Graph Laplacians - Definition

- "Laplace-type" matrices associated with general undirected graphs – useful in many applications
- Given a graph $G = (V, E)$ define
  - A matrix $W$ of weights $w_{ij}$ for each edge
  - Assume $w_{ij} \geq 0$, $w_{ii} = 0$, and $w_{ij} = w_{ji} \forall (i, j)$
  - The diagonal matrix $D = \text{diag}(d_i)$ with $d_i = \sum_{j \neq i} w_{ij}$
- Corresponding graph Laplacian of $G$ is: $L = D - W$
- Gershgorin’s theorem $\rightarrow L$ is positive semidefinite

Simplest case:

$w_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \& i \neq j \\ 0 & \text{else} \end{cases}$

$D = \text{diag} \left[ d_i = \sum_{j \neq i} w_{ij} \right]$

- Define the graph Laplacian for the graph associated with the simple mesh shown next. [use the simple weights of 0 or 1]
A few properties of graph Laplacians

Let $L = D - W$ with $D = \text{diag}(d_i)$ and $w_{ij} \geq 0$, $d_i = \sum_{j \neq i} w_{ij}$

Property 1: for any $x \in \mathbb{R}^n$:

$$x^T L x = \frac{1}{2} \sum_{i,j} w_{ij} |x_i - x_j|^2$$

Property 2: (generalization) for any $Y \in \mathbb{R}^{d \times n}$:

$$\text{Tr} [YLY^T] = \frac{1}{2} \sum_{i,j} w_{ij} \|y_i - y_j\|^2$$

What is the difference with the discretization of the Laplace operator in 2-D for case when mesh is the same as this graph?
Property 3: For the particular $L = I - \frac{1}{n}11^\top$

$$XLX^\top = \bar{X}\bar{X}^\top = n \times \text{Covariance matrix}$$

Property 4: $L$ is singular and admits the null vector $e = \text{ones}(n,1)$

Property 5: (Graph partitioning) Consider situation when $w_{ij} \in \{0, 1\}$. If $x$ is a vector of signs ($\pm 1$) then

$$x^\top LX = 4 \times \text{('number of edge cuts')}$$

edge-cut = pair $(i, j)$ with $x_i \neq x_j$

Would like to minimize $(Lx, x)$ subject to $x \in \{-1, 1\}^n$ and $e^\top x = 0$ [balanced sets]

Will solve a relaxed form of this problem

Consider any symmetric (real) matrix $A$ with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and eigenvectors $u_1, \ldots, u_n$

Recall that:

**(Min reached for $x = u_1$)**

$$\min_{x \in \mathbb{R}^n} \frac{(Ax, x)}{(x, x)} = \lambda_1$$

In addition:

**(Min reached for $x = u_2$)**

$$\min_{x \perp u_1} \frac{(Ax, x)}{(x, x)} = \lambda_2$$

For a graph Laplacian $u_1 = e = \text{vector of all ones and}$

...vector $u_2$ is called the Fiedler vector. It solves a relaxed form of the problem -

Define $v = u_2$ then $\text{lab} = \text{sign}(v - \text{med}(v))$

Partition graph in two using fiedler vectors

Cut largest in two ..

Repeat until number of desired partitions is reached

Use the Lanczos algorithm to compute the Fiedler vector at each step
Let $N$ be the incidence matrix: $N_{ij} = \pm 1$ if $i$-th edge is incident on the $j$-th vertex.

For example: $A \leftrightarrow C,D, B \leftrightarrow D, C \leftrightarrow A, D \leftrightarrow A,B$ (undirected graph):

\[
N = \begin{pmatrix}
1 & 0 & -1 & 0 \\
1 & 0 & 0 & -1 \\
0 & -1 & 0 & 1 \\
0 & -1 & 0 & 2
\end{pmatrix},
\]

yielding Laplacian = diagonal matrix of degrees — Adjacency matrix :

\[
N^T N = L = \begin{pmatrix}
2 & 0 & -1 & -1 \\
0 & 1 & 0 & -1 \\
-1 & 0 & 1 & 0 \\
-1 & -1 & 0 & 2
\end{pmatrix}.
\]

Mark a partitioning of the vertices: $n_- = 1, n_+ = 3$

$v = [1,1,1,-3]^T\sqrt{3} = [n_-, n_, n_, -n_+]^T\sqrt{n_-n_+}$. Then

\[
v^T L v = |\text{cut}| \cdot \left( \frac{1}{n_-} + \frac{1}{n_+} \right)
\]

and:

\[
v^T e = 0, \text{ where } e = [1,1,1,1]^T \text{ is eigenvector of } L.
\]

Approximately minimize this with an eigenvector of $L$:

\[-1.E-15 (.500000 .500000 .500000 .500000) \leftarrow \text{null vector}
\]

\[.585786 (-.27059 .653281 -.65328 .270598) \leftarrow \text{Fiedler}
\]

\[2.00000 (.500000 -.50000 -.50000 .500000) \text{ vector}
\]

\[3.41421 (.653281 .270598 -.27059 -.65328) \text{ vector}
\]

Let 1 amp current is applied between nodes 1 and $n$.

Assume unit resistances on every link.

What is the voltage drop?

Let $v =$ vector of voltage levels at each node.

Ohm’s Law: $N v = i =$ currents across every link.

Kirchoff’s Law: $N^T i = b$, where $b = (1, 0, \ldots, 0, -1)^T$.

Solve $N^T N v = b$ for voltages. Use $L = N^T N$.

Try $v = L^b$

Show $N^T N = L$ and $L v = b$.

Voltage drop from 1 to $n$ is proportional to the average commute time for a random walk from 1 to $n$ and back. This is a square of a metric distance between nodes.

Application: Spectral Graph Partitioning

Mark a partitioning of the vertices: $n_- = 1, n_+ = 3$

$v = [1,1,1,-3]^T\sqrt{3} = [n_-, n_, n_, -n_+]^T\sqrt{n_-n_+}$. Then

\[
v^T L v = |\text{cut}| \cdot \left( \frac{1}{n_-} + \frac{1}{n_+} \right)
\]

and:

\[
v^T e = 0, \text{ where } e = [1,1,1,1]^T \text{ is eigenvector of } L.
\]

Approximately minimize this with an eigenvector of $L$:

\[-1.E-15 (.500000 .500000 .500000 .500000) \leftarrow \text{null vector}
\]

\[.585786 (-.27059 .653281 -.65328 .270598) \leftarrow \text{Fiedler}
\]

\[2.00000 (.500000 -.50000 -.50000 .500000) \text{ vector}
\]

\[3.41421 (.653281 .270598 -.27059 -.65328) \text{ vector}
\]

Application: Google’s Page rank

Idea is to put order into the web by ranking pages by their importance.

Install the google-toolbar on your laptop or computer

http://toolbar.google.com/

Tells you how important a page is...

Google uses this for searches..

Updated regularly..

Still a lot of mystery in what is in it..
**Page-rank - explained**

**Main point:** A page is important if it is pointed to by other important pages.

- Importance of your page (its PageRank) is determined by summing the page ranks of all pages which point to it.
- Weighting: If a page points to several other pages, then the weighting should be distributed proportionally.
- Imagine many tokens doing a random walk on this graph:
  - \((\delta/n)\) chance to follow one of the \(n\) links on a page,
  - \((1 − \delta)\) chance to jump to a random page.
- What's the chance a token will land on each page?
- If www.cs.umn.edu/~boley points to 10 pages including yours, then you will get \(1/10\) of the credit of my page.

**Page-Rank - definitions**

If \(T_1, \ldots, T_n\) point to page \(T_i\) then

\[
\rho(T_i) = 1 - \delta + \delta \left[ \frac{\rho(T_1)}{|T_1|} + \frac{\rho(T_2)}{|T_2|} + \cdots + \frac{\rho(T_n)}{|T_n|} \right]
\]

- \(|T_j|\) = count of links going out of Page \(T_i\). So the 'vote' \(\rho(T_j)\) is spread evenly among \(|T_j|\) links.
- Sum of all PageRanks == 1: \(\sum_T \rho(T) = 1\)
- \(\delta\) is a 'damping' parameter close to 1 – e.g. 0.85
- Defines a (possibly huge) Hyperlink matrix \(H\):

\[
h_{ij} = \begin{cases} 
\frac{1}{|T_i|} & \text{if } i \text{ points to } j \\
0 & \text{otherwise}
\end{cases}
\]

- Row- sums of \(H\) are \(= 1\).
- Sum of all PageRanks will be one:
  \[
  \sum_{\text{All-Pages}} \rho(A) = 1.
  \]
- \(H\) is a stochastic matrix [actually it is forced to be by changing zero rows]
Algorithm (PageRank)

1. Select initial row vector \( v \) (\( v \geq 0 \))
2. For \( i=1:\text{maxitr} \)
3. \( v := (1 - \delta)e^T + \delta v H \)
4. end

Do a few steps of this algorithm for previous example with \( \delta = 0.85 \).

This is a row iteration.

\[
v = (1 - \delta)e^T + v \cdot \delta H
\]

Kleinberg’s Hubs and Authorities

Idea is to put order into the web by ranking pages by their degree of Authority or "Hubness".

• An Authority is a page pointed to by many important pages.
  • Authority Weight = sum of Hub Weights from In-Links.

• A Hub is a page that points to many important pages:
  • Hub Weight = sum of Authority Weights from Out-Links.

Source:

Computation of Hubs and Authorities

Simplify computation by forcing sum of squares of weights to be 1.

\[
\text{Auth}_j = x_j = \sum_{i: (i,j) \in \text{Edges}} \text{Hub}_i.
\]

\[
\text{Hub}_i = y_i = \sum_{j: (i,j) \in \text{Edges}} \text{Auth}_j.
\]

Let \( A = \text{Adjacency matrix: } a_{ij} = 1 \text{ if } (i, j) \in \text{Edges} \).

\[
y = Ax, x = A^T y.
\]

Iterate ... to leading eigenvectors of \( A^T A \) & \( AA^T \).

Answer: Leading Singular Vectors!