Normalized Cuts and Image Segmentation

Jianbo Shi, Upenn
Jitendra Malik, Berkeley

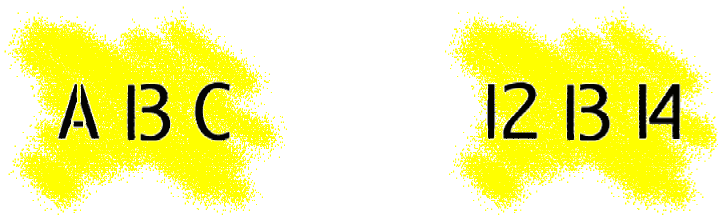
Perceptual Grouping and Organization (Wertheimer)

- Similarity
- Proximity
- Common fate
- Good continuation
- Past experience

=

Visual grouping

History



How to Partition?

- There is no single answer
- Prior World Knowledge:
  - Low Level
    - Coherence of Brightness
    - Color
    - Texture
    - Motion
  - Mid or High Level
    - Symmetries of Objects

How to Partition? (Cont'd)

- inherently hierarchical
- TREE STRUCTURE instead of flat partitioning
- OBJECTIVE:
  - low level cues for hierarchical partitions.
  - High level knowledge for further partitioning.

Tools

- G=(V,E)
- An edge is formed between every pair (Complete graph)
- *w(i,j)* function of similarity.
- Partition V into *disjoint* sets V1, V2,…, Vm
  - Similarity within sets maximum.
  - Similarity between sets minimum.

Graph Vs. Image

- What is the best criterion i.e. the weighting function
- How to compute efficiently
  - The graph is huge

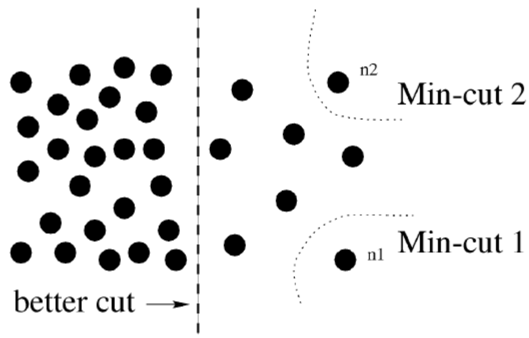
Definitions

- Optimal bipartition is minimum cut
- Minimum cut
  - computed efficiently
  - partitions out small chunks

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$

Example of a Bad Partition

- Similarity decreases as the distance increases



New Definition: Normalized Cut

- Normalize the cut.
- Still a disassociation (dissimilarity) measure.
  - Smaller the better.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$

New Definition: Normalized Association

- Total association (similarity) within groups.
  - The bigger the better.

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

- assoc(a,a): total connectivity within A.

Normalized Cut and Normalized Association Are Related

$$\begin{aligned} Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \\ &= \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} \\ &\quad + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)} \\ &= 2 - \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}\right) \\ &= 2 - Nassoc(A,B). \end{aligned}$$

Formulation for Normal Cut

$$
\begin{aligned}
Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(B,A)}{assoc(B,V)} \\
&= \frac{\sum_{(\boldsymbol{x}_i>0,\boldsymbol{x}_j<0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i>0} \boldsymbol{d}_i} \\
&\quad + \frac{\sum_{(\boldsymbol{x}_i<0,\boldsymbol{x}_j>0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i<0} \boldsymbol{d}_i}.
\end{aligned}
$$

System Linearization

$$4[Ncut(\boldsymbol{x})] = \frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{k\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}}$$

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D}-\mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}}$$

$$\boldsymbol{y} = (\mathbf{1}+\boldsymbol{x}) - b(\mathbf{1}-\boldsymbol{x})$$

$$b = \frac{k}{1-k} = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_{x_i<0} \boldsymbol{d}_i}$$

$$k = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_i \boldsymbol{d}_i}$$

Setting up the matrices

```
    d1 0   0   0 0 0 …… 0
    0   d2 0   0 0 0 …… 0
D=0   0   d3 0 0 0 …… 0
    ……………………………
    0   0   0   0 0 0 …… dN

        0   w12 w13 …. w1N
W= w21    0   w23 …. w2N
      ………………………..
      wN1 wN2 ….. ….   0
```

D-W=

d1	-w12	-w13	…	-w1N
-w21	d2	-w23	…	-w2N
-w31	-w32	d3	…	-w3N
…	…	…	…	…
-wN1	-wN2	-wN3	…	dN

Rayleigh Quotient

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D}-\mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}}$$

- Generalized eigenvalue system
- Minimized if y can be real.

$$(\mathbf{D}-\mathbf{W})y = \lambda \mathbf{D}y$$

- Convert generalized eigensystem to a standard eigensystem:

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}z = \lambda z,$$

- Where

$$z = \mathbf{D}^{\frac{1}{2}}y$$

Results

- D-W is positive semidefinite
  - eigenvectors are perpendicular.
- Second smallest eigenvector minimizes the normalized cut.
  - Rayleigh quotient
- Constraints (to be justified later):
  - y can take on either 1 or –b
  - $y^T \mathbf{D1} = 0$

Problem

- First constraint not satisfied
  - The solution gives real valued eigenvectors.
- Which pixels are above threshold, which ones are not?

Algorithm

- Set up G=(V,E)
  - All the points in the image as nodes
  - Complete graph.
- Weights proportional to the similarity between two pixels.
  - Similarity measure is yet to be decided.

Algorithm (cont'd)

- Solve $(D-W)x=\lambda Dx$ for smallest eigenvectors
  - Method is yet to be decided.
- Second smallest eigenvector bipartitions the graph.
  - Must decide the break point.
- Decide if current partition is good, if not recursively subdivide.

Example: Brightness Image

- Define the weighting function
  - To build matrices D and W:

$$w_{ij} = e^{\frac{-\|F_{(i)} - F_{(j)}\|_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_X^2}} & \text{if } \|X(i) - X(j)\|_2 < r \\ 0 & \text{otherwise.} \end{cases}$$

- Sigma: 10 to 20 percent of the total range of the distance function.

Example (Cont'd)

- Solve for $(\mathbf{D} - \mathbf{W})y = \lambda \mathbf{D} y$.
  - Transform into a standard eigenvalue problem

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}x = \lambda x.$$

- Normally takes O(n^3) operations
  - Use Lanczos method to reduce computation.

Lanczos Method

- Graphs are locally connected, thus sparse
  - Eigensystems are therefore sparse
- Only few smallest eigenvectors needed
- Precision requirement for eigenvectors is loose.

Dividing Point for Eigenvector

- Ideal case: second smallest eigenvector is integer.
- But in reality it takes real values
  - Use the median, or
  - Use 0, or
  - Use the value that makes Ncut(A,B) smallest.

Stability Criterion

- Sometimes an eigenvector takes on continues values.
- Not suitable for partitioning purpose
  - Hard to find a cut point
    - Similar Ncut values.
- Compute the histogram
  - ratio between the minimum and maximum values in the bins.

Drawbacks of 2-Way Cut

- Stability criterion is good
  - but it gets rid of subsequent eigenvectors
- Recursive 2-way cut is inefficient
  - uses only the second smallest eigenvector

Simultaneous K-Way Cut with Multiple Eigenvectors

- k-means is used for oversegmentation.
- Second step can vary
  - Greedy Pruning
  - Global Recursive Cut

Greedy Pruning

- Iteratively merge two segments until k segments are left.
- Each iteration choose the pair when merged minimizes the k-way Ncut criterion:

$$Ncut_k = \frac{cut(\mathbf{A}_1, \mathbf{V} - \mathbf{A}_1)}{assoc(\mathbf{A}_1, \mathbf{V})} + \frac{cut(\mathbf{A}_2, \mathbf{V} - \mathbf{A}_2)}{assoc(\mathbf{A}_2, \mathbf{V})} + \dots + \frac{cut(\mathbf{A}_k, \mathbf{A} - \mathbf{A}_k)}{assoc(\mathbf{A}_k, \mathbf{V})},$$

Global Recursive Cut

- Build a condensed graph
- Each node of the graph corresponds to one initial segment
- Weights Defined as the total connection from one initial partition to another
- Use exhaustive search or generalized eigenvalue system for solution.

Experiments

- Brightness
- Color
- Texture
- Motion
- Following equation always holds; it is F(i) that changes according to the application

Similarity * Spatial Proximity

$$w_{ij} = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_X}} & \text{if } \|X(i)-X(j)\|_2 < r \\ 0 & \text{otherwise,} \end{cases}$$

- F(i)=
  - 1, when segmenting point sets
  - I(i), the intensity value for segmenting scalar images
  - [v,v.s.sin(h), v.s.cos(h)](i) for color images
  - [|I*f1|, … , |I*fn|](i), for texture.
- The weight is always zero if proximity criterion is not met, i.e. The pixels are more than r pixels apart.

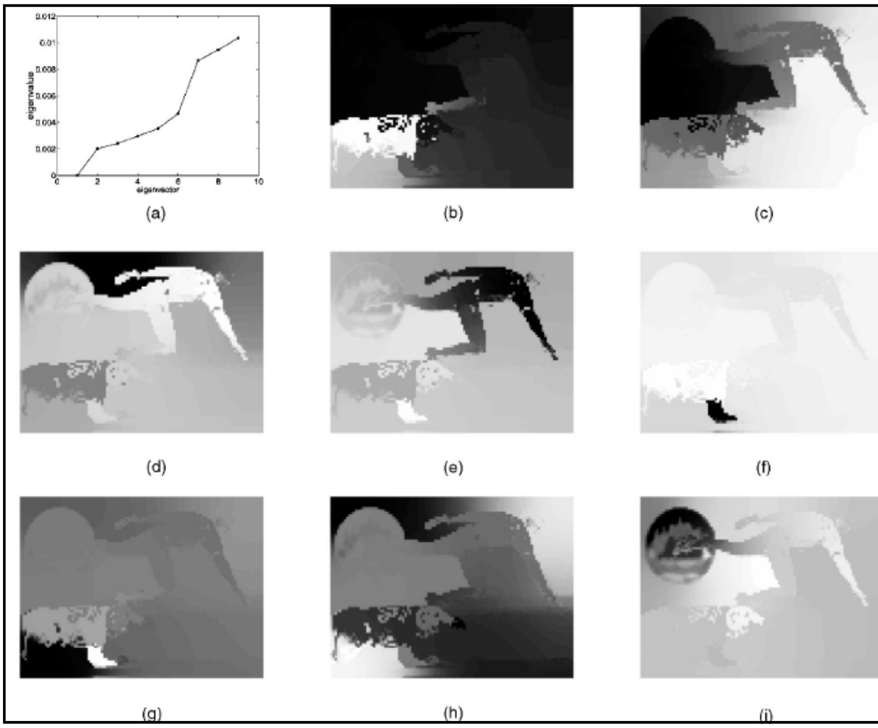
MOTION

- Spatiotemporal neighborhood

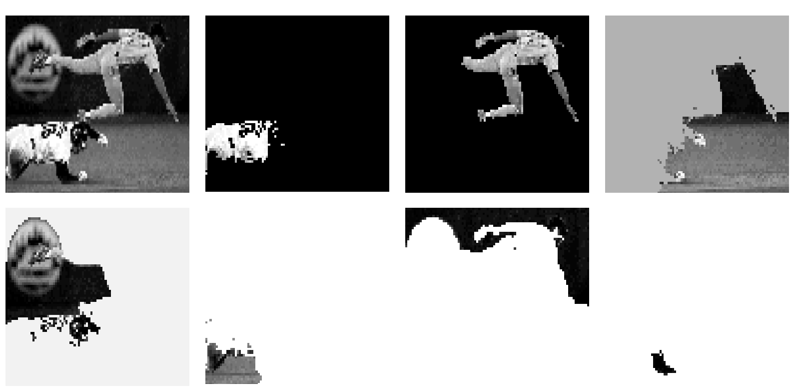
$$w_{ij} = \begin{cases} e^{\frac{-\mathbf{d}_m(i,j)^2}{\sigma_m^2}} & \text{if } \|X(i)-X(j)\|_2 < r \\ 0 & \text{otherwise,} \end{cases}$$

- Dm: motion distance

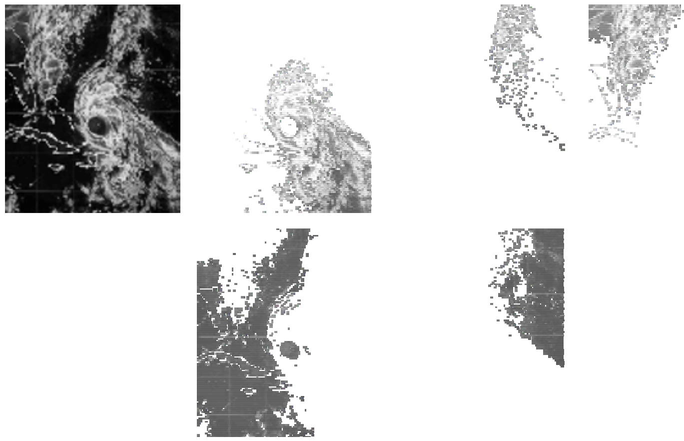
$$\mathbf{d}(i,j) = 1 - \sum_{\mathbf{dx}} P_i(\mathbf{dx}) P_j(\mathbf{dx})$$

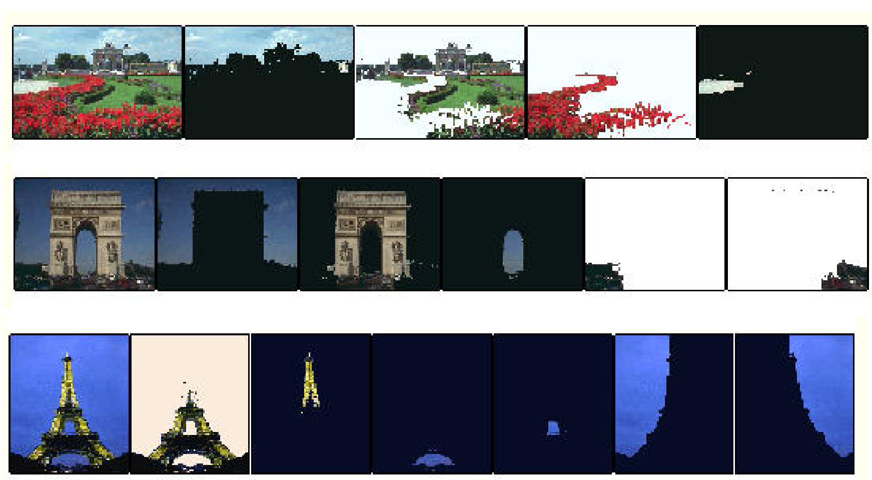


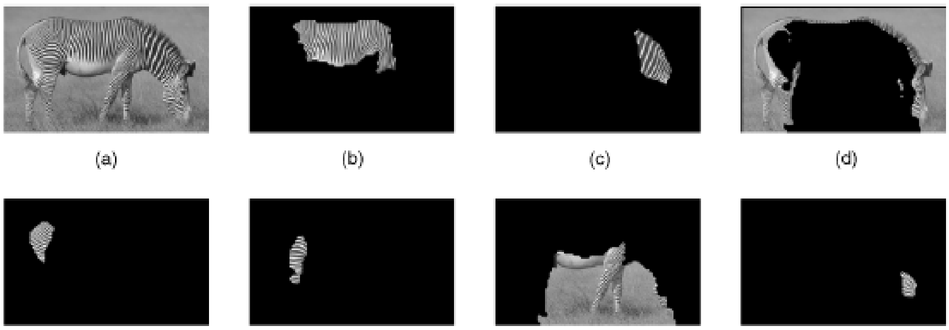
Experimental Results



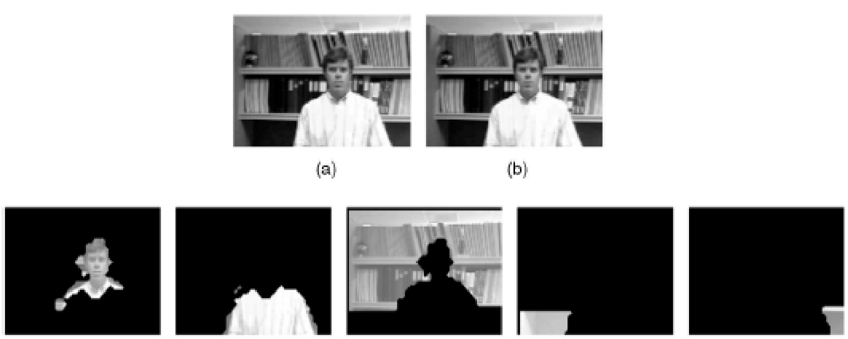
Experimental Results





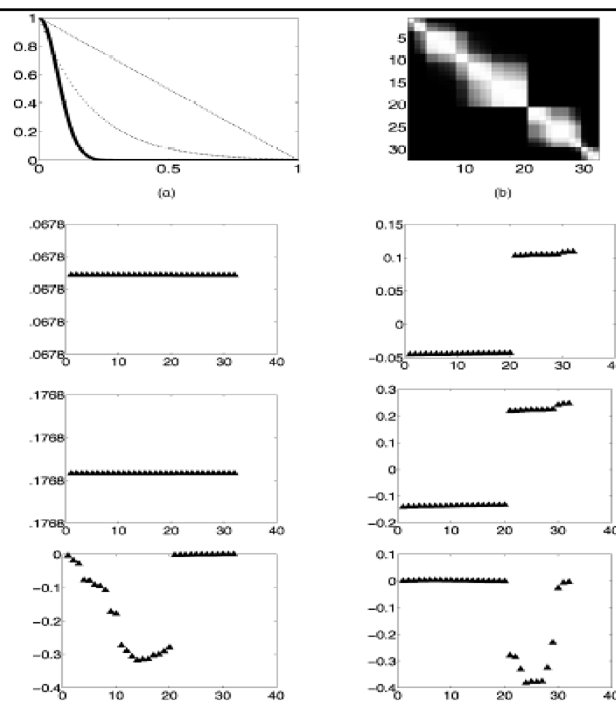

(a) (b) (c) (d)
(e) (f) (g) (h)

Motion


(a) (b)
(c) (d) (e) (f) (g)

Finding clumps ⟷ Finding splits

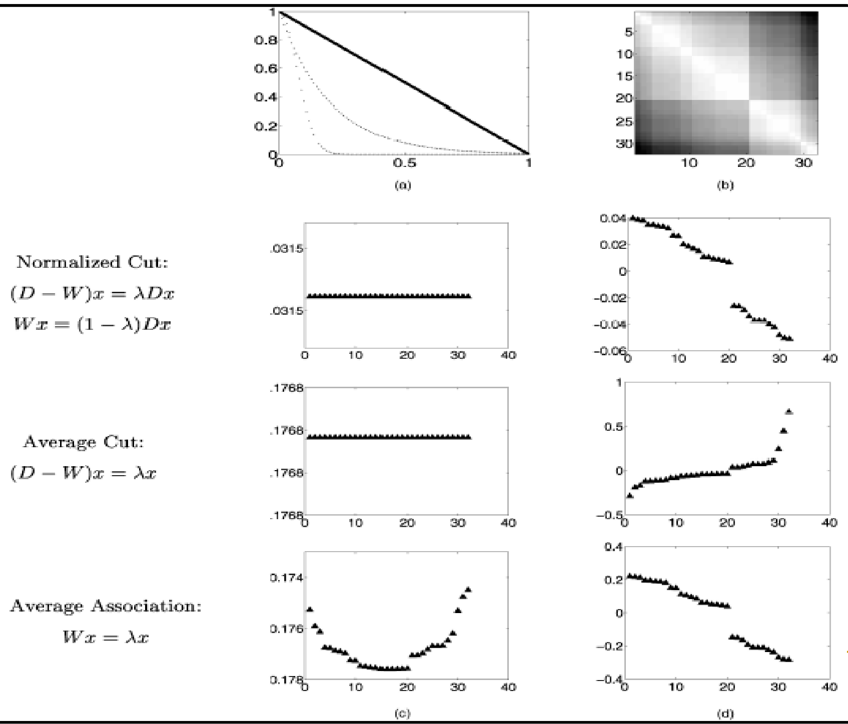
	Average association	Normalized Cut	Average cut
Discrete formulation	$\frac{asso(A,A)}{\lvert A\rvert} + \frac{asso(B,B)}{\lvert B\rvert}$	$\frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$ or $2 - (\frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)})$	$\frac{cut(A,B)}{\lvert A\rvert} + \frac{cut(A,B)}{\lvert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



Normalized Cut:
$(D-W)x = \lambda D x$
$Wx = (1-\lambda) D x$

Average Cut:
$(D-W)x = \lambda x$

Average Association:
$Wx = \lambda x$



Linearize the System

$$
\begin{aligned}
Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(B,A)}{assoc(B,V)} \\
&= \frac{\sum_{(\boldsymbol{x}_i>0,\boldsymbol{x}_j<0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i>0} \boldsymbol{d}_i} \\
&\quad + \frac{\sum_{(\boldsymbol{x}_i<0,\boldsymbol{x}_j>0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i<0} \boldsymbol{d}_i}.
\end{aligned}
$$

- Xi > 0 if node i belongs to A, Xi<0 otherwise.
- $\boldsymbol{d}(i) = \sum_j w(i,j)$
- $\boldsymbol{d}(i)$ is the connection from node i to all nodes.

Linearize the System (2)

- The equation linearizes to:

$$\frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{k\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}}$$

- D is a diagonal NxN matrix with d on its diagonal, W is the NxN weight matrix where w(i,j) is the weight of the edge between node i and j.
- 1 is a vector of all ones.

$$k = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_i \boldsymbol{d}_i}$$

Linearize the System (3)

$$\frac{(\alpha(\boldsymbol{x})+\gamma)+2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} - \frac{2(\alpha(\boldsymbol{x})+\gamma)}{M} + \frac{2\alpha(\boldsymbol{x})}{M} + \frac{2\gamma}{M}.$$

Let

$$\alpha(\boldsymbol{x}) = \boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x},$$
$$\beta(\boldsymbol{x}) = \mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x},$$
$$\gamma = \mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1},$$

and

$$M = \mathbf{1}^T\mathbf{D}\mathbf{1},$$

Linearize the System (4)

$$= \frac{(1-2k+2k^2)(\alpha(\boldsymbol{x})+\gamma)+2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} + \frac{2\alpha(\boldsymbol{x})}{M}$$

$$= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(\boldsymbol{x})+\gamma)+\frac{2(1-2k)}{(1-k)^2}\beta(\boldsymbol{x})}{\frac{k}{1-k}M} + \frac{2\alpha(\boldsymbol{x})}{M}.$$

Letting $b = \frac{k}{1-k}$, and since $\gamma = 0$, it becomes

$$= \frac{(1+b^2)(\alpha(\boldsymbol{x})+\gamma)+2(1-b^2)\beta(\boldsymbol{x})}{bM} + \frac{2b\alpha(\boldsymbol{x})}{bM}$$

$$= \frac{(1+b^2)(\alpha(\boldsymbol{x})+\gamma)}{bM} + \frac{2(1-b^2)\beta(\boldsymbol{x})}{bM} + \frac{2b\alpha(\boldsymbol{x})}{bM} - \frac{2b\gamma}{bM}$$

Linearize the System (5)

$$= \frac{(1+b^2)(\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}+\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{2(1-b^2)\mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{2b\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} - \frac{2b\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}}$$

$$= \frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{b^2(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} - \frac{2b(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{b\mathbf{1}^T\mathbf{D}\mathbf{1}}$$

$$= \frac{[(\mathbf{1}+\boldsymbol{x})-b(\mathbf{1}-\boldsymbol{x})]^T(\mathbf{D}-\mathbf{W})[(\mathbf{1}+\boldsymbol{x})-b(\mathbf{1}-\boldsymbol{x})]}{b\mathbf{1}^T\mathbf{D}\mathbf{1}}.$$

Setting $\boldsymbol{y} = (\mathbf{1} + \boldsymbol{x}) - b(\mathbf{1} - \boldsymbol{x})$, it is easy to see that

$$\boldsymbol{y}^T \mathbf{D1} = \sum_{x_i > 0} \boldsymbol{d}_i - b \sum_{x_i < 0} \boldsymbol{d}_i = 0$$

since $b = \frac{k}{1-k} = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_{x_i<0} \boldsymbol{d}_i}$ and

$$\begin{aligned}
\boldsymbol{y}^T \mathbf{D} \boldsymbol{y} &= \sum_{x_i > 0} \boldsymbol{d}_i + b^2 \sum_{x_i < 0} \boldsymbol{d}_i \\
&= b \sum_{x_i < 0} \boldsymbol{d}_i + b^2 \sum_{x_i < 0} \boldsymbol{d}_i \\
&= b\left(\sum_{x_i < 0} \boldsymbol{d}_i + b \sum_{x_i < 0} \boldsymbol{d}_i\right) \\
&= b \mathbf{1}^T \mathbf{D1}.
\end{aligned}$$

Putting everything together we have,

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T (\mathbf{D} - \mathbf{W}) \boldsymbol{y}}{\boldsymbol{y}^T \mathbf{D} \boldsymbol{y}},$$

with the condition $\boldsymbol{y}(i) \in \{1, -b\}$ and $\boldsymbol{y}^T \mathbf{D1} = 0$.