Graph concepts

Artificial intelligence (CK0031/CK0248)

Francesco Corona

Department of Computer Science
Federal University of Ceará, Fortaleza

Graph concepts
UFC/DC
CK0031/CK0248
2018.2
Generalities
Encoding

Graph concepts

Often times, we have good reasons to believe that one event affects another

- Or, conversely, that some events are independent

Incorporating such knowledge can yield models that are better specified

- Computationally, more efficient solutions

Graphs are mathematical objects that describe how objects are linked

- A convenient model for describing relationships

We introduce a graph structure among the variables of a probabilistic model

The objective is to produce a 'probabilistic graphical model'

$\rightsquigarrow$ A model that captures the relations among variables

Graph concepts
UFC/DC
CK0031/CK0248
2018.2
Generalities
Encoding

Generalities

Graph concepts

Graph concepts
UFC/DC
CK0031/CK0248
2018.2
Generalities
Encoding

Generalities

Definition

Graphs

A graph $\mathcal{K} = (\mathcal{A}, \mathcal{E})$ *is a data structure*

$\rightsquigarrow$ *A set of nodes*

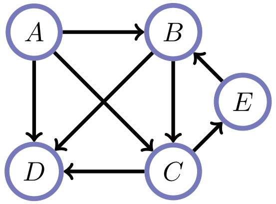
$$\mathcal{A} = \{A_1, \dots, A_N\}$$

$\rightsquigarrow$ *A set of edges between pairs of nodes in* $\mathcal{A}$

$$\mathcal{E} = \{e_1, \dots, e_M\}$$

- Edges may be directed ($A_i \rightarrow A_j$) or undirected ($A_i - A_j$)
- Edges can have associated weights

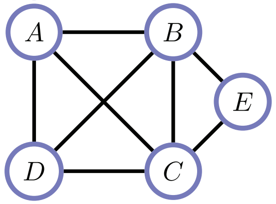
Generalities (cont.)



Directed graph $\mathcal{G}$

- All edges are directed

⇝ $(A_i \rightarrow A_j$ or $A_j \rightarrow A_i)$



Undirected graph $\mathcal{H}$

- All edges are undirected

⇝ $(A_i - A_j)$

Generalities (cont.)

Our use of graphs is to endow them with some probabilistic interpretation

- We develop a connection between directed graphs and probability

Undirected graphs are central in modelling/reasoning with uncertainty

⇝ Variables are independent if not linked by a path on the graph

Generalities (cont.)

Definition

Walks

*A **walk** $A \mapsto B$ from node A to node B is an alternating sequence of nodes and edges that connects A and B*

Let $A_0 = A$ and $A_M = B$, we have

$$A_0, e_1, A_1, e_2 \ldots, A_{M-1}, e_M, A_M$$

Each edge $(A_{m-1}, A_m)_{m=1}^{M}$ is in $\mathcal{K}$

- M is the **length** of the walk

Definition

Trails** and **paths

⇝ ***Trails**, walks without repeated edges*

⇝ ***Paths**, trails without repeated nodes*

They can be understood as refinements of a basic walk

Generalities (cont.)

Definition

Ancestors** and **descendants**, **parents** and **children

Consider a directed graph $\mathcal{K}$

- *Nodes A, such that $A \mapsto B$ and $B \not\mapsto A$ are the **ancestors** of B*
- *Nodes B, such that $A \mapsto B$ and $B \not\mapsto A$ are the **descendants** of A*

Suppose that we have an edge such that $A_i \rightarrow A_j \in \mathcal{E}$

⇝ A_j is the **child** of A_i in $\mathcal{K}$

$$\mathrm{ch}(A_j) \text{ denotes the children of } A_j$$

⇝ A_i is the **parent** of A_j in $\mathcal{K}$

$$\mathrm{pa}(A_i) \text{ denotes the parents of } A_i$$

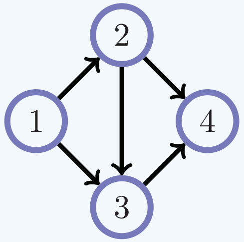
Generalities (cont.)

Definition

Cycles* and *loops

*A **cycle** is a directed path that starts and returns to the same node*

$$a \rightarrow b \rightarrow \cdots \rightarrow z \rightarrow a$$



*A **loop** is a path containing more than two nodes, irrespective of edge direction, that starts and returns to the same node*

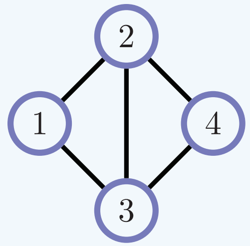
- $1 - 2 - 4 - 3 - 1$

*This graph is **acyclic***

Generalities (cont.)

Definition

Chords



*A **adjacency** is a notion of connectivity*

- *Two nodes A_i and A_j are said to be **adjacent** if joined by an edge in $\mathcal{E}$*

*A **chord** is an edge that connects two non-adjacent nodes in a loop*

- *Edge $2-3$ is a chord in the $1-2-4-3-1$ loop*

Generalities (cont.)

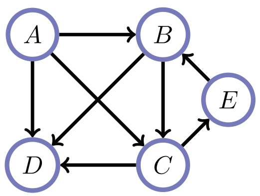
Definition

Directed Acyclic Graph*, *DAG

*A **DAG** is a particular graph $\mathcal{G}$ with directed edges between the nodes*

- *Consider following a path of nodes from one node to another*
- *In a path, we move along the direction of each edge*
- *In a path, we do not revisit edges and nodes*

In a DAG, there is no path that will revisit a node

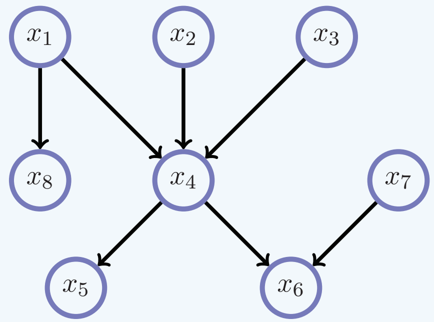


- ⇝ Ancestors of B are nodes who have a directed path ending at B
- ⇝ Descendants of A are nodes who have a directed path starting at A

Generalities (cont.)

Definition

Relations in a DAG



*The **parents** of x_4*

- $pa(x_4) = \{x_1, x_2, x_3\}$

*The **children** of x_4*

- $ch(x_4) = \{x_5, x_6\}$

*The **Markov blanket** of a node is a set of nodes consisting of its parents, its children and the parents of its children (the node itself is excluded)*

- *The Markov blanket of x_4 is $\{x_1, x_2, x_3, x_5, x_6, x_7\}$*

Graph concepts (cont.)

Definition

Neighbours and boundary

For an undirected graph $\mathcal{G}$, the neighbours of x, $ne(x)$, are those nodes directly connected to x

We define the boundary of x, $boundary(x)$, to be $pa(x) \cup ne(x)$

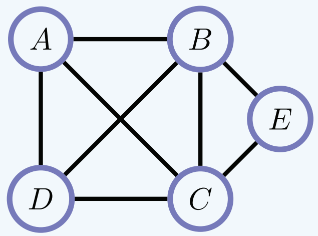
Generalities (cont.)

Definition

Clique

Given a graph, a clique is a fully connected (complete) subset of nodes

- *For the maximal clique, no larger clique containing the clique*



Two maximal cliques

- $\mathcal{C}_1 = \{A, B, C, D\}$
- $\mathcal{C}_2 = \{B, C, E\}$

Whilst $\{A, B, C\}$ is fully connected, this is a non-maximal clique

- *It is a cliquo*

$\{A, B, C, D\}$ is a larger fully connected set that contains $\{A, B, C\}$

Generalities (cont.)

Cliques play a central role in both modelling and inference

In modelling

- They describe variables that are all dependent on each other

In inference

- They describe sets of variables with no simpler structure describing the relationship between them
- No simpler efficient inference procedure is likely to exist

Generalities (cont.)

Definition

Connected graph

An undirected graph is said to be connected if there is a path between every pair of nodes (no isolated islands)

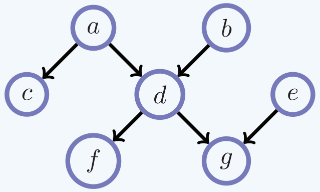
For a non-connected graph, the connected components are those subgraphs which are connected

Generalities (cont.)

Definition

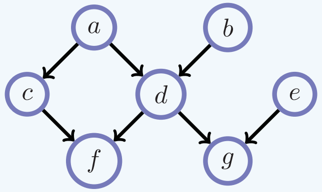
Singly- and multiply-connected graphs

*A graph is **singly connected** if there is only one path from any node A to any other node B, otherwise the graph is **multiply connected***



Singly-connected graph

- *Also called a **tree***



Multiply-connected graph

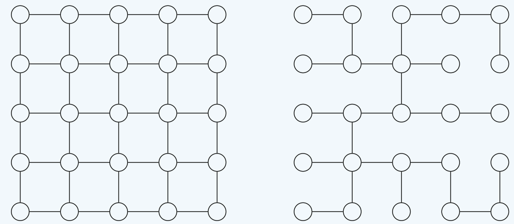
- *Also called **loopy***

Generalities (cont.)

Definition

Spanning tree

*A **spanning tree** of an undirected graph $\mathcal{G}$ is a singly-connected subset of edges such that the resulting singly-connected graph covers all nodes of $\mathcal{G}$*



A **maximum weight spanning tree** is a spanning tree such that the sum of all weights on the edges of the tree is at least as large as any spanning tree

Generalities (cont.)

Pseudo-code

Finding a maximal weight spanning tree

An algorithm to find a spanning tree with maximal weight is as follows

1. *Pick the edge with the largest weight and add it to the edge set*
2. *Pick the next candidate edge and add it to the edge set*
3. *If this results in an edge set with cycles, reject the candidate edge and propose the next largest edge weight*

Note that there may be more than one maximal weight spanning tree

Numerical encoding

Graph concepts

Numerical encoding

Our prime goal is to make computational implementations of inference

- $\rightsquigarrow$ We need to express graphs in a way that a computer can manipulate
- $\rightsquigarrow$ We want to incorporate graph structure into probabilistic models

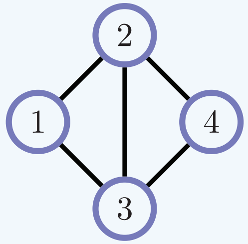
There are several equivalent possibilities

Edge list

Definition

*An **edge list** is a list containing which node-node pairs are in the graph*

$L = \{(1,2),(2,1),(1,3),(3,1),(2,3),(3,2),(2,4),(4,2),(3,4),(4,3)\}$



Undirected edges are listed twice

- *once for each direction*

Adjacency matrix

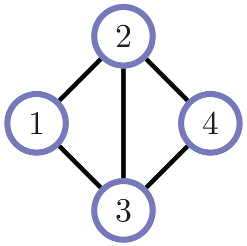
Definition

*The $|\mathcal{A}| \times |\mathcal{A}|$ binary matrix $\mathbf{A}$ called **adjacency matrix***

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{1}$$

$A_{ij} = 1$ if there is an edge from node i to node j, and $A_{ij} = 0$ otherwise

- *$\mathbf{A}$ may include self-connections*
- *1s on the diagonal ($A_{ii} = 1$)*



An undirected graph

- It has a symmetric adjacency matrix

Adjacency matrix (cont.)

Adjacency matrices are useful not only for storing connectivity info

- Certain operations on $\mathbf{A}$ yield additional info concerning $\mathcal{G}$

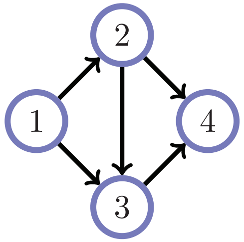
The row-sum $\mathbf{A}_{i+} = \sum_j A_{ij}$ is equal to the **degree** d_i of node i

- The **degree** of a node x is the number of edges incident on it
- A node x is **incident** on an edge e, if x is an endpoint of e

Adjacency matrix (cont.)

Let nodes be labelled in **ancestral order** (parents always before children)

- A directed graph can be represented as a triangular adjacency matrix



$$\mathbf{T} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

Adjacency matrix (cont.)

Definition

Consider a $N \times N$ adjacency matrix $\mathbf{A}$

Consider the k-th powers of the adjacency matrix $[\mathbf{A}^k]_{ij}$

They specify the number of paths from node i to node j, in k edge hops

Consider an adjacency matrix $\mathbf{A}$ and let the diagonal of $\mathbf{A}$ include 1s

Then, $[\mathbf{A}^{N-1}]_{ij}$ is non-zero when there is a path between i to j

- If $\mathbf{A}$ corresponds to a DAG, then the non-zero entries of the j-th row of $[\mathbf{A}^{N-1}]$ correspond to a descendant of node j

Consider an adjacency matrix $\mathbf{A}$

- $[\mathbf{A}]_{ij} = 1$ if one can reach state i from state j in one time step
- $[\mathbf{A}]_{ij} = 0$ otherwise

Element $[\mathbf{A}^k]_{ij}$ gives the number of paths from j to state i in k steps

Incidence matrix and graph Laplacian

Definition

Incidence matrix

$\mathbf{B}$, *$|\mathcal{A}| \times |\mathcal{E}|$ binary matrix capturing structure in $\mathcal{G}$*

$$B_{ij} = \begin{cases} 1, & \textit{if vertex } i \textit{ is incident to edge } j \\ 0, & \textit{otherwise} \end{cases}$$

We extend the incidence matrix $\mathbf{B}$ to a signed incidence matrix $\tilde{\mathbf{B}}$

- The entries 1 of $\mathbf{B}$ are given a $+$ or a $-$ sign

The sign indicates an arbitrarily assigned orientation of the edge

It can be shown that $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T = \mathbf{D} - \mathbf{A} = \mathbf{L}$

$\mathbf{D} = \text{diag}\big[(d_i)_{i \in \mathcal{V}}\big]$ is a diagonal matrix with the degree sequence

$\mathbf{L}$ is the $|\mathcal{V}| \times |\mathcal{V}|$ **graph Laplacian** of $\mathcal{G}$

For a $\mathbf{x} \in \mathcal{R}^{|\mathcal{V}|}$, we have $\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{\{i,j\} \in \mathcal{E}} (x_i - x_j)^2$

It gets closer to 0 as elements of $\mathbf{x}$ at adjacent nodes in $\mathcal{V}$ get more similar

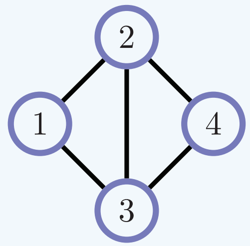
- It can be understood as a measure of *smoothness* of functions on $\mathcal{G}$

Clique matrix

Definition

Consider an undirected graph with N nodes and maximal cliques $\mathcal{C}_1, \dots, \mathcal{C}_k$

- *A **clique matrix** is a $N \times K$ matrix in which each column c_k has zeros except for ones on entries describing the clique*



0 if the node not on the clique

1 if the node is in the clique

$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \quad (3)$$

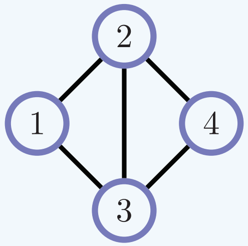
- *Cliques along the columns*
- *Nodes along the rows*

A **cliquo matrix** relaxes the constraint that cliques need be maximal

Clique matrix (cont.)

Definition

A cliquo matrix containing only two-node cliques is an incidence matrix



$$\mathbf{C}_{inc} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (4)$$

$\mathbf{C}_{\text{inc}}\mathbf{C}_{\text{inc}}^T$ is nearly equal to the adjacency matrix

The diagonals contain the **degree** of each node (number of nodes it touches)

- For any cliquo matrix, the diagonal entry of $[\mathbf{C}\mathbf{C}^T]_{ii}$ expresses the number of cliquos (columns) that node i occurs in
- Off-diagonal elements $[\mathbf{C}\mathbf{C}^T]_{ij}$ contain the number of cliquos that node i and j jointly inhabit