Graphs

non-planar embedding

planar embedding
Layout of a Large Graph

- “Happy Buddha” 3-D mesh model
- 50,000 nodes
- 100,000 faces
- Can be cut (with six cuts) into a simply connected graph
- Can we find a planar layout of this graph?
Graph Embedding


• Input is a graph $G$ consisting of nodes (1, ..., $N$) and edges $(i,j)$ ($1 \leq i < j \leq N$)
• Identify some of the nodes as \textit{boundary nodes} and assign them 2-D positions
• In this example, we have nodes (1,2,3,4,5,6,7,8) and edges (1,2), (1,3), (2,4), (3,4), (1,5), (2,6), (3,7), (4,8), (5,6), (5,7), (6,8) and (7,8)
• Identify nodes 1,2,3 and 4 as boundary nodes, and assign them coordinates (0,0), (1,0), (0,1) and (1,1)
Graph Embedding

- Create the graph Laplacian matrix
- Adjacency matrix with elements $L_{ij} = 1/\text{deg}(i)$ for an edge between node i and node j

$$L = \begin{bmatrix}
0 & 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 \\
1/3 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\
1/3 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 \\
0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 1/3 \\
1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\
0 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 1/3 \\
0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 1/3 \\
0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 & 0
\end{bmatrix}$$
Graph Embedding

- Zero out the rows for the nodes we have already positioned
- Subtract it from the identity matrix

\[
A = I - L = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-\frac{1}{3} & 0 & 0 & 0 & 1 & -\frac{1}{3} & -\frac{1}{3} & 0 \\
0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\
0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 1 & -\frac{1}{3} \\
0 & 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 1
\end{bmatrix}
\]
Graph Embedding

- Create linear systems of equations
- Solve $Ax = b_x$ for the $x$ coordinates
Graph Embedding

- Create linear systems of equations
- Solve $Ax = b_x$ for the $x$ coordinates
- Solve $Ay = b_y$ for the $y$ coordinates

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & 0 & 0 & 1 & -\frac{1}{3} & -\frac{1}{3} & 0 \\
0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\
0 & -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} & 0 \\
0 & 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 1 \\
0 & 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8 \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
1 \\
1 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]
Graph Embedding

- Create linear systems of equations
- Solve $Ax = b_x$ for the $x$ coordinates
- Solve $Ay = b_y$ for the $y$ coordinates

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-\frac{1}{3} & 0 & 0 & 0 & 1 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\
0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\
0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\
0 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
1 \\
0 \\
1 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

$x_1 = 0, \quad x_2 = 1$

$x_3 = 0, \quad x_4 = 1$

$x_5 = (x_1 + x_6 + x_7)/3$

$x_6 = (x_2 + x_5 + x_8)/3$

$x_7 = (x_3 + x_5 + x_8)/3$

$x_8 = (x_4 + x_6 + x_7)/3$
Graph Embedding

- Create linear systems of equations
- Solve \( Ax = b_x \) for the x coordinates
- Solve \( Ay = b_y \) for the y coordinates

\[
\begin{align*}
x_1, y_1 &= (0,0), \quad x_2, y_2 = (1,0) \\
x_3, y_3 &= (0,1), \quad x_4, y_4 = (1,1) \\
x_5, y_5 &= ((x_1, y_1) + (x_6, y_6) + (x_7, y_7))/3 \\
x_6, y_6 &= ((x_2, y_2) + (x_5, y_5) + (x_8, y_8))/3 \\
x_7, y_7 &= ((x_3, y_3) + (x_5, y_5) + (x_8, y_8))/3 \\
x_8, y_8 &= ((x_4, y_4) + (x_6, y_6) + (x_7, y_7))/3
\end{align*}
\]
Graph Embedding

- Create linear systems of equations
- Solve $A\mathbf{x} = \mathbf{b}_x$ for the $x$ coordinates
- Solve $A\mathbf{y} = \mathbf{b}_y$ for the $y$ coordinates

\[x_1, y_1 = (0,0), \quad x_2, y_2 = (1,0)\]
\[x_3, y_3 = (0,1), \quad x_4, y_4 = (1,1)\]
\[x_5, y_5 = \frac{(x_1, y_1) + (x_6, y_6) + (x_7, y_7)}{3} = (1/3, 1/3)\]
\[x_6, y_6 = \frac{(x_2, y_2) + (x_5, y_5) + (x_8, y_8)}{3} = (2/3, 1/3)\]
\[x_7, y_7 = \frac{(x_3, y_3) + (x_5, y_5) + (x_8, y_8)}{3} = (1/3, 2/3)\]
\[x_8, y_8 = \frac{(x_4, y_4) + (x_6, y_6) + (x_7, y_7)}{3} = (2/3, 2/3)\]
GEM Force Directed Layout

- Edges exert spring force on their nodes
  - preset globally uniform desired edge length
  - spring force diminished by node degree, making higher degree nodes “heavier”
- Nodes mutually repel each other
  - strength $\approx 1/$distance
- All nodes experience global forces
  - gravitational force toward center
  - small random perturbation force

Centralities

• Node degree is a simple centrality measure
  – high degree nodes better connected than low degree nodes
  – PageRank: sum of PageRanks of incoming links
• Isolation metric is total distance to all other nodes
  – Closeness centrality = 1/isolation metric
  – Graph centrality = 1/distance to the farthest node
• Betweenness centrality of a node
  – portion of all shortest paths between any two nodes that pass through the given node
  – Can also compute for an edge

red = low BC, blue = high BC
© 2007 Claudio Rocchini
Creative Commons
Simplification

- Removing lowest BC edges that don’t disconnect graph results in a tree shaped “backbone”
- Simplifies force-directed layout, then can later add less significant edges back in
Edge Bundles

- Aggregate “similar” edges into “wire bundles” to simplify presentation
- Will need measure of edge similarity


389 Enron emails between 132 employees on 8/2001

radial layout  “small worlds” layout  force-directed layout
Community Discovery

• Remove edges in order of decreasing BC, from highest to lowest
• At lowest level reveals communities
• Creates community hierarchies as higher BC edges merge lower-level communities
Edge Bundle Communities
Enron

actual communities

discovered communities
Filtering & Details
Multidimensional Scaling

• Graph encodes nominal relationships
  – Length of edge irrelevant
• How to embed quantitative relationships?
  – Length of edge indicates strength of relationship
• Metric MDS
  – Dimensionality reduction projection of data points from high dimensional space that tries to preserve distances
<table>
<thead>
<tr>
<th></th>
<th>Boston</th>
<th>NYC</th>
<th>DC</th>
<th>Miami</th>
<th>Chicago</th>
<th>Seattle</th>
<th>SF</th>
<th>LA</th>
<th>Denver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NYC</td>
<td>206</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>429</td>
<td>244</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miami</td>
<td>1,504</td>
<td>1,308</td>
<td>1,075</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chicago</td>
<td>963</td>
<td>803</td>
<td>671</td>
<td>1,329</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seattle</td>
<td>2,976</td>
<td>2,815</td>
<td>2,684</td>
<td>3,273</td>
<td>2,013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF</td>
<td>3,095</td>
<td>2,934</td>
<td>2,799</td>
<td>3,053</td>
<td>2,142</td>
<td>808</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LA</td>
<td>2,979</td>
<td>2,786</td>
<td>2,631</td>
<td>2,687</td>
<td>2,054</td>
<td>1,131</td>
<td>379</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denver</td>
<td>1,949</td>
<td>1,771</td>
<td>1,616</td>
<td>2,037</td>
<td>996</td>
<td>1,307</td>
<td>1,235</td>
<td>1,059</td>
<td></td>
</tr>
</tbody>
</table>
Multidimensional Scaling

• Given distances $d_{ij}$ between datapoints $i$ and $j$
• Find positions $(x_i, y_i)$
• That minimize a stress function

$$\Sigma_{j=1..N-1} \Sigma_{i=i+1..j} (\| (x_i, y_i) - (x_j, y_j) \| - d_{ij})^2$$

• Can solve with any number of non-linear optimization methods
MDS Uses

• Visualization of affinities
  – areas of collaboration based on co-authorship
  – number of attributes in common

• HCI
  – layouts based on task affinities

• Marketing
  – perceptual maps provide a landscape of products based on attributes