Addendum to Section 2.9: "Connectedness"

Michele Coscia (mcos@itu.dk)

August 2, 2018

Abstract

Here we introduce the concept of strongly and weakly connected components, and of in- and out-components.

1 Strong/Weak Connectivity

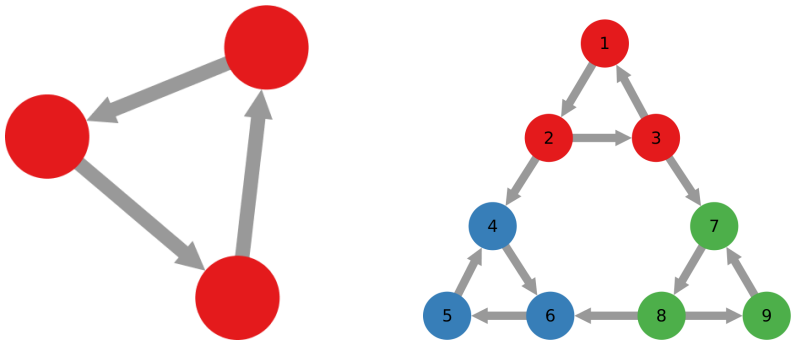
In section 2.9 of the textbook we saw that the measure of a network's usefulness is *connectedness*. If there is no way to follow the edges of the network from node i to node j, then i has no way to influence – or communicate to – j. However, we dealt only with the case of undirected graphs. What if our edges are not symmetric, but have a direction?

In that case we have two different scenarios. In the first scenario, which we call *strong*, we want to ensure the same ability that the undirected network endowed us: i must be able to contact j, and vice versa. Figure 1a shows an example of such a component. No matter where we choose to start our path, we can always go back. Therefore, any i can reach any j, and vice versa. Since this strong requirement is satisfied, we call these "strongly connected components" (SCC).

It is not a surprise to reveal that strongly connected components contain cycles. By definition, if you can find a pair of nodes that you cannot join with a cycle – meaning starting from i and passing through j makes it impossible to go back to i – then those nodes are not part of the same strongly connected component. SCCs are important: if you are playing a message-passing game where messages can only go in one direction, you can always hear back from the players in the same strongly connected component as you.

The definition of SCC leaves the door open for some confusion. Even by visually inspecting a network that appears to be connected in a single component, you will find multiple different SCCs – as in Figure 1b. In the figure, there is no path that respects the edge directions and leads from node 1 to node 7 and back. The best one could do is $1 \to 2 \to 3 \to 7 \to 8 \to 6 \to 5 \to 4$.

However, it *feels* like this network should have one component, because we can see that there are no cuts, no isolated vertices. If we were to ignore edge directions, Figure 1b would really look like a connected component in an undirected network. This feeling of uneasiness led network scientists to create the



(a) (b)

Figure 1: (a) A strongly connected component. (b) A network with multiple strongly connected components, coded by different node colors.

concept of "weakly connected components" (WCC). WCCs are exactly what I just wrote: take a directed network, ignore edge directions, and look for connected components in this undirected version of it. Under this definition, Figure 1b has only one weakly connected component.

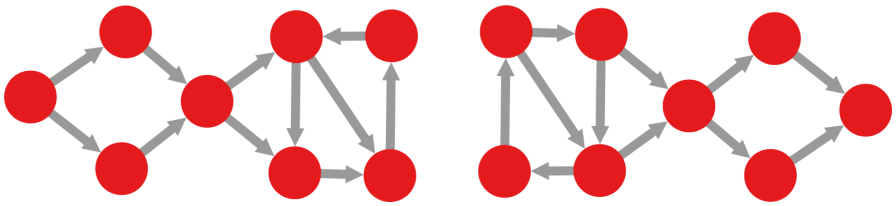
2 In/Out Connectivity

Not all weakly connected components are created equal. In large networks, one can find any sort of weird things. Suppose you are working in an office. The core of the office works on documents together, by passing them to each other multiple times and changing the version of the document. But you're not part of the core of the office, you are in a weakly connected component. Your job is simply to receive a document, stamp it, and pass it to the next desk.

Since you are on a WCC, you know you're never going to see the same document twice. That would imply that there is a cycle, and thus that you are in a strongly connected component with someone. However, what you see in the document can be radically different. The document might arrive to you completely virgin, without any versioning yet. Or it could already have had multiple versions. These two scenarios are quite different.

If you are in the first scenario, it means your WCC is positioned "before" the core. Documents pass through it and they are put *in* the core. The flow of information originates from you or from some other member of the weakly connected component, and it is poured *in*to the core. This is the scenario in Figure 2a: you are one of the four leftmost nodes. In this paragraph I highlighted the word *in* because we decided to call these special WCC *in*-components.

If you are in the second scenario, it means your WCC is positioned "after" the core. The core does its magic on the documents, and then *out*puts them into your weakly connected component. The flow of information originates from the core and it is poured *out* to your WCC. This is the scenario in Figure 2b:



(a) (b)

Figure 2: (a) An in-component, composed by the four leftmost nodes. (b) A out-component, composed by the four rightmost nodes.

you are one of the four rightmost nodes. In this paragraph I highlighted the word *out* because we decided to call these special WCC *out*-components.