of each node
  - Then D-A is the *(unnormalized) Laplacian*
  - $W=AD^{-1}$ is a *probabilistic adjacency matrix*
  - I-W is the *(normalized or random-walk) Laplacian*
  - etc….
- The largest eigenvectors of W correspond to the smallest eigenvectors of I-W
  - So sometimes people talk about "*bottom eigenvectors of the Laplacian*"

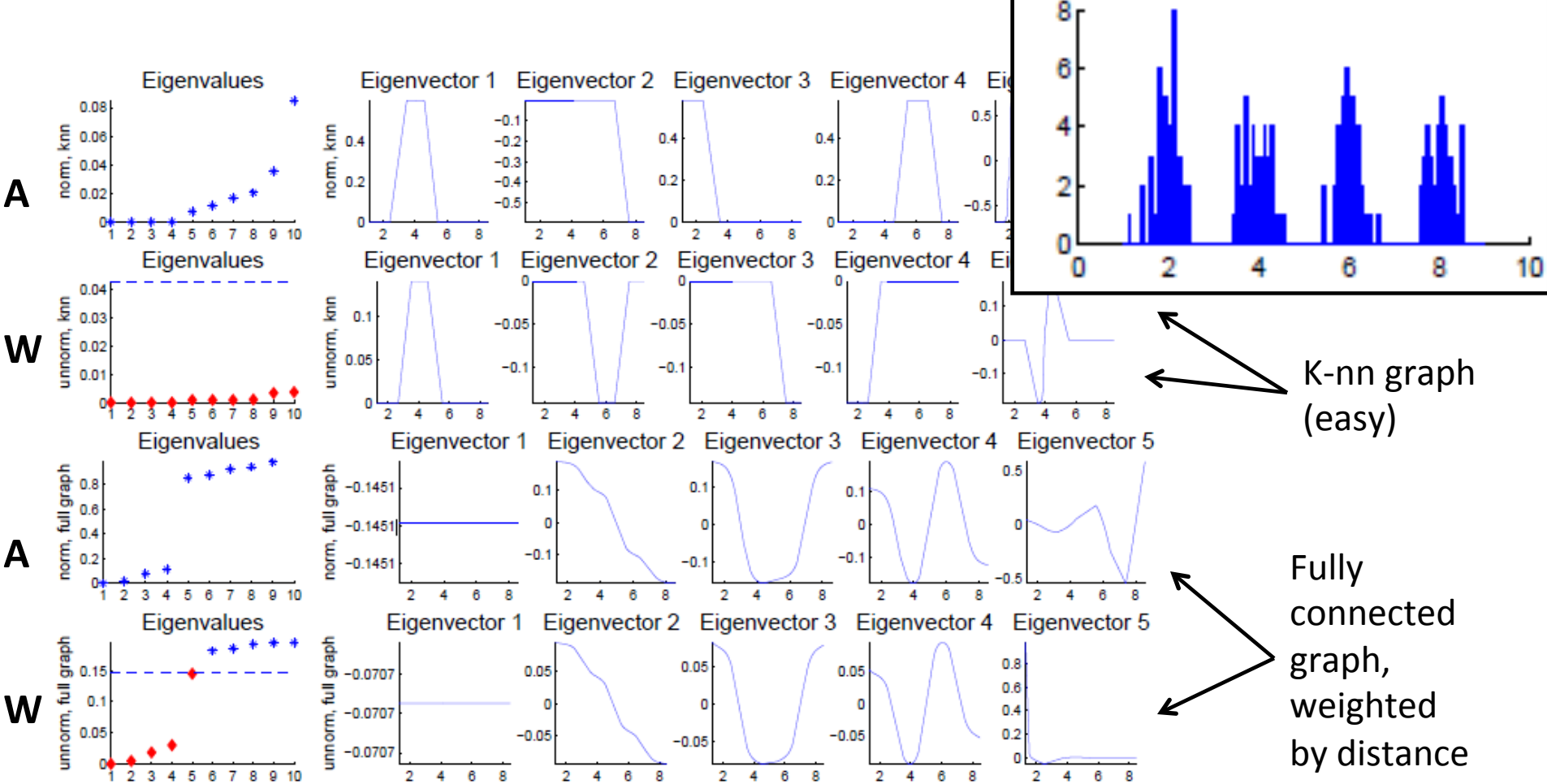
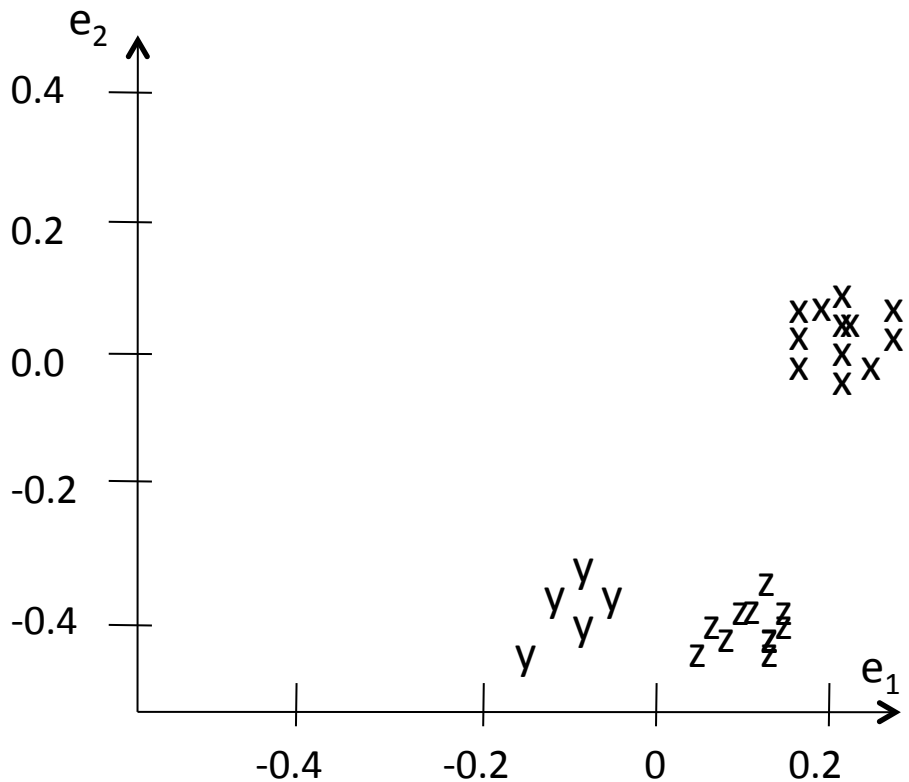
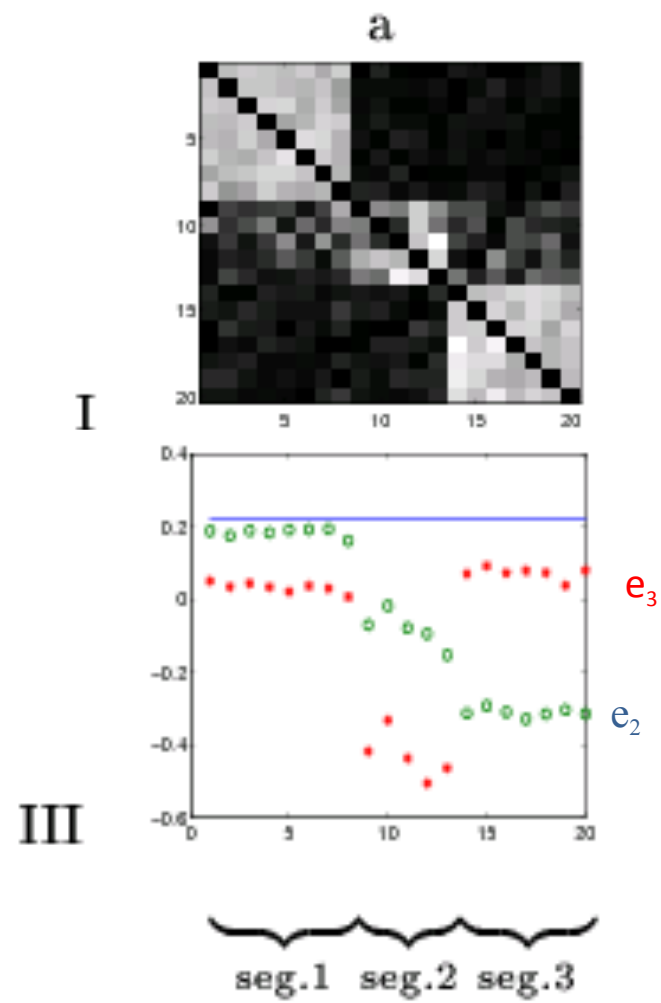


Figure 1: Toy example for spectral clustering where the data points have been drawn from a mixture of four Gaussians on $\mathbb{R}$. Left upper corner: histogram of the data. First and second row: eigenvalues and eigenvectors of L_{rw} and L based on the k-nearest neighbor graph. Third and fourth row: eigenvalues and eigenvectors of L_{rw} and L based on the fully connected graph. For all plots, we used the Gaussian kernel with $\sigma = 1$ as similarity function. See text for more details.

Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"





[Shi & Meila, 2002]

M

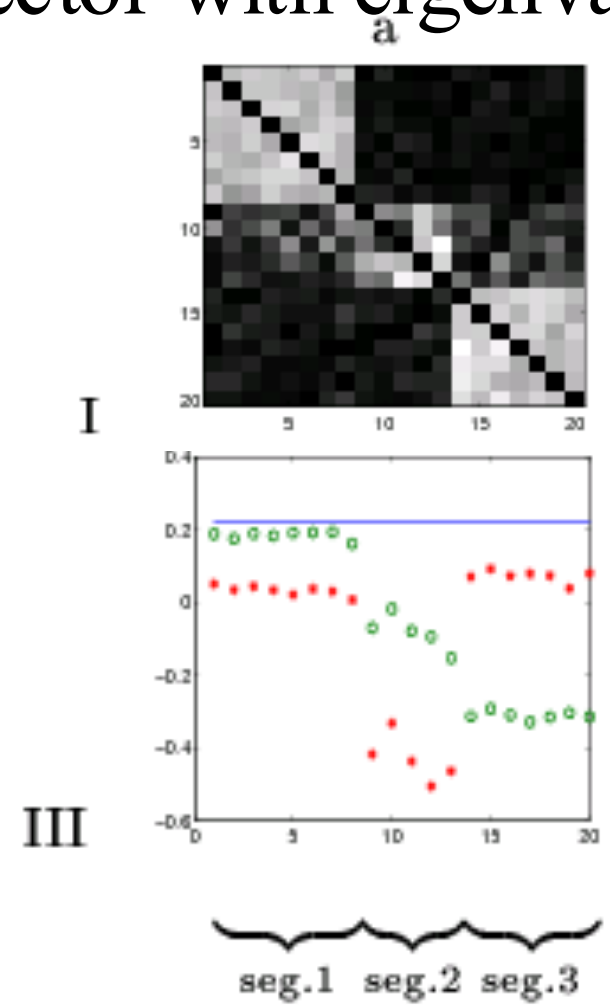
Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

If Wis connected but roughly block diagonal with *k* blocks then

- the top eigenvector is a constant vector
- the next *k* eigenvectors are roughly piecewise constant with "pieces" corresponding to blocks



M

Spectral Clustering: Graph = Matrix

$W*v_1 = v_2$ "propogates weights from neighbors"

$$\mathbf{W} \cdot \mathbf{v} = \lambda \mathbf{v} : \mathbf{v} \text{ is an eigenvector with eigenvalue } \lambda$$

If **W** is connected but roughly block diagonal with *k* blocks then

- the "top" eigenvector is a constant vector
- the next *k* eigenvectors are roughly piecewise constant with "pieces" corresponding to blocks

Spectral clustering:

- Find the top *k+1* eigenvectors $\mathbf{v}_1,\ldots,\mathbf{v}_{k+1}$
- Discard the "top" one
- Replace every node *a* with *k*-dimensional vector $x_a = \langle \mathbf{v}_2(a),\ldots,\mathbf{v}_{k+1}(a) \rangle$
- Cluster with *k*-means

M

Experimental results

Prob Method

Spectral

(c) Best alignment: Social networks

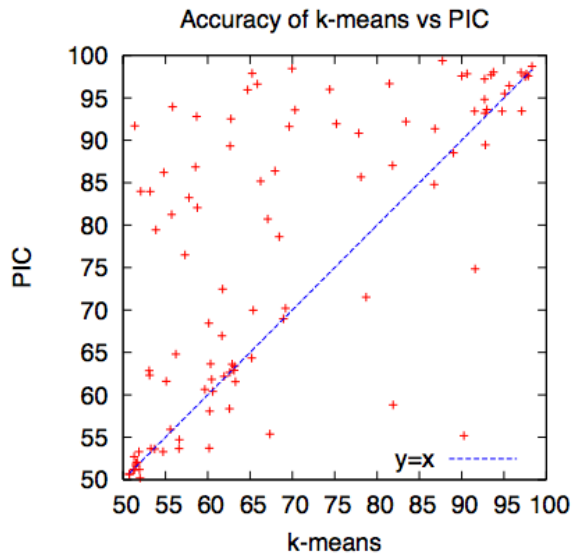
Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
Karate	**1.00**	0.91	0.93	0.95	0.95	0.95
Dolphin	0.90	**0.98**	0.98	**0.98**	**0.98**	**0.98**
UMBC	0.95	0.93	0.95	0.95	0.95	**0.96**
AG	**0.95**	0.91	0.94	0.94	0.52	0.51
MSP	**0.88**	0.63	0.63	0.63	0.63	0.64
Senate	0.98	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
PolBook	0.78	0.80	0.81	**0.83**	0.82	0.80
Football	**0.76**	0.47	0.51	0.66	0.72	0.67
MGEmail	0.28	0.39	0.40	**0.64**	0.59	0.56
CiteSeer	0.33	0.51	0.48	**0.55**	0.48	0.52
Cora	**0.47**	0.46	0.40	0.45	0.29	0.42
Average	0.75	0.73	0.73	**0.78**	0.72	0.73

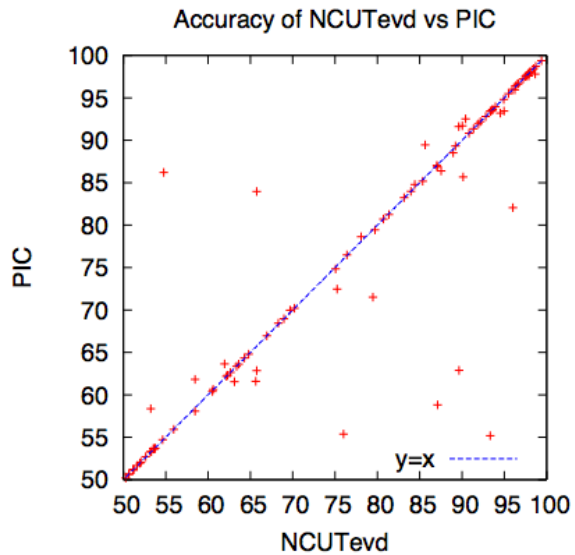
Experimental results

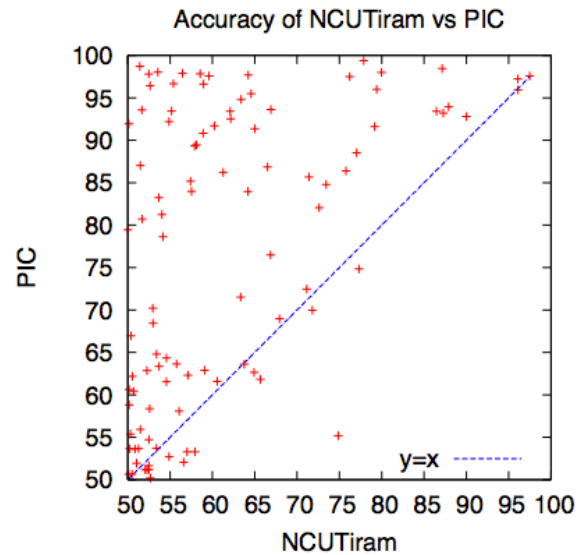
(e) 1-NN: Social networks

Dataset	PSK	PIC_D	PIC_R	PIC_{R4}	NCut	NJW
Karate	**1.00**	**1.00**	0.99	0.99	1.00	0.97
Dolphin	0.89	0.95	0.95	0.95	0.95	**0.98**
UMBC	0.92	0.93	0.93	0.93	0.92	**0.94**
AG	0.92	**0.94**	0.93	0.93	0.88	0.89
MSP	0.84	0.76	0.73	**0.86**	0.64	0.59
Senate	0.97	1.00	**1.00**	**1.00**	**1.00**	**1.00**
PolBook	0.79	0.68	0.76	0.80	**0.84**	0.78
Football	0.89	0.43	0.45	0.85	0.94	**0.95**
MGEmail	0.22	0.27	0.26	0.72	0.80	**0.81**
CiteSeer	0.34	0.55	0.54	**0.71**	0.69	0.66
Cora	0.45	0.56	0.51	**0.80**	0.47	0.75
Average	0.75	0.73	0.73	**0.87**	0.83	0.85

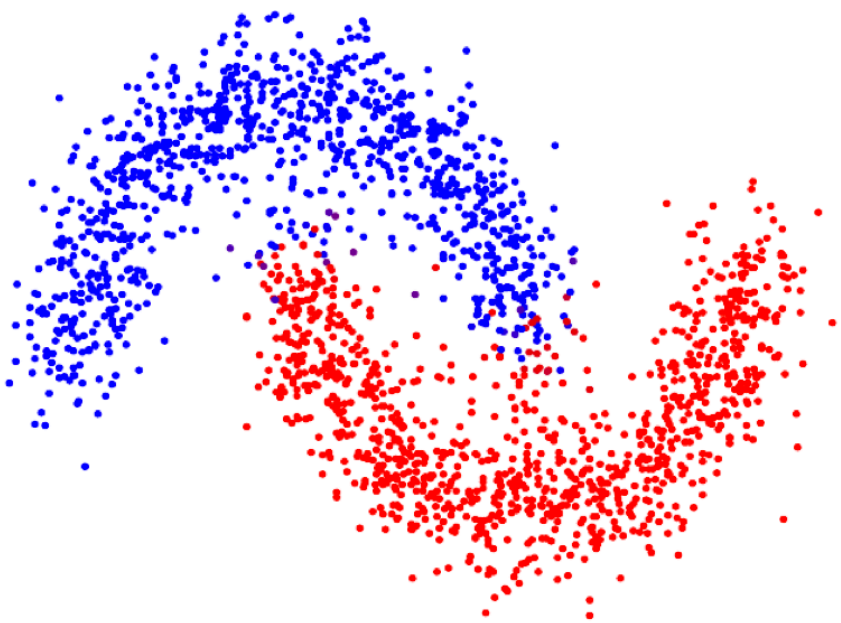
Experiments on Text Clustering

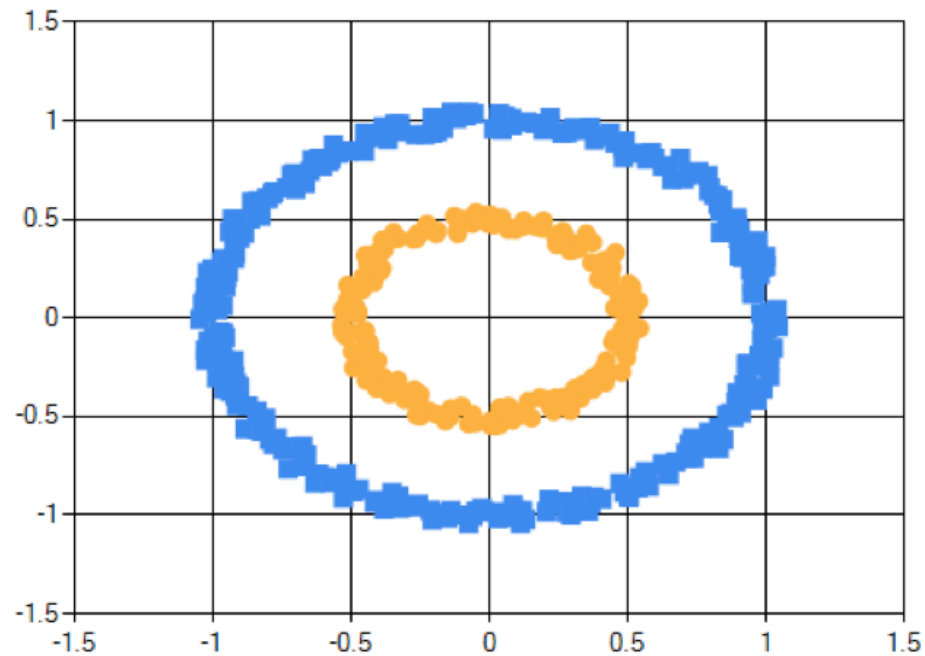






Other Spectral Clustering Examples





Spectral Clustering: Pros and Cons

- Elegant, and well-founded mathematically
- Works quite well when relations are approximately transitive (like similarity)
- Very noisy datasets cause problems
  - "Informative" eigenvectors need not be in top few
  - Performance can drop suddenly from good to terrible
- Expensive for very large datasets
  - Computing eigenvectors is the bottleneck
  - Scalable approximate methods sometimes perform less well