Graphs, graph algorithms (for image segmentation), ... in progress

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
166 36 Prague 6, Jugoslávských partyzánů 3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz

Courtesy: Jianbo Shi

Outline of the talk:
- Graph-based image segmentation, ideas
- Graphs, concepts
- Flow network
- Minimum spanning tree
- Dummy 5.
- Dummy 6.
Graph-based image segmentation, main ideas

◆ Convert an image into a graph

  ● Graph vertices correspond to individual pixels.
  
  ● Edges connect neighboring pixels.
  
  ● Additional graph vertices and edges encode other constraints.
    
    Example: a special node (source) denotes objects and a special node (sink) denotes background in object/background segmentation.
    
    The source/sink concepts come from flow networks.

◆ Manipulate the graph to segment the image.
Related seminal papers

- Y. Boykov, M.-P. Jolly: Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images, ICCV 2001
  - $A$: Pixel classified as object or background. Novelty: adding interactivity.
  - Minimize energy function $E(A) = B(A) + \lambda R(A)$, where $B(A) =$ the cost of all edges between object pixels and background pixels; $R(A) =$ the cost of deciding if a pixel is object or background.

  - Cluster the vertices based on edge weight.

- C. Rother, V. Kolmogorov, A. Blake: GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. ACM Transactions on Graphics (SIGGRAPH’04), 2004
An image represented as a graph

- Nodes correspond to pixels.
- Edges connect neighboring pixels. 4-neighbors are considered in the example.
- Edges weights express the similarity between the neighboring pixels (binary relation).
Image regions, region adjacency graph

- Pixels are grouped to larger regions in the process of (iterative) segmentation.
- Superpixels are regions grouped from pixels of similar intensity or color, in which interpretation is not taken into account.
- Region Adjacency Graph.

input image  regions (here superpixels)  RAG

The above is just an approximation drawn visually. The RAG wasn't computed by any algorithm.

 Courtesy: Images by Vignesh Birodkar.
Graphs, concepts

- It is assumed that a student has studied related graph theory elsewhere. Main concepts are reminded here only.

- [http://web.stanford.edu/class/cs97si/](http://web.stanford.edu/class/cs97si/), lecture 6, 7
- [https://www.cs.indiana.edu/~achauhan/Teaching/B403/LectureNotes/10-graphalgo.html](https://www.cs.indiana.edu/~achauhan/Teaching/B403/LectureNotes/10-graphalgo.html)

### Concepts
- Graphs: directed, undirected
- Adjacency, similarity matrices
- Degree, volume of a vertex
- Graph cut

### Related graph algorithms
- Minimum spanning tree
- Shortest path
- Max graph flow = min graph cut
Undirected/directed graph

Undirected graph

- $G = (V, E)$ is composed of vertices $V$ and undirected edges $E \subset V \times V$ representing an unordered relation between two vertices.
- No self-loops.
- The number of edges $|E| = \mathcal{O}(|V|^2)$.

Directed graph

- $G = (V, E)$ is composed of vertices $V$ and directed edges $E$ representing an ordered relation between two vertices.
- Oriented edge $e = (u, v)$ has the tail $u$ and the head $v$ (notated by the arrow $\rightarrow$). The edge $e$ is different from $e' = (v, u)$ in general.
A weighted graph associates weights with either the edges or the vertices or both. 
E.g., a road map: graph edges are weighted with distances.
**Multigraph, bipartite graph**

- A **multigraph** allows multiple edges between the same vertices. *E.g., the call graph in a program (a function can get called from multiple points in another function).*

- A **bipartite graph** is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent.

  ![Bipartite Graph Diagram](image)

  Courtesy: Wolfram MathWorld.

- Bipartite graphs are equivalent to two-colorable graphs.

- A bipartite graph is a special case of a $k$-partite graph with $k = 2$. 
Path, graph cycle, acyclic graph

- A path from a vertex \( u \) to a vertex \( v \) is a sequence \((v_0, v_1, \ldots, v_k)\) of vertices where \( v_0 = v \), \( v_k = u \), and \((v_i, v_{i+1}) \in E\) for \( i = 0, 1, \ldots, k - 1 \).

- A cycle of a graph \( G \) is a subset of the edge set of \( G \) that forms a path such that the first node of the path corresponds to the last.

- A cycle that uses each graph vertex of a graph exactly once is called a Hamiltonian cycle.

- A graph containing no cycles of any length is known as an acyclic graph. Other graphs are cyclic.

- An acyclic graph is bipartite.

- A cyclic graph is bipartite iff all its cycles are of even length.

A simple path is a path, in which all vertices, except possibly in the first and last, are different.

The length of a path is defined as the number of edges in the path.

If the graph is weighted than the length of the path is the sum of edges weights in the path.

If the graph $G$ is connected then there is a path between every pair of vertices.
\[ |E| \geq |V| - 1. \]

A forest is an acyclic graph.

A tree is a connected acyclic graph. If $|E| = |V| - 1$, then $G$ is a tree.
Degree of a vertex

- The degree of vertex $i$ is the number of graph edges incident on vertex $i$.
- Directed graphs have in-degree, out-degree.

Degree of node: $d_i = \sum_j S_{ij}$

Courtesy: Jinbo Shi
Dense/sparse graph

- Running times are typically expressed in terms of $|E|$ and $|V|$ (often dropping the “’s”).
- The graph is **dense** if $|E| \approx |V|^2$.
- The graph is **sparse** if $|E| \approx |V|$.
- Many interesting graphs are sparse.
  *E.g., planar graphs, in which no edges cross, have $|E| = O(|V|)$ by Euler’s formula.*
- If you know you are dealing with dense or sparse graphs, different data structures may make sense.
Graph represented by the adjacency matrix

- **Adjacency relationship** is:
  - Symmetric if the graph $G$ is undirected.
  - Not necessarily so if $G$ is directed.

- **Adjacency matrix** $A$ has elements $a_{ij}$, $i, j = 1, \ldots, |V|$. Elements $a_{ij} = 1$ if graph vertices $i, j$ share an edge; 0 otherwise.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
In the case of weighted graph, the adjacency matrix is generalized to the similarity matrix $S$ of the graph.

Elements of $A$ have value of the weight of the edge between two respective vertices, $a_{ij} = w_{ij}$.

\[
\begin{array}{cccc}
  a & b & c & d \\
  a & 0 & 1 & 4 & 0 \\
  b & 0 & 0 & 3 & 0 \\
  c & 0 & 0 & 0 & 1 \\
  d & 0 & 0 & 6 & 0 \\
\end{array}
\]
Similarity matrix $S = [S_{ij}]$ is a generalized adjacency matrix.
Adjacency list: for each vertex $v \in V$, store a list of vertices adjacent to $v$.

Our example:

- $\text{Adj}[a] = \{ a, c \}$
- $\text{Adj}[b] = \{ c \}$
- $\text{Adj}[c] = \{ \}$
- $\text{Adj}[d] = \{ c \}$

Variation: For oriented graphs, it is possible to store the list of graph edges coming into the vertex.
Volume of a set:

$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$

Courtesy: Jinbo Shi
Cut in a graph

\[ \text{cut}(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} S_{i,j} \]
A cut is a set of edges \( C \subset E \) such that two vertices (called terminals) became separated on the induced graph \( G' = (V, E \setminus C) \).

Denoting a source terminal as \( s \) and a sink terminal as \( t \), a cut \((S, T)\) of \( G = (V, E) \) is a partition of \( V \) into \( S \) and \( T = V \setminus S \), such that \( s \in S \) and \( t \in T \).
Visual summary of a graph terminology

Similarity matrix \( S = [ S_{ij} ] \)

Degree of node: \( d_i = \sum_j S_{ij} \)

Volume of set: \( \mathcal{A} \)

Graph Cuts

Courtesy: Jinbo Shi
Flow network, flow

- A flow network is a directed graph with nonnegative edge weights (called also capacities).

- A flow $f$ is a real-valued (often integer) function, which satisfies the following three properties:
  
  1. Capacity $c$ constraint
     
     For all $u, v \in V$, $f(u, v) \leq c(u, v)$.

  2. Skew symmetry
     
     For all $u, v \in V$, $f(u, v) = -f(v, u)$.

  3. Flow conservation
     
     For all $u \in (V \setminus \{s, t\})$, $\sum_{v \in V} f(u, v) = 0$. 
**Spanning tree**

- **Definition:** A spanning tree $T$ of an undirected graph $G$ is a subgraph that is a tree which includes all of the vertices of $G$.

- In general, a graph may have several spanning trees.

- If a graph that is not connected will not contain a spanning tree. (cf. spanning forest).

- If all of the edges of $G$ are also edges of a spanning tree $T$ of $G$, then $G$ is a tree and is identical to $T$ (that is, a tree has a unique spanning tree and it is itself).

*One of several possible spanning trees of a $4 \times 4$ grid with 4-neighborhood.*

Minimum Spanning Tree

- **Given**: Given an undirected weighted graph $G = (V, E)$, where $n$ is number of vertices and $m$ is number of edges.

- **Task**: Find a subset of $E$ with the minimum total weight that connects all the nodes into a tree.

---

**Kruskal’s algorithm**

- Takes $\mathcal{O}(m \log m)$ time.

- Easy to code.

- Generally slower than Prim’s algorithm.

**Prim’s algorithm**

- Time complexity depends on the implementation. Can be $\mathcal{O}(n^2 + m)$, $\mathcal{O}(m \log n)$, or $\mathcal{O}(m + n \log n)$.

- More difficult to code.