Clustering

with the

SVD

Carl Meyer
meyer@ncsu.edu
N.C. State U.
Mathematics Dept.

Ibai Basabe
ibaichuk@msn.com
C. of Charleston
Mathematics Dept.

Amy Langville
langvillea@cofc.edu
C. of Charleston
Mathematics Dept.

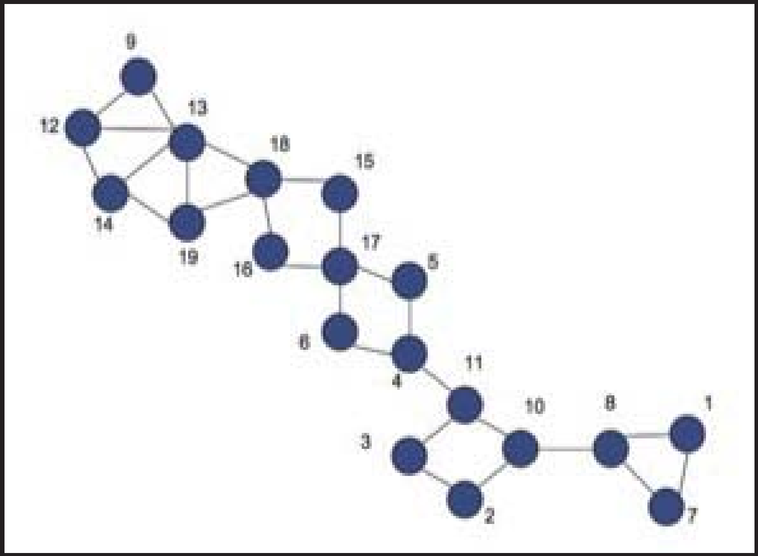
Monopoli 9/11/2007

Outline

- Fielder Method
- Clustering with the SVD
- Extended Fielder Method
- Vismatrix Tool
- Cluster Aggregation

Clustering

- A graph: directed, undirected



- A data matrix: square, rectangular, symmetric, asymmetric

(related, but subtle differences)

Fiedler Method

Fiedler Method

Clustering on an undirected graph

Matrices

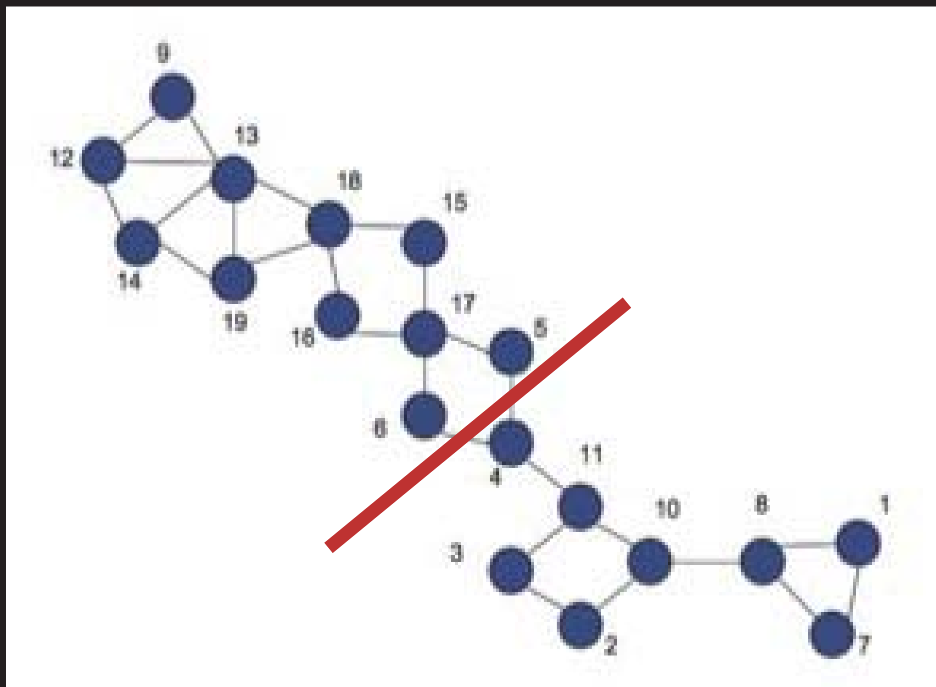
- Adjacency: **A** symmetric
- Diagonal: **D** of row sums
- Laplacian: **L** = **D** − **A**

Properties of **L**

- **L** is spd so all $\lambda_i \geq 0$.
- $\exists \ \lambda_i = 0, \forall$ connected component i
- If **L** is scc, then $\lambda_1 = 0$, $\lambda_i > 0$, $\forall \ i = 2, \ldots, n$.
- **Le** = $0 \Rightarrow \mu_L = 0$ (**D** as a type of centering)
- **L** has orthogonal e-decomp. $\Rightarrow$ $\mathbf{v}_2$ centered about 0
  (signs in $\mathbf{v}_2$ + bisection + recursion)

Fiedler Example

The subdominant eigenvector $\mathbf{v}_2$ gives info. on clustering.



	$\mathbf{v}_2$
1	-0.3502
7	-0.3502
8	-0.3259
10	-0.2549
2	-0.2450
3	-0.2182
11	-0.1762
4	-0.0433
5	0.0246
6	0.0246
17	0.0908
15	0.1539
16	0.1539
18	0.2464
19	0.2477
13	0.2556
14	0.2639
12	0.2702
9	0.2723

Why does Fiedler vector cluster?

Two-way partition $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix}$ $\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{D}_4 \end{bmatrix}$

- Assign each node to one of two clusters. $\Rightarrow$ Create decision variables x_i

  $x_i = 1$, if node i goes in cluster 1

  $x_i = -1$, if node i goes in cluster 2

- Objective: minimize the number of between-cluster links
  maximize the number of in-cluster links

$\Rightarrow$ $\min_{\mathbf{x}} \ \mathbf{x}^T \mathbf{L} \mathbf{x}$

- Suppose $\mathbf{x} = \begin{bmatrix} \mathbf{e} \\ -\mathbf{e} \end{bmatrix}$. Then

$\Rightarrow$ $\min_{\mathbf{x}} \ \mathbf{x}^T \mathbf{L} \mathbf{x}$

$$= (\mathbf{e}^T \mathbf{D}_1 \mathbf{e} + \mathbf{e}^T \mathbf{D}_4 \mathbf{e}) + (\mathbf{e}^T \mathbf{A}_2 \mathbf{e} + \mathbf{e}^T \mathbf{A}_3 \mathbf{e}) - (\mathbf{e}^T \mathbf{A}_1 \mathbf{e} + \mathbf{e}^T \mathbf{A}_4 \mathbf{e})$$

$\downarrow$ for balancing $\downarrow$ between-cluster links $\uparrow$ in-cluster links

Why does Fiedler vector cluster?

- Optimization Problem $\min \mathbf{x}^T \mathbf{L} \mathbf{x}$ is NP-hard.
  $\Rightarrow$ Relax from discrete to continuous values for $\mathbf{x}$.

- By Rayleigh theorem,

$$\min_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{L} \mathbf{x} = \lambda_1,$$

with $\mathbf{x}^*$ as the eigenvector corresponding to λ_1.

BUT, $\mathbf{x}^* = \mathbf{e}$, which is not helpful for clustering!

Optimization Solution

- Add constraint $\mathbf{x}^T\mathbf{e} = 0$.
- By Courant-Fischer theorem,

$$\min_{\|\mathbf{x}\|_2=1,\ \mathbf{x}^T\mathbf{e}=0} \mathbf{x}^T\mathbf{L}\mathbf{x} = \lambda_2,$$

with $\mathbf{x}^* = \mathbf{v}_2$ as the eigenvector corresponding to λ_2.

$\mathbf{v}_2$ is called the Fiedler vector.

Notes on Fiedler

- getting $\mathbf{v}_2$, skipping over $\mathbf{v}_1 = \mathbf{e}$ whose e-value is 0.

- You could use $\mathbf{L} = \mathbf{D} - \mathbf{A}$ (s.p.d.) and find 2^{nd} smallest e-vector

  OR

  You could use $\mathbf{L} = \mathbf{A} - \mathbf{D}$ (s.n.d.) and find 2^{nd} largest e-vector.

  Gives the same clustering.

- When **A** is symmetric and square,

  e-vectors of $\mathbf{L}$ = e-vectors of $\mathbf{L}^T\mathbf{L}$ = s-vectors of $\mathbf{L}$

  — Given $\mathbf{L}$ has e-decomp. $\mathbf{V}\mathbf{L}\mathbf{V}^T = \mathbf{D}$ and $\mathbf{L} = \mathbf{L}^T$.

  — Then $\mathbf{L}\mathbf{L}^T$ has e-decomp. $\mathbf{V}\mathbf{L}^T\mathbf{L}\mathbf{V}^T = \mathbf{V}\mathbf{L}^2\mathbf{V}^T = \mathbf{D}^2$

  $\Rightarrow$ $\mathbf{L}$ and $\mathbf{L}^T\mathbf{L}$ have same e-vectors.

What happens if A is asymm. or rect.?

e-vectors not centered about 0 or do not exist

Solution 1: Symmetrize so that standard Fiedler can be used.

Solution 2: Use s-vectors instead.

Singular Value Decomposition

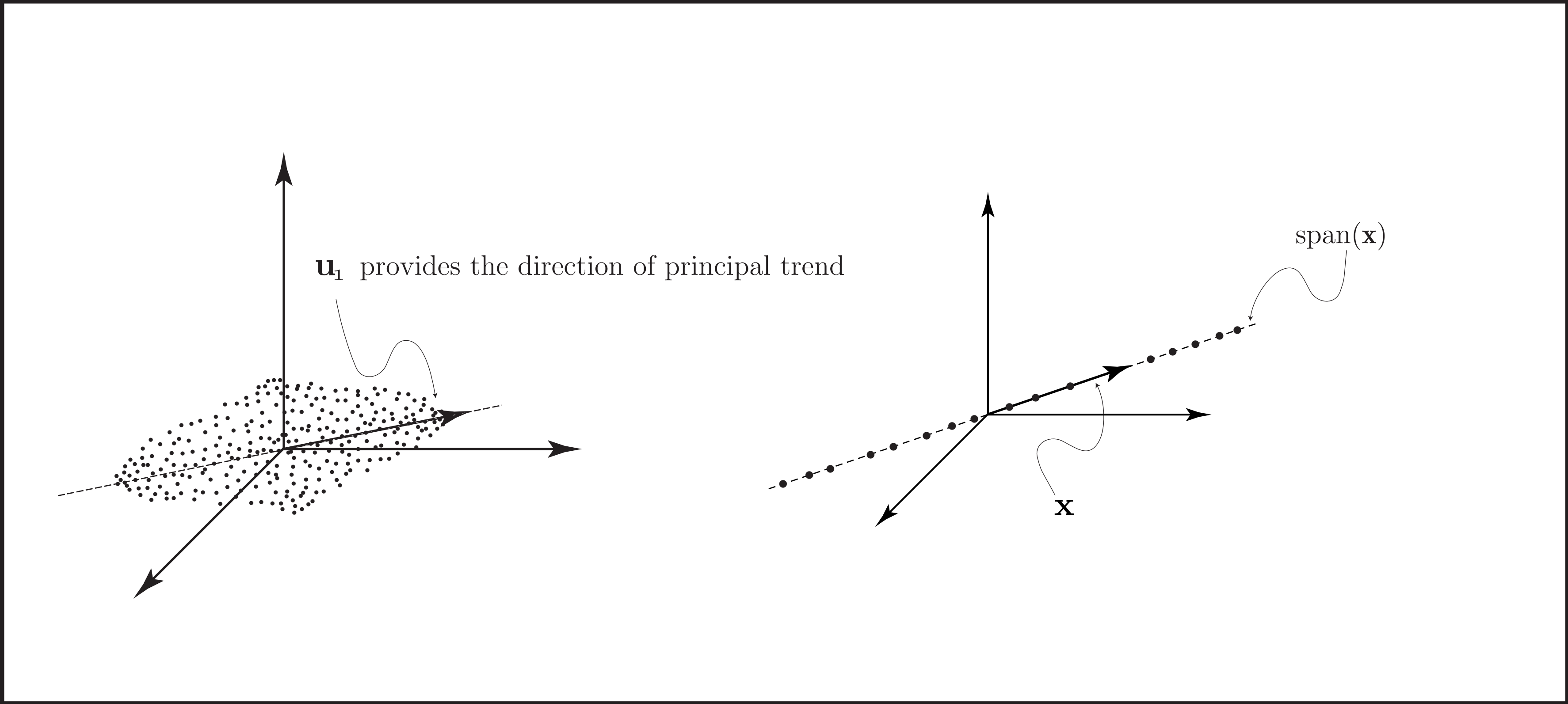
$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times m}\Sigma\mathbf{V}_{n\times n}^T$$

- orthogonal matrix $\mathbf{U} = [\,\mathbf{u}_1|\mathbf{u}_2|\ldots|\mathbf{u}_m\,]$ of left singular vectors
- orthogonal matrix $\mathbf{V} = [\,\mathbf{v}_1|\mathbf{v}_2|\ldots|\mathbf{v}_n\,]$ of right singular vectors
- rank-r matrix $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i\mathbf{u}_i\mathbf{v}_i^T$

Why does svd's $\mathbf{v}_2$ work?

$\mathbf{u}_1$ as direction of maximal variance/principal trend

(PCA)



- need data that has been centered, i.e., mean is 0.

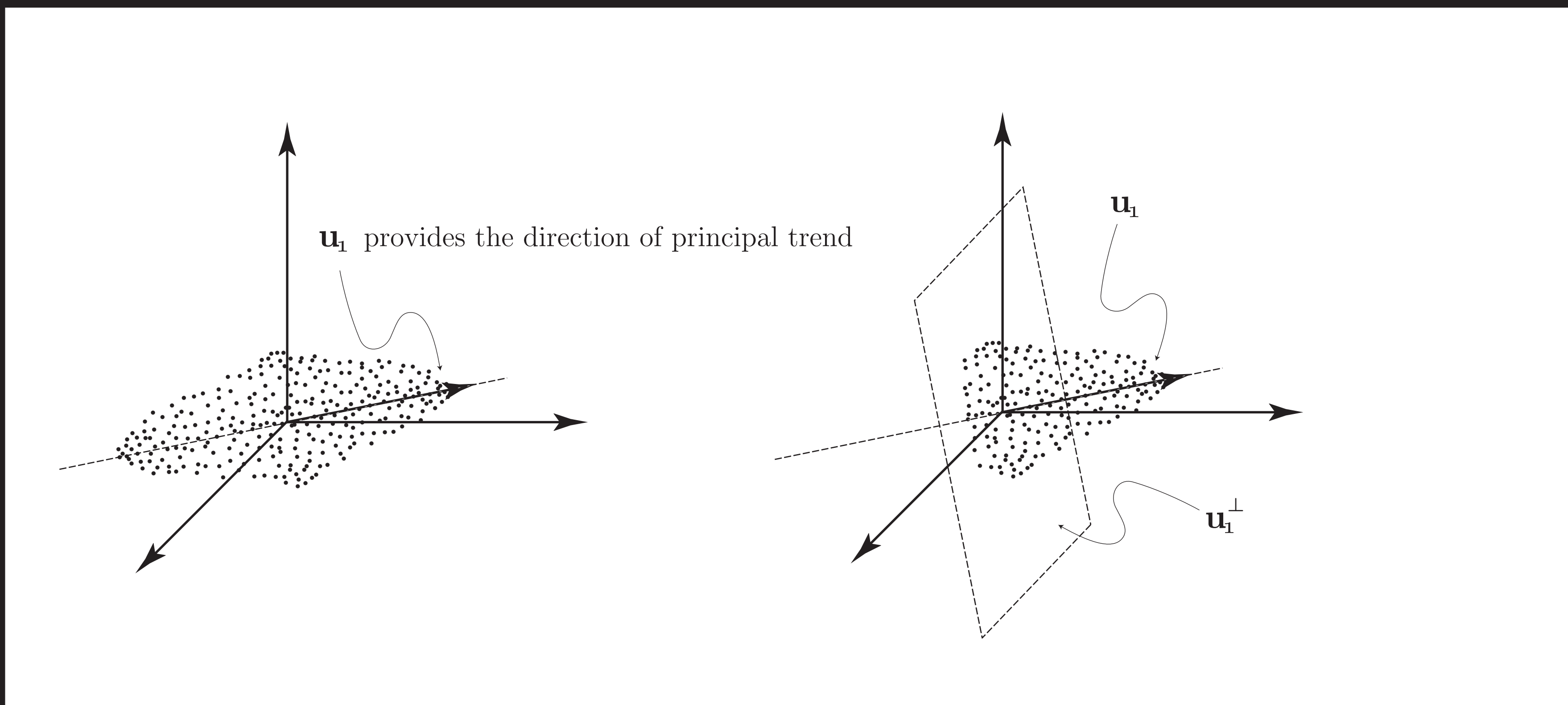
Centered Data

$$\mathbf{C} = \mathbf{A} - \mu \mathbf{e}^T$$

($\boldsymbol{\mu}$ is mean of columns of **A**)

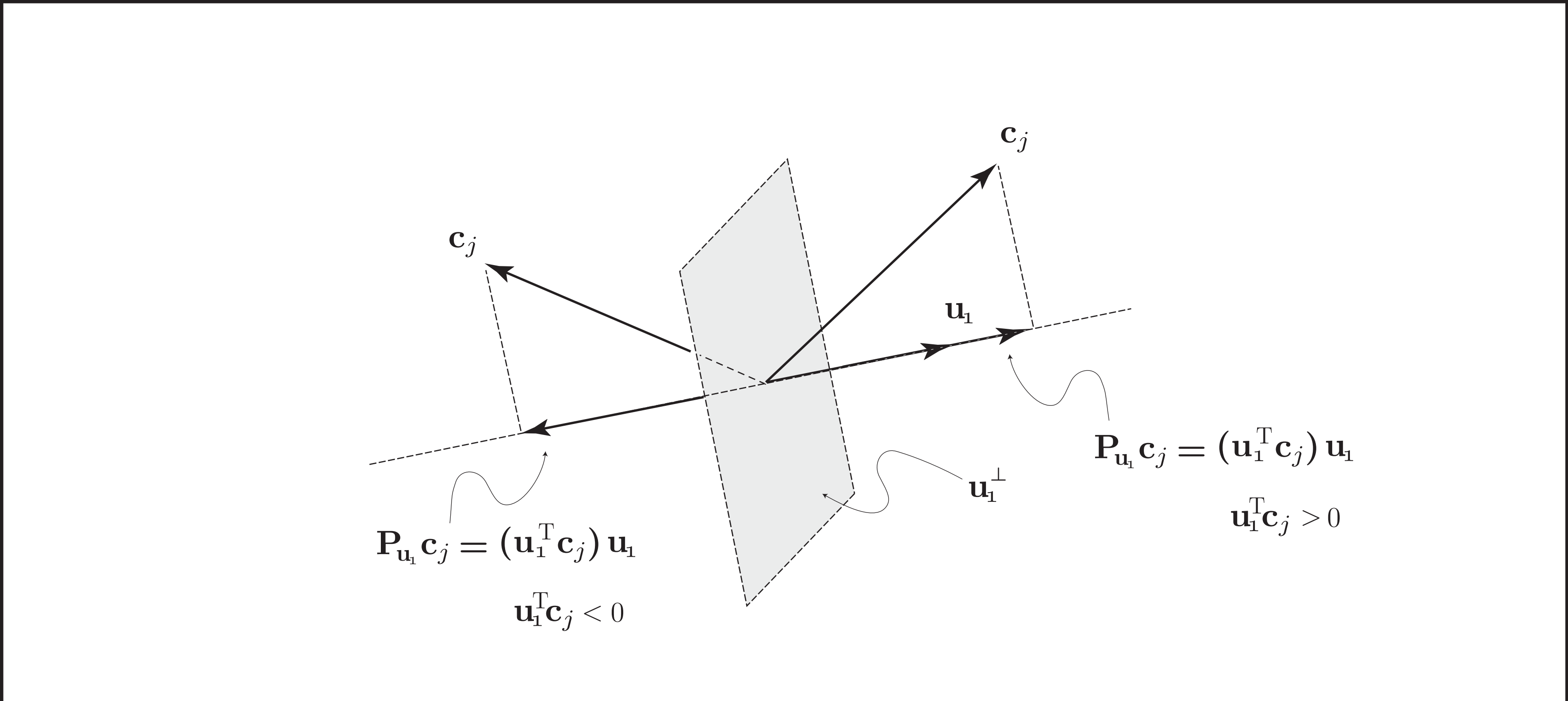
(Recall Laplacian **L=A−D**.)

- Partition data into two sets using $\mathbf{u}_1^{\perp}$ wall



In Front of or Behind the Wall?

IN FRONT when $\mathbf{u}_1^T \mathbf{c}_j > 0$ and BEHIND when $\mathbf{u}_1^T \mathbf{c}_j < 0$



- Because

$$\mathbf{u}_1^T \mathbf{C} = \mathbf{u}_1^T \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sigma_1 \mathbf{v}_1^T$$

the signs in $\mathbf{v}_1$ give information about the principal partition.

Further Partitioning

Recursion

- work on submatrices (PDDP)

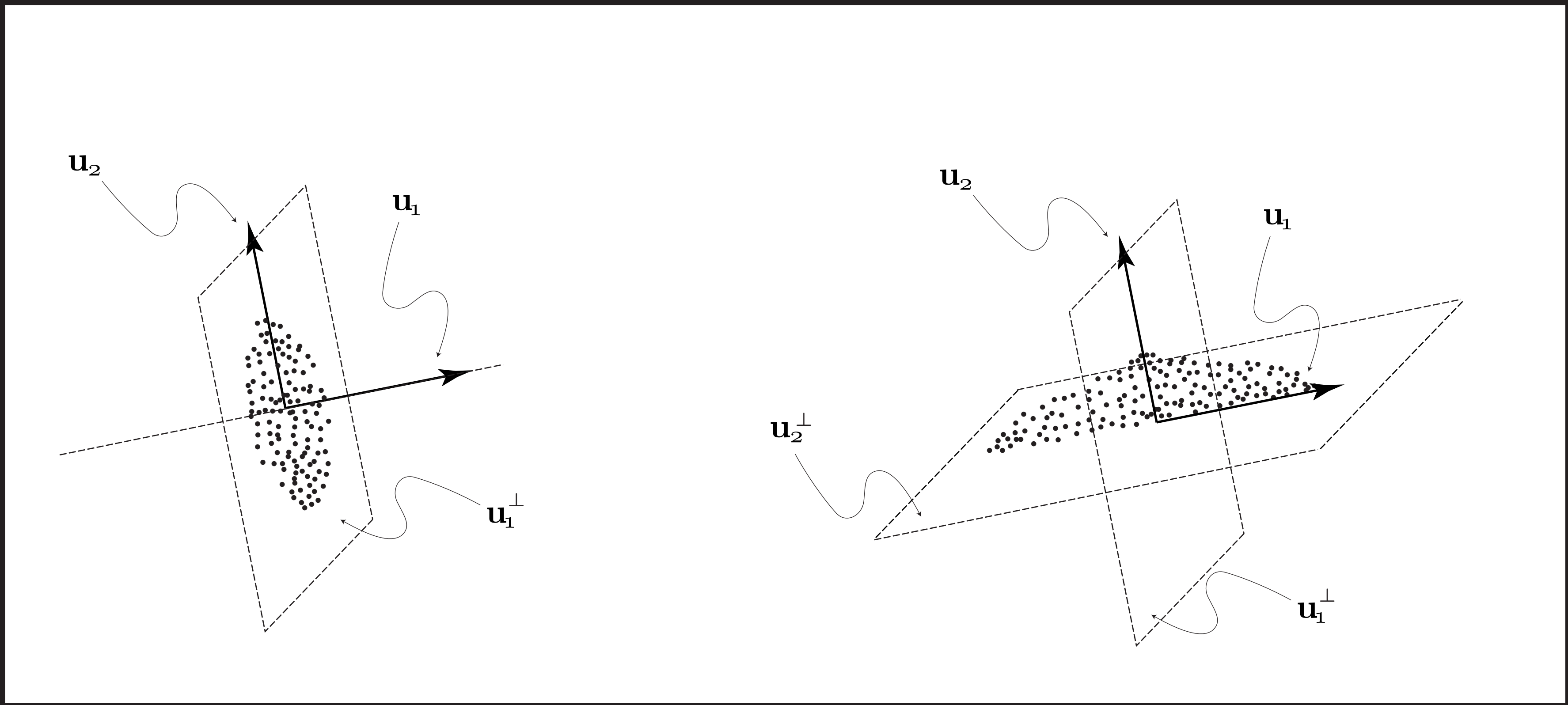
Secondary Partitions

- use secondary s-vectors (Extended Fiedler)

Extended Fiedler

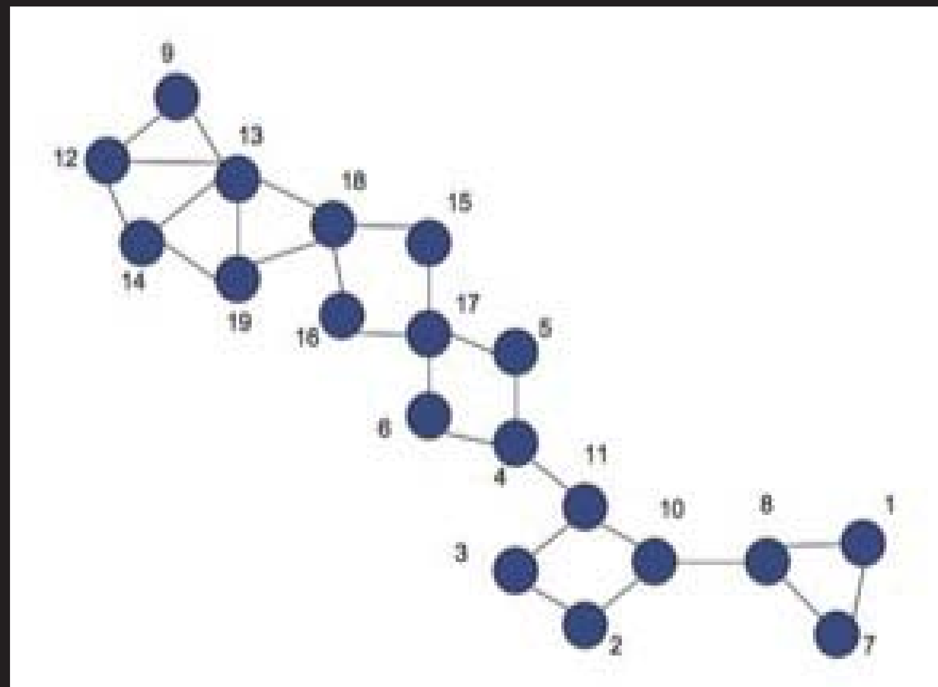
Other subdominant s-vectors

- If $\mathbf{v}_1$ gives approximate info. about clustering, what about the other s-vectors $\mathbf{v}_2, \mathbf{v}_3, \ldots$?



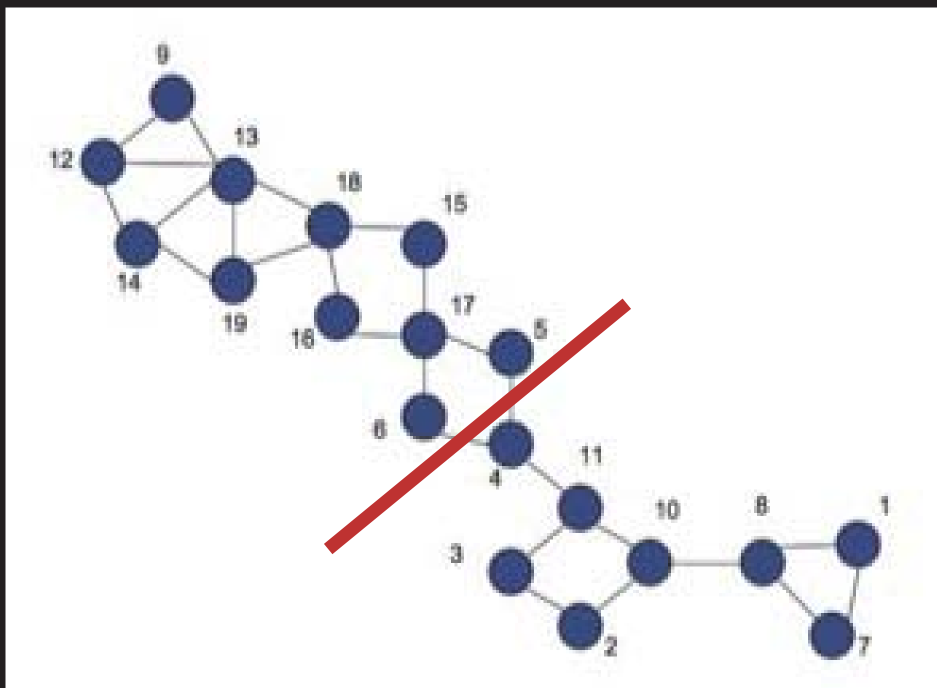
- $\mathbf{u}_1$ and $\mathbf{u}_2$ create quadrants to partition data.
  signs in $\mathbf{v}_1$ and $\mathbf{v}_2$ tell us which quadrant data point is in.
- $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \Rightarrow$ octants

Example Graph

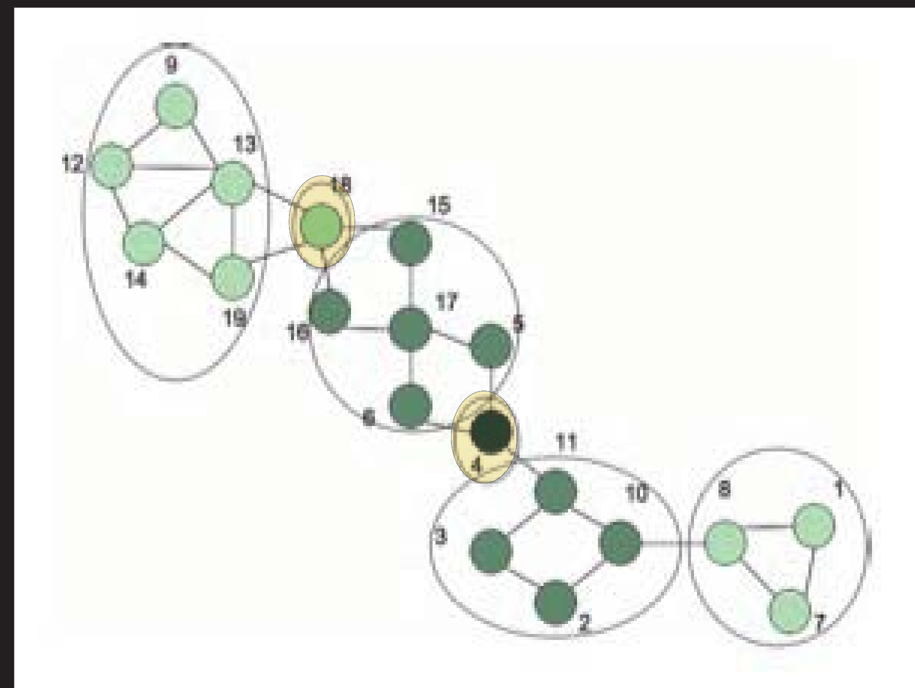


Clustered Example Graph

using one vector



using three vectors



- Nodes 4 and 18 are “saddle nodes” that straddle a few clusters.

Clustered Example Eigenvectors

using three vectors

- - +	1	-0.3502	-0.3427	0.2823	0.0292	0.0006
	7	-0.3502	-0.[illegible]427	0.2[illegible]23	0.0292	0.0006
	8	-0.3259	-0.2439	0.1415	-0.0023	-0.0005
- + -	10	-0.2549	0.0243	-0.2107	-0.0629	-0.0019
	2	-0.2450	0.1199	-0.4371	-0.2083	-0.1154
	3	-0.2182	0.1[illegible]09	-0.4454	-0.1288	-0.0125
	11	-0.1762	0.1898	-0.2315	0.0898	0.1138
- + +	4	-0.0433	0.3093	0.[illegible]70	0.3641	0.1424
+ + +	5	0.0246	0.3245	0.2121	0.3047	0.0232
	6	0.0246	0.3245	0.2121	0.3047	0.0232
	17	0.0908	0.2[illegible]61	0.2[illegible]14	-0.0836	-0.1395
	15	0.1539	0.1321	0.2104	-0.4268	-0.1714
	16	0.1539	0.1321	0.2104	-0.4268	-0.1714
+ - +	18	0.2[illegible]64	-0.0199	0.0[illegible]45	-0.3092	0.1182
+ - -	19	0.2477	-0.1537	-0.0609	-0.0938	0.5190
	13	0.2556	-0.1846	-0.0990	0.0443	0.0749
	14	0.2639	-0.2122	-0.1[illegible]77	0.0849	0.3906
	12	0.2702	-0.2372	-0.1595	0.2126	-0.1545
	9	0.2723	-0.2465	-0.1722	0.2791	-0.6393

6 clusters,

$6 \in [3, 2^3]$

- Sign patterns in e/s-vectors give clusters and saddle nodes.
- Number of clusters found is not fixed, but lies in $[j, 2^j]$, where j is the number of vectors used.
- takes a global view for clustering (whereas recursive Fiedler tunnels down, despite possible errors at initial iterations.)

Term-by-Document Matrices

SVD to cluster Term-Doc Matrices

$$\mathbf{C} \approx \mathbf{C}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$$

$\Rightarrow$ sign pattern in $\mathbf{U}_k$ will cluster terms.

$\Rightarrow$ sign pattern in $\mathbf{V}_k$ will cluster documents.

Pseudocode

For Term Clustering

k = truncation point for SVD

- input $\mathbf{U}_k$ (matrix of k left singular vectors of **C**)
- input j (user-defined scalar, # of clusters $\in [j, 2^j]$. Note: $j \leq k$.)
- create $\mathbf{B} = \mathbf{U}(:, 1 : j) >= 0)$; binary matrix with sign patterns
- associate a number x(i) with each unique sign pattern

```
x=zeros(n,1);
for i=1:j
    x=x+(2^(j-i))*(B(:,i));
end
```

- reorder rows of **A** by indices in sorted vector x

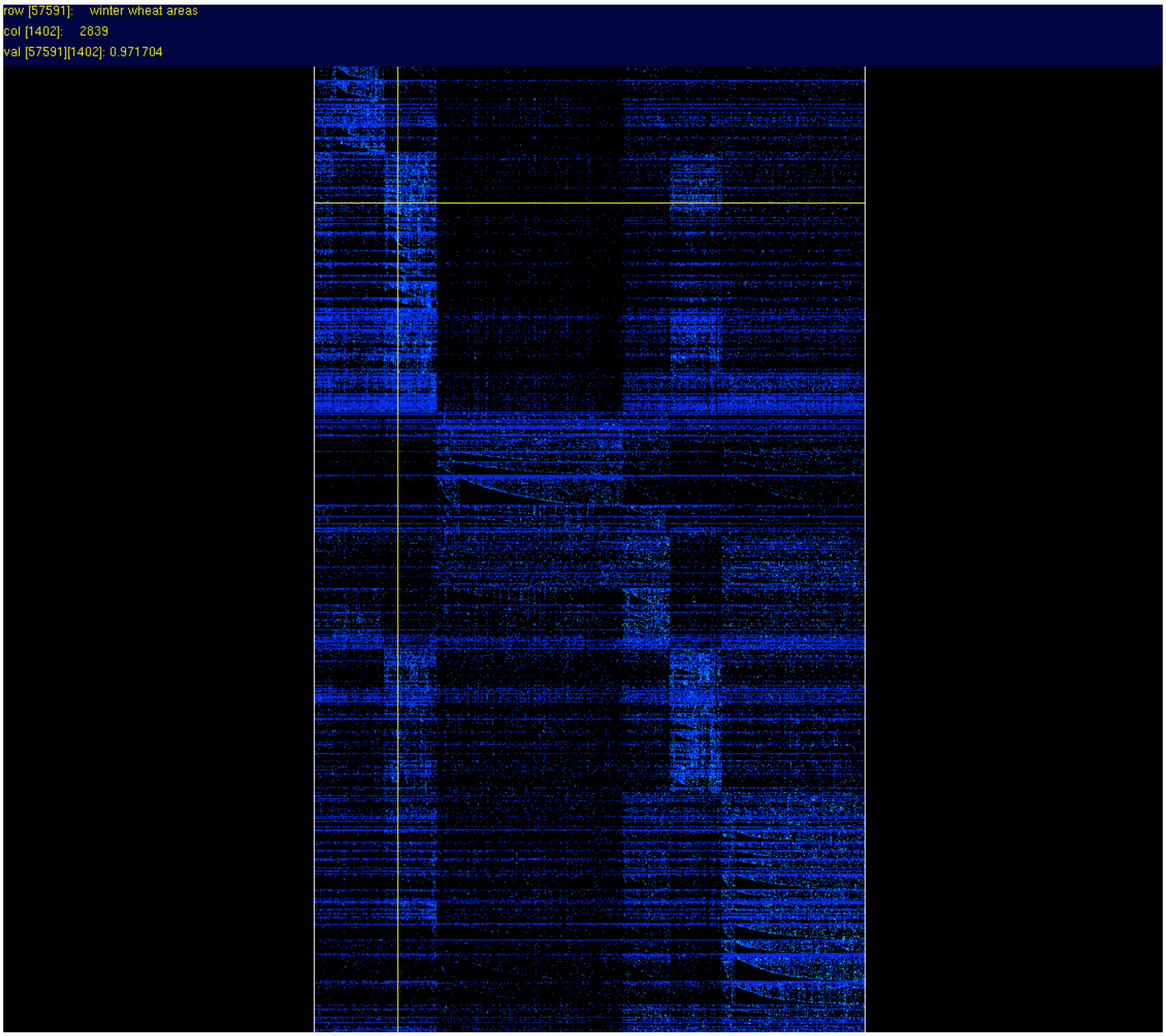
vismatrix tool

David Gleich

SVD Clustering of Reuters10

72K terms, 9K docs, 415K nonzeros

- $j = 10$ for terms produces 486 term clusters
  $j = 4$ for documents produces 8 document clusters

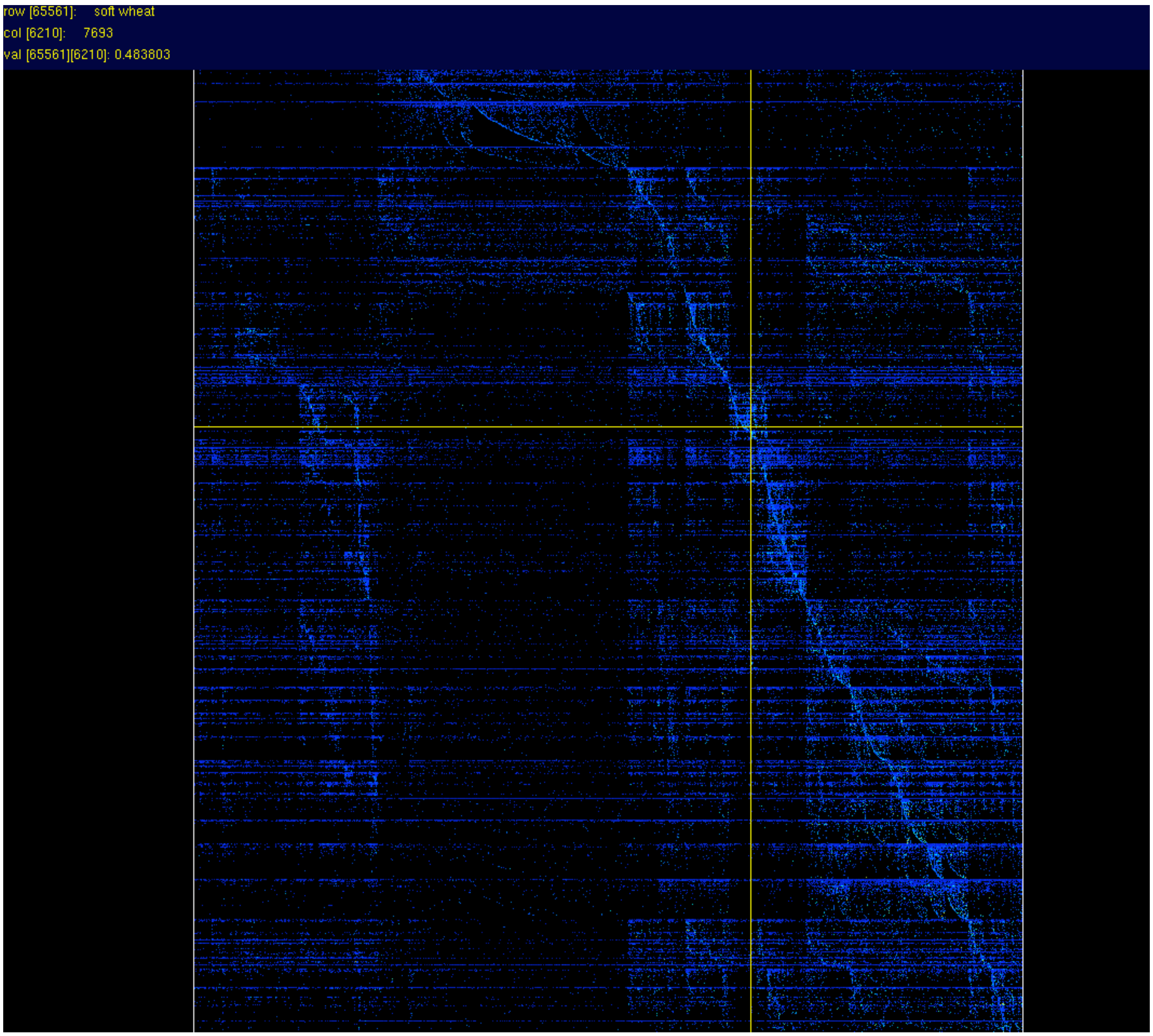

row [57591]: winter wheat areas
col [1402]: 2839
val [57591][1402]: 0.971704

SVD Clustering of Reuters10

72K terms, 9K docs, 415K nonzeros

- $j = 10$ for terms produces 486 term clusters
  $j = 4$ for documents produces 8 document clusters

- $j = 10$ for terms produces 486 term clusters
  $j = 10$ for documents produces 391 document clusters


row [65561]: soft wheat
col [6210]: 7693
val [65561][6210]: 0.483803

Summary of SVD Clustering

+ variable # of clusters returned between j and 2^j
+ sign pattern idea allows for natural division within clusters
+ clusters ordered so that similar clusters are nearby each other
+ less work than recursive Fiedler
+ can choose different # of clusters for terms and documents
+ can identify "saddle terms" and "saddle documents"

– only hard clustering is possible
– picture can be too refined with too many clusters
– as j increases, range for # of clusters becomes too wide

  EX: j=10, # of clusters is between 10 and 1024

– In some sense, terms and docs are treated separately.

  (due to symmetry requirement)

Practical Issues

- Centering data and sparsity—modified Lanczos
- 2-way vs. 3-way splitting
- splitting at 0 vs. gap partitioning
- magnitudes vs. signs
- meaning of singular values

Cluster Aggregation

many clustering algorithms = many clustering results

$\Rightarrow$ Can we combine many results to make one super result?

- Create aggregation matrix **F**

  $\mathbf{f}_{ij}$ = # of methods having items i and j in the same cluster

- Run favorite clustering method on **F**

Cluster Aggregation Example

Method 1

item	cluster assignment
1	1
2	3
3	2
4	1
5	1
6	2

1 4 5

3 6

2

Method 2

item	cluster assignment
1	3
2	1
3	2
4	3
5	1
6	2

1 4

3 6

2 5

Method 3

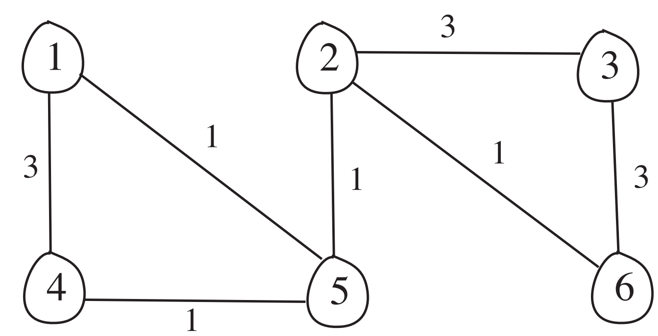
item	cluster assignment
1	1
2	2
3	2
4	1
5	3
6	2

1 4

2 3 6

5

Cluster Aggregated Graph



Cluster Aggregated Results

Fiedler using just one eigenvector

1 4 5

2 3 6

Fiedler using two eigenvectors

1 4

3 6

5

2

Conclusions

Clustering with the SVD

- explains why Fiedler method works and explains use of **D**
- does not require expensive recursion
- # of clusters returned is not a user-defined parameter
- clusters ordered so that similar clusters are nearby each other
- can choose different # of clusters for terms and documents
- can identify “saddle terms” and “saddle documents”
- cluster aggregation emphasizes strong connections across methods, dilutes effect of outliers