Networks and Flows on Graphs

Fixing Graph Theoretical Terminology

Bashar Dudin

February 11, 2019

EPITA

What Is a Graph?

Definition

A *directed graph* (or *digraph*) G is given by a set V of *vertices* together with a set A of *arrows*; an arrow being an (ordered) couple $a = (x, y)$ of vertices. The vertex x is called the *source* of a while y is its *target*. We write $G = (V, A)$ for the digraph G.

What Is a Graph?

Definition

A *directed graph* (or *digraph*) G is given by a set V of *vertices* together with a set A of *arrows*; an arrow being an (ordered) couple $a = (x, y)$ of vertices. The vertex x is called the *source* of a while y is its *target*. We write $G = (V, A)$ for the digraph G.

Question

What is the most general digraph you can draw? Can you have two loops for a single edge? How many arrows in between two vertices?

What Is a Graph?

Definition

An *(undirected) simple graph* G is given by a set V of vertices together with a set A of *edges*; an edge being an (unordered) pair $e = \{x, y\}$ of vertices. We write $G = (V, E)$ for the digraph G.

What Is a Graph?

Definition

An *(undirected) simple graph* G is given by a set V of vertices together with a set A of *edges*; an edge being an (unordered) pair $e = \{x, y\}$ of vertices. We write $G = (V, E)$ for the digraph G.

Remark : Notice that to be a pair $\{x, y\}$ you need to have $x \neq y$. So, simple graphs do not have loops.

What Is a Graph?

Definition

An *(undirected) simple graph* G is given by a set V of vertices together with a set A of *edges*; an edge being an (unordered) pair $e = \{x, y\}$ of vertices. We write $G = (V, E)$ for the digraph G.

Remark : Notice that to be a pair $\{x, y\}$ you need to have $x \neq y$. So, simple graphs do not have loops.

There is a canonical way of attaching a simple graph to a digraph G :

- Delete loops of the set of arrows of G, these are given by couples having two identical entries

What Is a Graph?

Definition

An *(undirected) simple graph* G is given by a set V of vertices together with a set A of *edges*; an edge being an (unordered) pair $e = \{x, y\}$ of vertices. We write $G = (V, E)$ for the digraph G.

Remark : Notice that to be a pair $\{x, y\}$ you need to have $x \neq y$. So, simple graphs do not have loops.

There is a canonical way of attaching a simple graph to a digraph G :

- Delete loops of the set of arrows of G, these are given by couples having two identical entries
- build up the set of edges as the collectiong of pairs $\{x, y\}$ for each arrow (x, y) in G.

What Is a Graph?

Definition

An *(undirected) simple graph* G is given by a set V of vertices together with a set A of *edges*; an edge being an (unordered) pair $e = \{x, y\}$ of vertices. We write $G = (V, E)$ for the digraph G.

Remark : Notice that to be a pair $\{x, y\}$ you need to have $x \neq y$. So, simple graphs do not have loops.

Question

Can you think of other ways of defining a graph? For instance how would you give a definition to allow multiple arrows in a digraph? Or to allow loops and multiple edges in a graph?

Finding Your Way in Graphs

Graph

Chain : A sequence of edges where each edge has a common vertex with the preceding one (except the first), the other being common with the next edge (except the last).

Cycle : A closed chain.

Simple chain : Containing each edge at most once.

Elementary chain : Containing each vertex at most once.

Hamiltonian chain : Passing once by each vertex.

Eulerian chain : Passing once by each edge.

length of chain : The number of edges in the chain.

Digraph

Path : A sequence of arrows where each arrow's target is the source of the next arrow (except the last).

Circuit : A closed path.

Simple path : Containing each arrow at most once.

Elementary path : Containing each vertex at most once.

Hamiltonian path : Passing once by each vertex.

Eulerian path : Passing once by each arrow.

length of chain : The number of arrows in the chain.

Connectedness

Graph

Connectedness : A graph is said to be *connected* if any two distinct vertices are the endpoints of a chain.

Connected components : Maximal subgraphs that are connected.

Digraph

Strong connectedness : A digraph is said to be *strongly connected* if any two distinct vertices are the initial source and target of a path.

Strongly connected components : Maximal subraphs that are strongly connected.

Remark : A digraph is said to be connected if its underlying graph is connected. Connected components of a digraph are defined the same way.

Two extreme cases of connected graphs

Trees

A tree T is a graph where each two vertices are linked by exactly one chain. It is equivalently given by

- T is connected and cycle-free
- T is connected of maximal order
- T is connected and deleting any edge disconnects it
- T is cycle-free and adding any edge creates one.

Complete Graphs

The complete graph $\mathscr{K}_n$ is the graph having n vertices and all possible edges linking them.

It has exactly $\frac{n(n-1)}{2}$ edges.

Can you draw $\mathscr{K}_5$ on a paper without having two edges overlapping?

Adjacency Matrix

Definition

Let $G = (V, E)$ be a graph. Two distinct vertices of G are said to be adjacent if they are endpoints of the same edge.

Adjacency Matrix

Definition

Let $G = (V, E)$ be a graph. Two distinct vertices of G are said to be adjacent if they are endpoints of the same edge.

One can represent a graph in a machine in one of the two following ways:

Adjacency Matrix

Definition

Let $G = (V, E)$ be a graph. Two distinct vertices of G are said to be adjacent if they are endpoints of the same edge.

One can represent a graph in a machine in one of the two following ways:

- As a collection of vertices, each coming with its list of adjacent vertices

Adjacency Matrix

Definition

Let $G = (V, E)$ be a graph. Two distinct vertices of G are said to be adjacent if they are endpoints of the same edge.

One can represent a graph in a machine in one of the two following ways:

- As a collection of vertices, each coming with its list of adjacent vertices
- As a matrix called the *adjacency matrix* of the graph.

Adjacency Matrix

Definition

Let $G = (V, E)$ be a graph. Two distinct vertices of G are said to be adjacent if they are endpoints of the same edge.

One can represent a graph in a machine in one of the two following ways:

- As a collection of vertices, each coming with its list of adjacent vertices
- As a matrix called the *adjacency matrix* of the graph.

Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices. The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. Entry (i, j) is 1 iff there is an edge (arrow) from i to j, otherwise it is zero.

Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices. The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. Entry (i, j) is 1 iff there is an edge (arrow) from i to j, otherwise it is zero.

Remark : The adjacency matrix of a graph is symmetric. A digraph having a symmetric adjacency matrix has, for any given arrow (x, y) between distinct vertices, an arrow (y, x) going the other way around.

Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices. The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. Entry (i, j) is 1 iff there is an edge (arrow) from i to j, otherwise it is zero.

Remark : The adjacency matrix of a graph is symmetric. A digraph having a symmetric adjacency matrix has, for any given arrow (x, y) between distinct vertices, an arrow (y, x) going the other way around. That's the reason why some mathematical textbooks see a graph as a digraph having this property. In that context arrows are called half-edges.

Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices. The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. Entry (i, j) is 1 iff there is an edge (arrow) from i to j, otherwise it is zero.

Remark : The adjacency matrix of a graph is symmetric. A digraph having a symmetric adjacency matrix has, for any given arrow (x, y) between distinct vertices, an arrow (y, x) going the other way around. That's the reason why some mathematical textbooks see a graph as a digraph having this property. In that context arrows are called half-edges.

Question

Do you know any interesting theoretical results about symmetric matrices?

Adjacency Matrix

Proposition

Let G be either a graph or a digraph and write M for its adjacency matrix. For any given $k \in \mathbb{N}^*$, the matrix M^k has entry (i, j) equal to the number of chains (paths) having source i and target j.

Adjacency Matrix

Proposition

Let G be either a graph or a digraph and write M for its adjacency matrix. For any given $k \in \mathbb{N}^*$, the matrix M^k has entry (i, j) equal to the number of chains (paths) having source i and target j.

Proof : This is done by induction. In case $k = 1$, a path of length 1 is just an edge (or an arrow) and this is just the definition of the adjacency matrix.

Adjacency Matrix

Proposition

Let G be either a graph or a digraph and write M for its adjacency matrix. For any given $k \in \mathbb{N}^*$, the matrix M^k has entry (i, j) equal to the number of chains (paths) having source i and target j.

Proof : This is done by induction. In case $k = 1$, a path of length 1 is just an edge (or an arrow) and this is just the definition of the adjacency matrix. Assume that entries of the matrix M^k correspond to the number of chains (paths) from the row to the column index. Let $a_k[i, j]$ be the (i, j) coefficient of M^k. Then the entry $a_{k+1}[i, j]$ of M^{k+1} is given by

$$\begin{aligned} a_{k+1}[i, j] &= \sum_{\ell=1}^{n} a_k[i, \ell] a_1[\ell, j] \\ &= \sum_{\{\ell \mid \ell \text{ is adjacent to } j\}} a_k[i, \ell] \end{aligned}$$

which is exactly what we are looking for. ■

Generalization and Variant of Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices and *weighted* edges (resp. arrows). The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. If there is an edge from i to j then entry (i, j) gets the weight of that edge, otherwise entry (i, j) is zero.

Generalization and Variant of Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices and *weighted* edges (resp. arrows). The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. If there is an edge from i to j then entry (i, j) gets the weight of that edge, otherwise entry (i, j) is zero.

Given such a graph G with possible weights different from 1, then the k-th power of its adjacency matrix M is less easily interpretable; the contribution of each edge or arrow to a path will be taken with the corresponding weight.

Generalization and Variant of Adjacency Matrix

Definition

Let G be a graph (resp. digraph) having n vertices and *weighted* edges (resp. arrows). The *adjacency matrix* of G is a square matrix having n columns and n rows, identically indexed by the vertices. If there is an edge from i to j then entry (i, j) gets the weight of that edge, otherwise entry (i, j) is zero.

Given such a graph G with possible weights different from 1, then the k-th power of its adjacency matrix M is less easily interpretable; the contribution of each edge or arrow to a path will be taken with the corresponding weight.

Sometimes we are only interested in the fact there is an edge, arrow, chain or path between two given vertices. In that case we look at the adjacency matrix as a *boolean* matrix. This will be worked out in an exercice later on.

It’s all for now!