Monday April 26, 1999 Notes for CS280

Last Time – image segmentation, normalized cut:

$$NCut(A,B) = \frac{y^T(D-W)y}{y^T Dy} \qquad y_i \in \{1,-b\},\ y^T D1 = 0$$

The eigenvector with the second smallest eigenvalue of the generalized eigensystem:

$$(D-W)y = \lambda Dy$$

is the solution to the constrained Raleigh quotient:

$$y_1 = \arg.\min._{y^T D1=0} \frac{y^T(D-W)y}{y^T Dy}$$

How are W and D obtained:

W_{ij} is associated with the edges of the graph of pixels, it represents the 'connection' of a pixel i with a pixel j, when i=j, $W_{ij}=1$

W_{ij} is close to 1 if pixels i and j are 'similar'

W_{ij} is close to 0 if pixels i and j are not 'similar'

W is the matrix made up of the W_{ij} components. D is the diagonal matrix calculated from W in the following manner:

$$D_{ii} = \sum_j W_{ij}$$

New Material – Aside, Generalized eigenvalue problem.

$Ay = \lambda y$	$Ay = \lambda By$
Standard eig val problem	Generalized eig val problem

The following short proof demonstrates that these two problem are in fact mathematically equivalent (Normalized cut eigenvalue problem is used as example):

$$(D-W)y = \lambda Dy$$

$$D^{-\frac{1}{2}}(D-W)y = \lambda D^{-\frac{1}{2}} Dy$$

$$D^{-\frac{1}{2}}(D-W)y = \lambda D^{\frac{1}{2}} y$$

defining new variable z:

$$z = D^{1/2} y$$

Allows us to rewrite the generalized eigenvalue problem as the standard eigenvalue problem:

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

Image partitioning – Simple example using the distance metric

Let there be eight pixels in two clusters:

….   ….
1-4   5-8   <= Pixel index

The weights between pixels will be defined on proximity only:

$$W_{ij} = e^{-d_{ij}^2}$$

with d_{ij} being the distance between pixel i and j.
Once this criterion is established the W matrix is easily constructed, in the case of this example W will be a matrix of the following form:

$$W = \begin{bmatrix} A & B \\ B & A \end{bmatrix} \qquad A = \begin{bmatrix} 1 & \sim 1 & \sim 1 & \sim 1 \\ \sim 1 & 1 & \sim 1 & \sim 1 \\ \sim 1 & \sim 1 & 1 & \sim 1 \\ \sim 1 & \sim 1 & \sim 1 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \sim 0 & \sim 0 & \sim 0 & \sim 0 \\ \sim 0 & \sim 0 & \sim 0 & \sim 0 \\ \sim 0 & \sim 0 & \sim 0 & \sim 0 \\ \sim 0 & \sim 0 & \sim 0 & \sim 0 \end{bmatrix}$$

The matrix D can be easily obtained from W, and the second eigenvector to the appropriate eigensystem will have a bimodal distribution. Each pixels will have a corresponding component of the eigenvector similar in value to other corresponding components of other pixels nearby, but dissimilar to pixels far away.
This example has been assigned as a homework problem.

General Comments on form of the solution to NC eigensystem

0) In matlab eigenvectors are ordered from the smallest to largest eigenvalue.
1) First eigenvector is all ones, with a corresponding eigenvalue of zero.
2) All eigenvalues are non-negative, in the range 0-2

3) Second eigenvector is the important one used for segmentation, ideally it is bimodal with a small eigenvalue.
4) Later eigenvectors split the image in less and less significant groups, the image is always split in two.
5) Each eigenvector divides the image into two groups, if no such division is possible the eigenvector may not be bimodal, and the eigenvalue will be large
6) The magnitude of the eigenvalue corresponding to the eigenvector is the metric for how well the two groups divide, the smaller the eigenvalue the cleaner the division. Typical cut off for a significant group is lambda<0.05.

The NC method splits the image into two groups for each eigenvector past the first one, in general there are two methods for splitting the image into multiple groups:

1) Recursive Bisection- split the image on the second eigenvector into two parts, treat each part as a new image, repeat split recursively. Stop each branch when the second eigenvalue exceeds some threshold.
2) Multiple Eigenvector- Use all eigenvectors with a sufficiently small eigenvalue to generate divisions in the image use all the division together to generate image groups.

Intuitive analogue – Physical picture

(D-W) – is equivalent to the stiffness matrix **K**
D - is equivalent to the mass matrix **M**

The NC e-system problem is equivalent (up to a minus sign) to the mass spring system:

$$\mathbf{M}\ddot{x} = -\mathbf{K}x_d$$

or

$$M_i\ddot{x}_i = \sum_j -k_{ij}(x_i - x_j)$$

Where M is the mass matrix, K the spring constant matrix and x_d the vector of displacements from equilibrium.

This can be seen easily by inserting D_i as the mass and W_{ij} as the spring constant k_{ij}.

$$D_i\ddot{x}_i = \sum -W_{ij}(x_i - x_j)$$

$$D_i\ddot{x}_i = -D_i x_i + \sum_j W_{ij}x_j$$

$$\mathbf{D}\ddot{x} = -(\mathbf{D} - \mathbf{W})x$$

The generalized eigenvectors are the fundamental modes of oscillation. This is because when the system of springs is in an eigenvector the acceleration, given by the above equation, of each mass point is proportional to its displacement (x), with the constant of proportionality being the eigenvalue of the mode. Such a state implies that each point oscillates as a simple harmonic oscillator with the frequency which is the square root of the eigenvalue. Since the each point is vibrating at the same frequency the vibration of the entire system is stable and predictable, hence it is termed to be in a fundamental mode of vibration.

General Comments about NC procedure

1) This method is good at tolerating noise, better then local methods because it uses the entire image for segmentation.
2) It is very important to use effective, carefully chosen W_{ij} , the success of this method primarily depends on this choice.
3) The W_{ij} should span the entire dynamic range from 0 to 1.
4) NC procedure is applicable to other clustering problems not only image segmentation.

Computability Considerations

Consider an image 100x100 pixels, implies 10e4 nodes in the NC graph, or 10e8 edges. This implies W sizes in gigabytes range, and long processing time since most eigenvalue algorithms are N^2 or N^3.

The problem can be simplified by setting most of the W matrix to 0, and making W sparse.

1) Setting the connections between far removed pixels to 0, decreases the nonzero entries in W from N^2x N^2 to C* N^2, for an image with N^2 pixels. Where C is the number of nearest neighbor pixels taken into account.
2) This approach makes W a block diagonal matrix, with significant values only close to the diagonal.
3) Fast algorithms such as the Lanczos algorithm are available for evaluating sparse matrices.

Overall NC Procedure

1) Construct a weighted graph from image.
2) Connect each pair of pixels, and assign graph edge weight, W_{ij}, based on the probability that pixel i and j are in the same region.
3) Solve $(D-W)y = \lambda Dy$ for the second eigenvector.
4) Each eigenvector with a small eigenvalue will split the image into two parts, pixels in one part will have one cluster of eigenvector components, the second part will have another.
5) Multiple regions can be achieved by using multiple eigenvectors, or recursively dividing the image.

More General NC approach comments

1) Global decision good, local bad, due to noise.
2) Efficient computation, draws on ideas from spectral graph theory to define an eigenvalue problem which can be solved to achieve segmentation.
3) W matrix must incorporate visual cues to effectively segment the image.

Cue Integration

Visual cues can be integrated into the W matrix:

$$W_{ij}= W_{ij}^{texture}\ W_{ij}^{contour}\ W_{ij}^{intensity}\ W_{ij}^{proximity}$$

$W_{ij}^{texture}$ is the probability that the pixels i and j have the same texture. Usually based texton histograms and measured with the chi-squared metric.
$W_{ij}^{contour}$ is the probability that pixels i and j are bound by the same contour.
$W_{ij}^{intensity}$ is a gaussian in the intensity difference between i and j.
$W_{ij}^{proximity}$ is a gaussian in the distance between i and j.

Note that some cues must be turned off at some portions of the image. For example it does not make sense to use the intensity cue when in a textured region.

Texture Cue

1) Uses elongated directions gaussian derivatives (1^{st} and 2^{nd}).
2) Creates a histogram of filter outputs.
3) Compares each region using the chi-squared metric.

Example Images

Penguin-

1) The image segments the penguin into the white an black parts.
2) The algorithm successfully separates the gravel/ice ground using a stochastic texture cue.

Woman with dotted shirt-

1) The algorithm successfully segments the shirt, head-neck-arm, and belt.
2) Texture window size can cause spill over of the textured region into other parts of the image if it is too large.

Tiger –

1) The algorithm successfully segments the tiger, using the texture cue.
2) The water above and below the tiger is segmented into different groups.