Some basic definition and concepts from Graph Theory used in Graphical Model Theory

Guillaume Obozinski

Swiss Data Science Center

African Masters of Machine Intelligence, 2018-2019, AIMS, Kigali

Graphs

Graph defined as a pair $G = (V, E)$

- V is a finite set of nodes
- E is a set of edges

Graphs

Graph defined as a pair $G = (V, E)$

- V is a finite set of nodes
- E is a set of edges

Edges in directed graphs are directed

Directed edges are couples of nodes $(u, v) \in E \subset V \times V$

Graphs

Graph defined as a pair $G = (V, E)$

- V is a finite set of nodes
- E is a set of edges

Edges in directed graphs are directed

Directed edges are couples of nodes $(u, v) \in E \subset V \times V$

Edges in undirected graphs are undirected

Directed edges are pairs of nodes $\{u, v\} \in E \subset \binom{V}{2}$

Clearly $(u, v) \neq (v, u)$ and $\{u, v\} = \{v, u\}$.

Remarks on graph definitions

Remark In this course we will only consider graphs with no self-edges ($\{v, v\}$ or (v, v))

Outline

Concepts in undirected graphs: Neighbors and cliques

Neighbors $\mathcal{N}(u)$ of a node u

$$\mathcal{N}(u) = \{\, v \in V \mid \{u, v\}\}$$

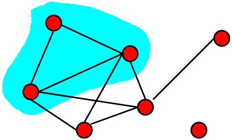
Concepts in undirected graphs: Neighbors and cliques

Neighbors $\mathcal{N}(u)$ of a node u

$$\mathcal{N}(u) = \{\, v \in V \;\mid\; \{u, v\}\}$$

Clique

A totally connected subset of nodes.



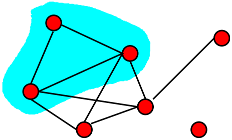
Concepts in undirected graphs: Neighbors and cliques

Neighbors $\mathcal{N}(u)$ of a node u

$$\mathcal{N}(u) = \{\, v \in V \ \mid \ \{u, v\}\}$$

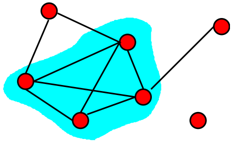
Clique

A totally connected subset of nodes.



Maximal clique

A clique that is not contained in a larger clique.



Concepts in undirected graphs: Paths and cycles

Path

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t. $\forall i$, $\{v_{i-1}, v_i\} \in E$.

Concepts in undirected graphs: Paths and cycles

Path

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t. $\forall i$, $\{v_{i-1}, v_i\} \in E$.

Cycle

A sequence of nodes $(v_0, v_1, \ldots, v_{k-1}, v_0)$ s.t. $(v_0, v_1, \ldots, v_{k-1})$ is a path and $\{v_{k-1}, v_0\} \in E$.

Concepts in undirected graphs: Connectedness

The relation $a \sim_G b$ defined by "there exists a path between a and b" is an *equivalence relation*[1].

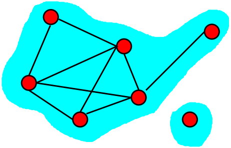
[1] A binary relation which is reflexive, symmetric and transitive.

Concepts in undirected graphs: Connectedness

The relation $a \sim_G b$ defined by "there exists a path between a and b" is an *equivalence relation*[1].

Concepts in undirected graphs: Connected component

The connected components of G are the equivalence classes of the relation $\sim$.



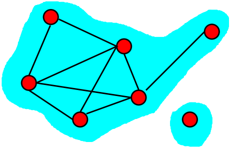
[1] A binary relation which is reflexive, symmetric and transitive.

Concepts in undirected graphs: Connectedness

The relation $a \sim_G b$ defined by "there exists a path between a and b" is an *equivalence relation*[1].

Concepts in undirected graphs: Connected component

The connected components of G are the equivalence classes of the relation $\sim$.



Connected graph

A graph is connected iff it has a single connected component.

[1] A binary relation which is reflexive, symmetric and transitive.

Other concepts in undirected graph

Induced graph

If $G = (V, E)$ is a graph. The *induced graph* on $A \subset V$ is the graph

$$G|_A := (A, E \cap A \times A).$$

Other concepts in undirected graph

Induced graph

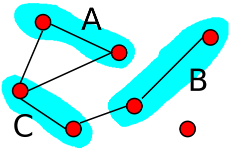
If $G = (V, E)$ is a graph. The *induced graph* on $A \subset V$ is the graph

$$G|_A := (A, E \cap A \times A).$$

Separation

Let A, B, S three disjoint subsets of V.
S separates A from B iff

- all paths from $a \in A$ to $b \in B$ go through S
- equivalently: any connected component K the graph induced on $V \setminus S$ is such that either $K \cap A = \varnothing$ or $K \cap B = \varnothing$



Outline

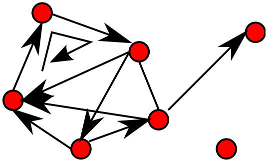
Directed Acyclic Graph (DAG)

A directed graph is called *acyclic* if if contains no (directed) cycle.

Directed Acyclic Graph (DAG)

A directed graph is called *acyclic* if if contains no (directed) cycle.

Counterexample:



Some definitions in directed graphs

Parent and Child

u is a parent of v iff v is a child of u iff $(u, v) \in E$

Some definitions in directed graphs

Parent and Child

u is a parent of v iff v is a child of u iff $(u, v) \in E$

Path (or directed path)

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t. $\forall i,\ (v_{i-1}, v_i) \in E$.

Some definitions in directed graphs

Parent and Child

u is a parent of v iff v is a child of u iff $(u, v) \in E$

Path (or directed path)

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t. $\forall i,\ (v_{i-1}, v_i) \in E$.

Trail

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t.

$$\forall i, \quad \text{either} \quad (v_{i-1}, v_i) \in E \quad \text{or} \quad (v_i, v_{i-1}) \in E.$$

Some definitions in directed graphs

Parent and Child

u is a parent of v iff v is a child of u iff $(u, v) \in E$

Path (or directed path)

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t. $\forall i,\ (v_{i-1}, v_i) \in E$.

Trail

A sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ s.t.

$$\forall i, \quad \text{either} \quad (v_{i-1}, v_i) \in E \quad \text{or} \quad (v_i, v_{i-1}) \in E.$$

Some more definitions in a DAG

Ancestor

u is a ancestor of v ($u \preceq_G v$) iff there is a directed path from u to v.
u is strict ancestor if in addition $u \neq v$.

Some more definitions in a DAG

Ancestor

u is a ancestor of v ($u \preceq_G v$) iff there is a directed path from u to v.
u is strict ancestor if in addition $u \neq v$.

Descendant

v is a (strict) descendant of u iff u is a (strict) ancestor of v.

Topological order for a DAG

A topological order is a *total order* $\prec$ on V compatible with the partial order $\prec_G$ in the sense that

$$u \prec_G v \;\Rightarrow\; u \prec v$$

In other words, if $v_1, v_2, \ldots, v_n$ are in topological order the ancestors of v_i are among $(v_j)_{j \leq i}$.

Topological order for a DAG

A topological order is a *total order* $\prec$ on V compatible with the partial order $\prec_G$ in the sense that

$$u \prec_G v \;\Rightarrow\; u \prec v$$

In other words, if $v_1, v_2, \ldots, v_n$ are in topological order the ancestors of v_i are among $(v_j)_{j \leq i}$.

Proposition

A topological order always exists.

Proof: Induction by removing a maximal element.

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Directed tree

An directed tree is a (connected) DAG in which each node has a single parent.

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Directed tree

An directed tree is a (connected) DAG in which each node has a single parent.

Remarks:

- This definition is the one used in graphical model theory. (Not universally used in CS)

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Directed tree

An directed tree is a (connected) DAG in which each node has a single parent.

Remarks:

- This definition is the one used in graphical model theory. (Not universally used in CS)
- A directed tree is *not* an undirected tree with any orientation of the edges

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Directed tree

An directed tree is a (connected) DAG in which each node has a single parent.

Remarks:

- This definition is the one used in graphical model theory. (Not universally used in CS)
- A directed tree is *not* an undirected tree with any orientation of the edges
- Sometimes called a *rooted tree* because edges must be oriented away from a root.

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Directed tree

An directed tree is a (connected) DAG in which each node has a single parent.

Remarks:

- This definition is the one used in graphical model theory. (Not universally used in CS)
- A directed tree is *not* an undirected tree with any orientation of the edges
- Sometimes called a *rooted tree* because edges must be oriented away from a root.
- A DAG whose underlying undirected graph is a tree is called a *polytree* (or an *oriented tree*).

Trees

Undirected tree

An undirected tree is a (connected) undirected graph without cycle

Directed tree

An directed tree is a (connected) DAG in which each node has a single parent.

Remarks:

- This definition is the one used in graphical model theory. (Not universally used in CS)
- A directed tree is *not* an undirected tree with any orientation of the edges
- Sometimes called a *rooted tree* because edges must be oriented away from a root.
- A DAG whose underlying undirected graph is a tree is called a *polytree* (or an *oriented tree*).

Forest vs trees, etc

- In graph theory, a forest is a disjoint union of trees (both in the directed and undirected case)
- In this course, we will often not make the distinction and use the word tree even for a graph which is a forest, because the same theory applies to both.
- More generally, when a graph has several connected components, we will be able to treat one component at a time for all relevant task of graphical model theory.