Spectral Graph Clustering

Zhirong Yang

5.12.2007

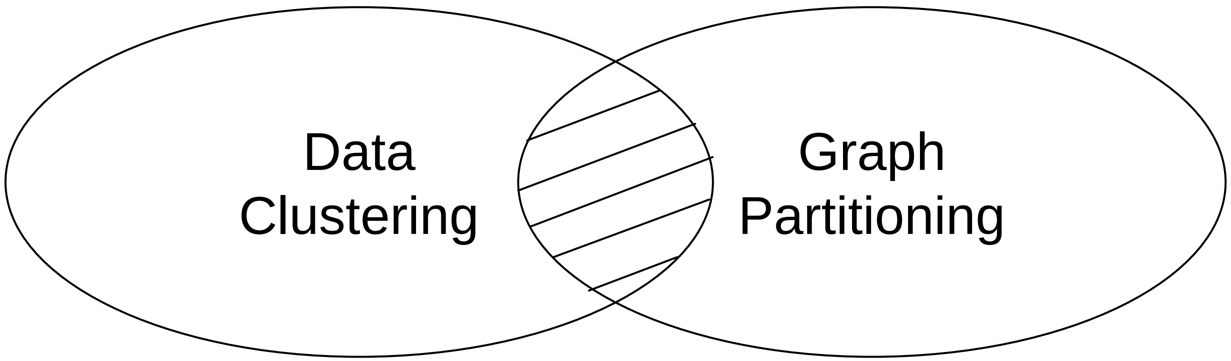
M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74, 036104 (2006).

Satu Elisa Schaeffer. Graph Clustering. Computer Science Review 1(1):27-64,2007.

Ravi Kannan, Santos Vempala, Adrian Vetta, On clusterings: good, bad and spectral. Journal of the ACM (JACM) 51(3), 497--515, 2004

Graph Partitioning & Data Clustering

- Data clustering is to partition a data set into subsets (clusters), so that the data in each subset share some common trait - often proximity according to some defined distance measure — Wikipedia
- Given a similarity measure, data clustering can be viewed as a partioning problem of an undirected (weighted) graph.
- Some properties, e.g. power law and self-similarity, generally does not hold for the data clustering graphs.
- Other methods for data clustering exist.



Spectral Graph Clustering

- Most ideal graph clustering objectives require NP-hard optimization.
- An alternative is to get an approximated solution by studying the the spectrum of some matrix.
- Devise a matrix **G** based on the adjacency matrix **W**.
- Solve the eigenvalue decomposition problem of **G**.
- The eigenvectors approximately indicate the membership of nodes to the clusters.

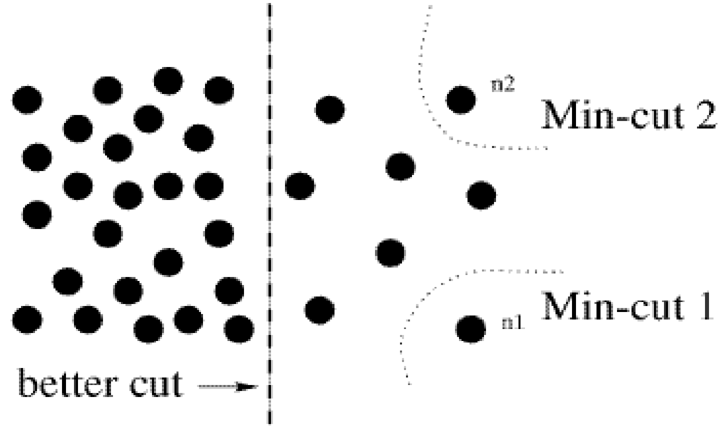
$$cut(A,B) = \sum_{u \in A, v \in B} W_{uv} \qquad asso(A,V) = \sum_{u \in A, t \in V} W_{ut}$$

Average Association

- Maximize $\frac{asso(A,A)}{|A|} + \frac{asso(B,B)}{|B|}$
- $\mathbf{W}\mathbf{x}=\lambda\mathbf{x}$
- x_i=1 if vertex$_i$ belongs to the cluster and 0 otherwise.
- Finds the most cohesive clusters.
- + Good for feaure extraction
- - the approximation is not tight
- - may result in small but tight clusters in the data.

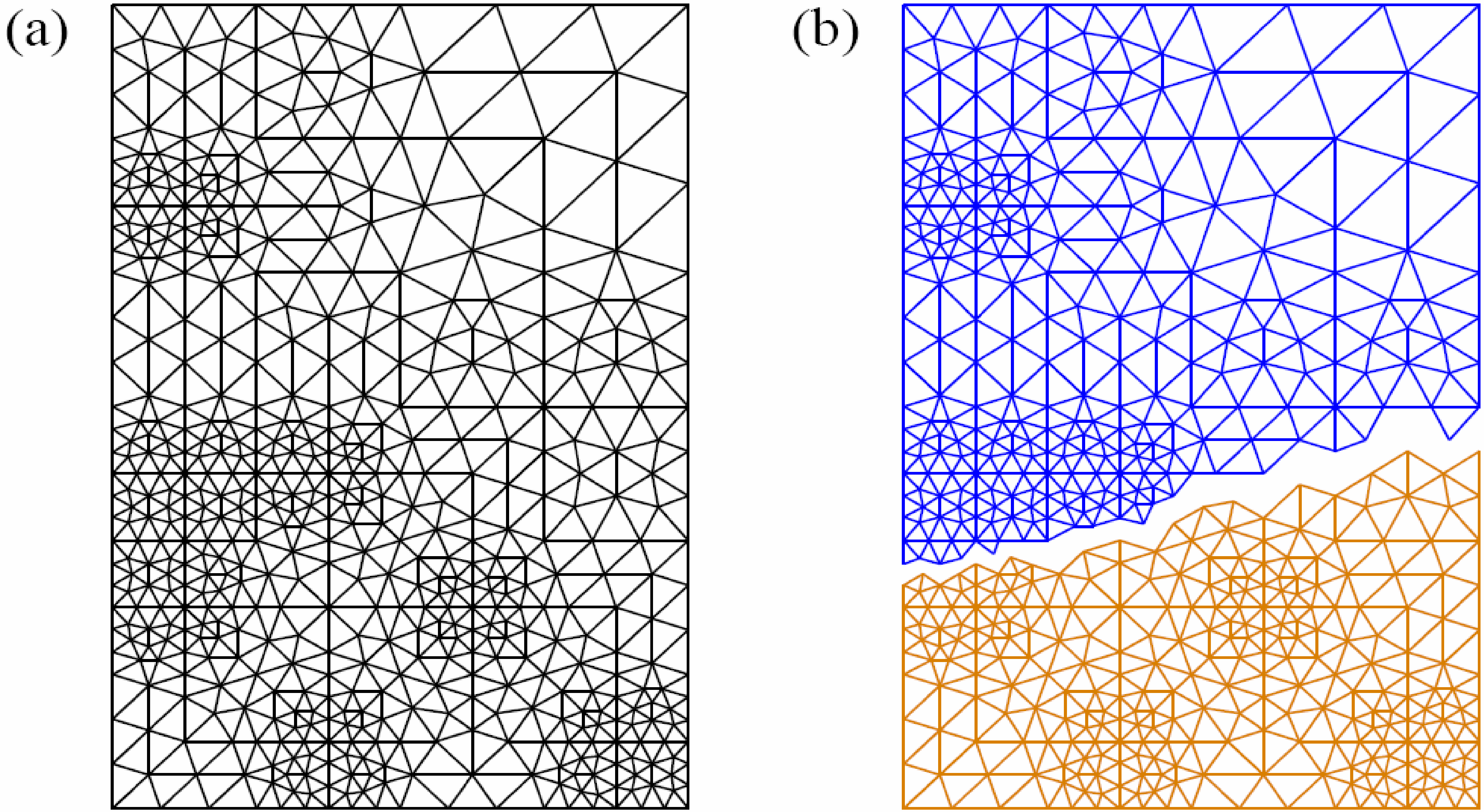
Minimizing Average Cut

- Minimize $\frac{cut(A,B)}{|A|} + \frac{cut(A,B)}{|B|}$
- (**D**-**W**)**x**=λ**x**, where **D** = diag(sum(**W**))
- **L** = **D** – **W** is called the Laplacian matrix.
- The sign of s_i indicates the membership.
- - Unweighted minimum cut tends to favor cutting off small regions.



- - Average cut cannot ensure the two partitions computed will have tight within-group similarity.

A Bad Result by Average Cut



Minimizing Normalized Cut

- Minimize $$\frac{cut(A,B)}{asso(A,V)}+\frac{cut(A,B)}{asso(B,V)}$$
- $(\mathbf{D}-\mathbf{W})\mathbf{x}=\lambda\mathbf{D}\mathbf{x}$
- Approximates Beltrami-Laplace operator in a Riemannian manifold.
- The smallest eigenvalue corresponds to a trivial eigenvector (1,1,1,...,)
- The Fiedler vector, the eigenvector with the second smallest eigenvalue, serves as the most significant indicator.

Cut-based Methods Summary

Finding clumps ⟷ Finding splits

	Average association	Normalized Cut	Average cut
Discrete formulation	$\frac{asso(A,A)}{\lvert A \rvert} + \frac{asso(B,B)}{\lvert B \rvert}$	$\frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$ or $2 - \left(\frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)}\right)$	$\frac{cut(A,B)}{\lvert A \rvert} + \frac{cut(A,B)}{\lvert B \rvert}$
Continuous solution	$Wx = \bar{\lambda} x$	$(D-W)\, x = \bar{\lambda} D\, x$ or $W\, x = (1-\bar{\lambda}) D\, x$	$(D-W)\, x = \bar{\lambda}\, x$

Maximizing Modularity

- "A good division of a network into communities is not merely one in which the number of edges running between groups is small. Rather, it is one in which the number of edges between groups is smaller than expected."
- Q = (number of edges within communities) − (expected number of such edges).

$$Q = \frac{1}{2m} \sum_{ij} \left[W_{ij} - P_{ij} \right] \delta(g_i, g_j)$$

g_i: the community to which vertex i belongs
$\delta(r,s)=1$ if r=s and 0 otherwise
m: total number of edges
P_{ij}: the expected number of edges between i and j

A Suggested P_{ij}

- Constraint 1: $$\sum_{ij} P_{ij} = \sum_{ij} W_{ij} = 2m$$
- P_{ij}=constant is not a good representation of most real world networks.
- Constraint 2: $$\sum_{j} P_{ij} = k_i$$

  k_i is actual degree of vertex i in the real network, i.e. $$k_i = \sum_j W_{ij}$$
- The suggested model $$P_{ij} = \frac{k_i k_j}{2m}$$

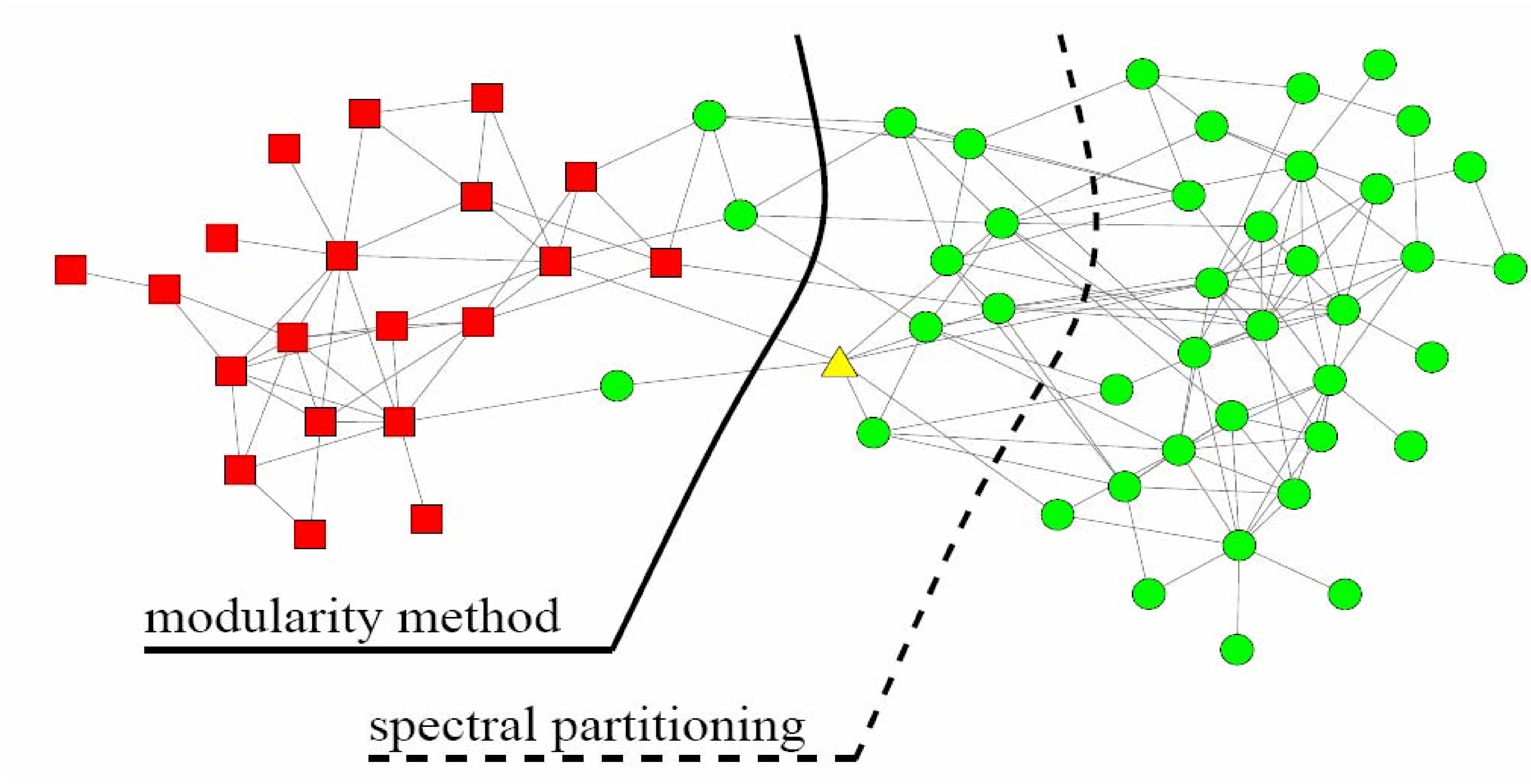
Spectral Optimization of Modularity

- Write $\delta(g_i,g_j)=(x_i+x_j)/2$

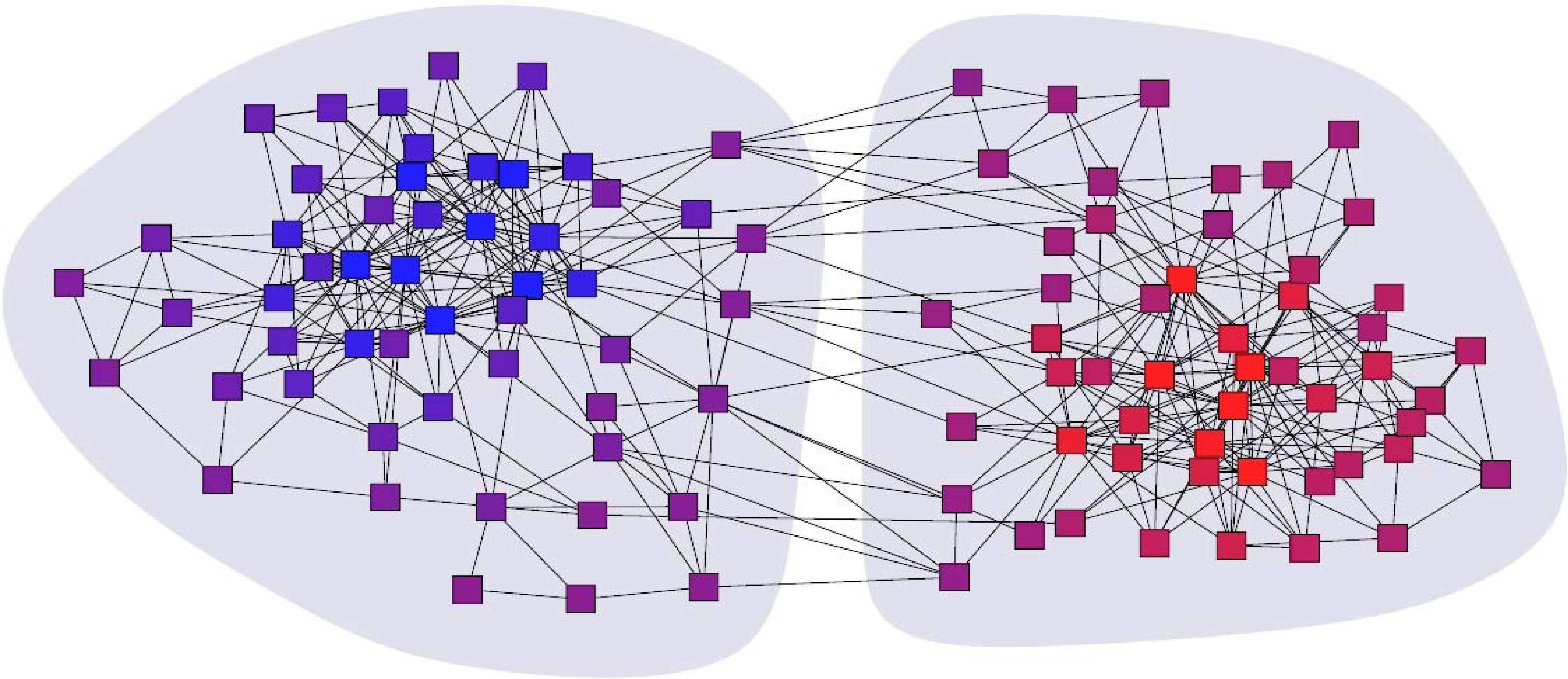
$$Q=\frac{1}{4m}\sum_{ij}\left[W_{ij}-P_{ij}\right]x_i x_j=\frac{1}{4m}\mathbf{x}^T(\mathbf{W}-\mathbf{P})\mathbf{x}=\frac{1}{4m}\mathbf{x}^T\mathbf{G}\mathbf{x}$$

- The most significant indicator is given the leading eigenvector of **G**

Example 1: Dolphin Social Network



Example 2: Political Books



Other Eigenvectors

- The leading eigenvector divides the network into only two communities.
- The information in the other eigenvectors may also be useful.
- Problem: only those eigenvectors corresponding to positive eigenvalues can give positive contributions to the modularity.

$$Q = n\alpha + \mathrm{Tr}\left[\mathbf{X}^{\mathrm{T}}\mathbf{U}(\mathbf{\Lambda} - \alpha\mathbf{I})\mathbf{U}^{\mathrm{T}}\mathbf{X}\right]$$

$$= n\alpha + \sum_{j=1}^{n}\sum_{k=1}^{c}(\beta_j - \alpha)\left[\sum_{i=1}^{n} U_{ij} X_{ik}\right]^2$$

svd(**G**)=**UΛU**$^{\mathrm{T}}$, with Λ=diag(β).
n: total number of vertices
c: number of communities

$$X_{ij} = \begin{cases} 1 & \text{if vertex i belongs to community j} \\ 0 & \text{otherwise} \end{cases}$$

Vector Partitioning Algorithm

- Provided $\alpha \leq \beta_p$, define vertex vectors $[\mathbf{r}_i]_j = \sqrt{\beta_j - \alpha} U_{ij}$

$$Q \approx n\alpha + \sum_{j=1}^{p} \sum_{k=1}^{c} \left[\sum_{i=1}^{n} \sqrt{\beta_j - \alpha} U_{ij} X_{ik} \right]^2$$

$$= n\alpha + \sum_{k=1}^{c} \sum_{j=1}^{p} \left[\sum_{i \in C_k} [\mathbf{r}_i]_j \right]^2$$

$$= n\alpha + \sum_{k=1}^{c} | \mathbf{R}_k |^2$$

C_k is the set of vertices comprising group k

communitity vectors $\mathbf{R}_k = \sum_{i \in C_k} \mathbf{r}_i$

Vector Partitioning (cont.)

(1,1,1,...) is always an eigenvector and the eigenvectors are orthogonal

$$\Rightarrow \quad \sum_{i=1}^{n} [\mathbf{u}_j]_i = \sqrt{n}\mathbf{u}_1^T \mathbf{u}_j = 0$$

$$\Rightarrow \quad \sum_{i=1}^{n} [\mathbf{r}_i]_j = \sqrt{\beta_j - \alpha} \sum_{i=1}^{n} U_{ij} = \sqrt{\beta_j - \alpha} \sum_{i=1}^{n} [u_j]_i = 0$$

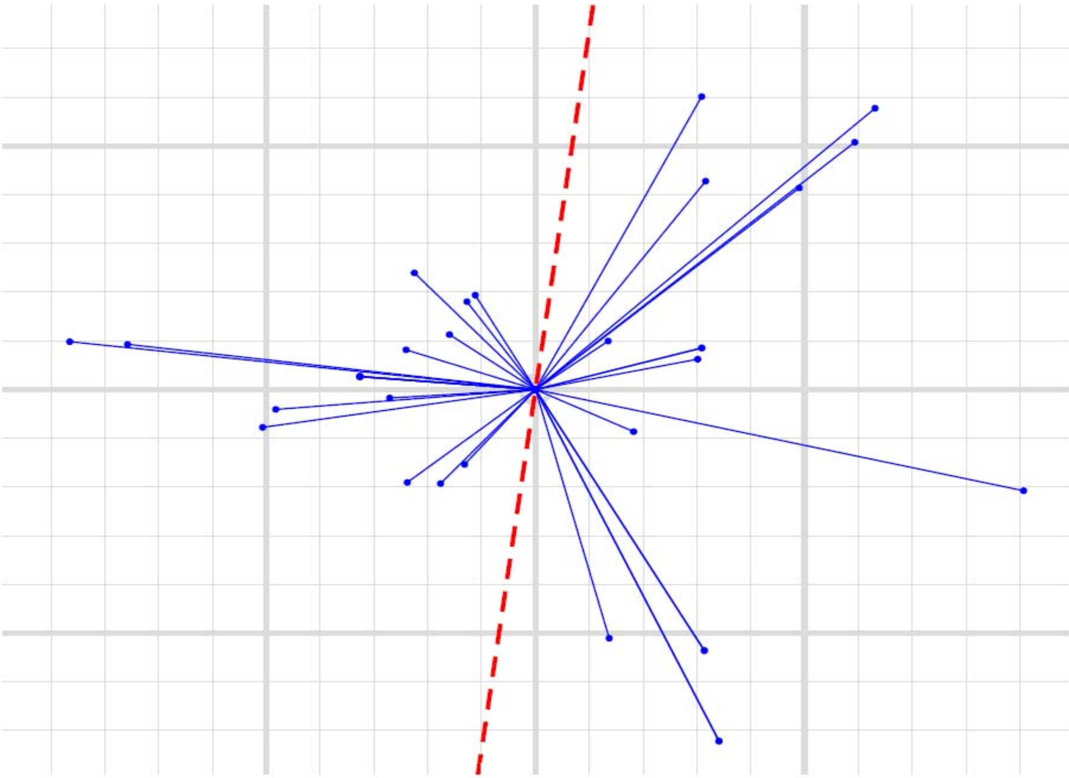
$$\Rightarrow \quad \sum_{i=1}^{n} \mathbf{r}_i = 0$$

$$\Rightarrow \quad \sum_{k=1}^{c} \mathbf{R}_k = \sum_{k=1}^{c} \sum_{i \in C_k} \mathbf{r}_i = \sum_{i=1}^{n} \mathbf{r}_i = 0$$

Vector Partitioning (cont.)

Remove a vertex i from a community k where $\mathbf{R}_k \cdot \mathbf{r}_i < 0$. Then

$$|\mathbf{R}_k - \mathbf{r}_i|^2 - |\mathbf{R}_k|^2 = |\mathbf{r}_i|^2 - 2\mathbf{R}_k \mathbf{r}_i > 0$$



Choice of α

$$\chi^2 = \mathrm{Tr}\left[\mathbf{U}(\mathbf{\Lambda} - \alpha\mathbf{I})\mathbf{U}^T - \mathbf{U}(\mathbf{\Lambda}' - \alpha\mathbf{I}')\mathbf{U}^T\right]^2$$
$$= \mathrm{Tr}\left[(\mathbf{\Lambda} - \alpha\mathbf{I}) - (\mathbf{\Lambda}' - \alpha\mathbf{I}')\right]^2$$
$$= \sum_{i=p+1}^{n} (\beta_i - \alpha)^2$$

Setting the derivative $d\chi^2/d\alpha=0$,

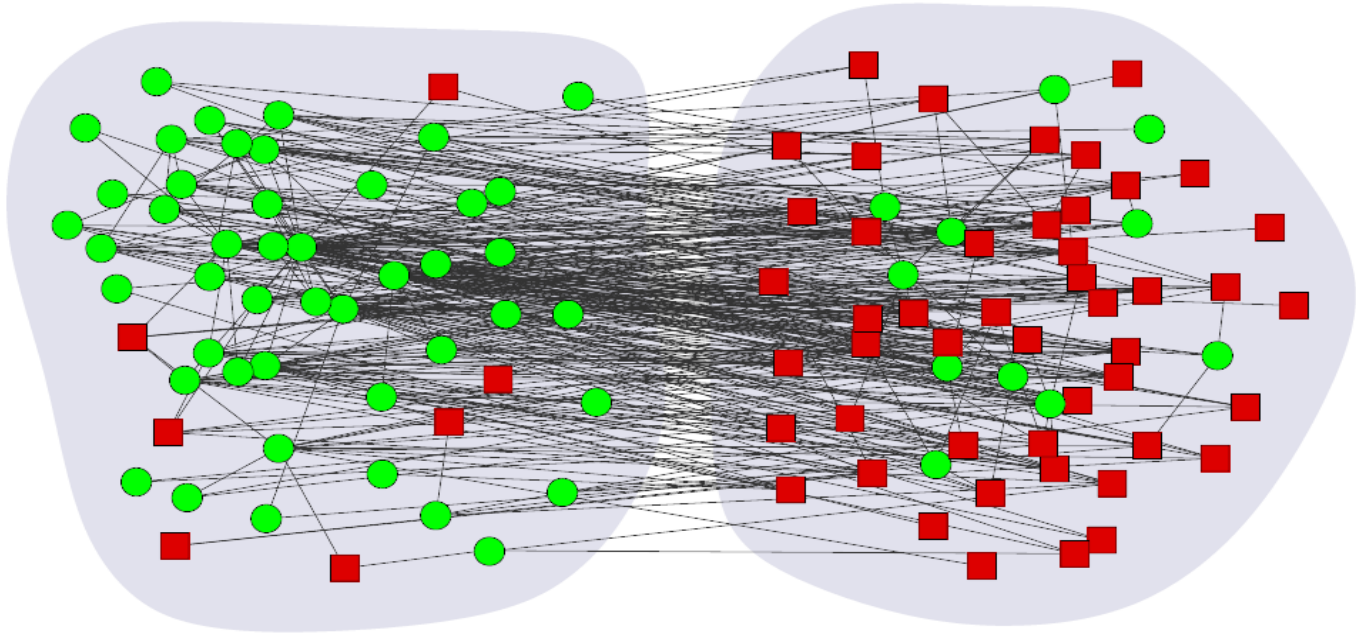
$$\alpha = \frac{1}{n-p}\sum_{i=p+1}^{n}\beta_i$$

Miscellenous

- Iterative subdivision checking criterion

$$\Delta Q = \sum_{i,j \in C} \sum_{k=1}^{c} G_{ij} X_{ik} X_{jk} - \sum_{i,j \in G} G_{ij}$$

- Negative eigenvalues and bipartite structure



Network Correlations

$$
\begin{aligned}
r &= \frac{1}{2m}\mathbf{x}^T\mathbf{G}\mathbf{x} \\
&= \frac{1}{2m}\sum_{ij}\left[W_{ij} - \frac{k_i k_j}{2m}\right]x_i x_j \\
&= \frac{\sum_{ij} W_{ij} x_i x_j}{\sum_{ij} W_{ij}} - \left[\frac{\sum_{ij} W_{ij} x_i}{\sum_{ij} W_{ij}}\right]^2 \\
&= \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle
\end{aligned}
$$

Community Centrality

Suppose there are p positive eigenvalues and q negative ones.

$$[\mathbf{s}_i]_j = \sqrt{\beta_j} U_{ij} \qquad [\mathbf{t}_i]_j = \sqrt{-\beta_{n+1-j}} U_{i,n+1-j}$$

$$Q = \sum_{k=1}^{c} \sum_{j=1}^{p} \left[\sum_{i=1}^{n} \sqrt{\beta_j} U_{ij} X_{ik} \right]^2 - \sum_{k=1}^{c} \sum_{j=1}^{q} \left[\sum_{i=1}^{n} \sqrt{-\beta_{n+1-j}} U_{i,n+1-j} X_{ik} \right]^2$$

$$= \sum_{k=1}^{c} \sum_{j=1}^{p} \left[\sum_{i \in C_k} [\mathbf{s}_i]_j \right]^2 - \sum_{k=1}^{c} \sum_{j=1}^{q} \left[\sum_{i \in C_k} [\mathbf{t}_i]_j \right]^2$$

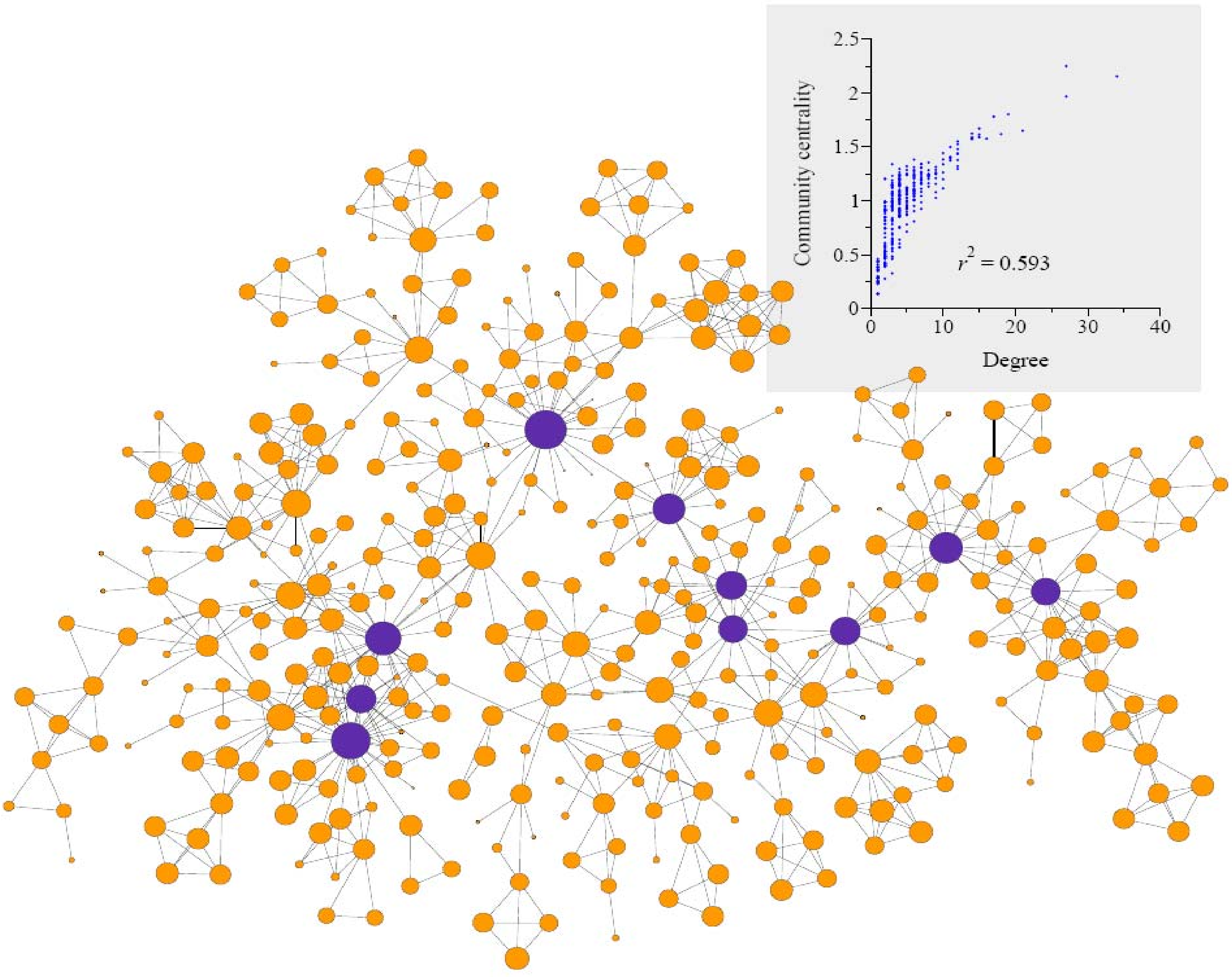
$$= \sum_{k=1}^{c} | \mathbf{S}_k |^2 - \sum_{k=1}^{c} | \mathbf{T}_k |^2$$

$$\mathbf{S}_k = \sum_{i \in C_k} \mathbf{s}_i \qquad \mathbf{T}_k = \sum_{i \in C_k} \mathbf{t}_i$$

Community Centrality (cont.)

$$
\begin{aligned}
&|\mathbf{s}_i|^2 - |\mathbf{t}_i|^2 \\
&= \sum_{j=1}^{p} \left(\sqrt{\beta_j} U_{ij}\right)^2 - \sum_{j=1}^{q} \left(\sqrt{-\beta_{n+1-j}} U_{i,n+1-j}\right)^2 \\
&= \sum_{j=1}^{n} U_{ij} \beta_j U_{ji} \\
&= G_{ii}
\end{aligned}
$$

Community Centrality (cont.)



Discussions

- + rich maths support
- + obtain a good approximate in a single step
- - no scalable
- ? similarity for data clustering
- ? network with power laws and self-similarity