Analysis of Citation Networks
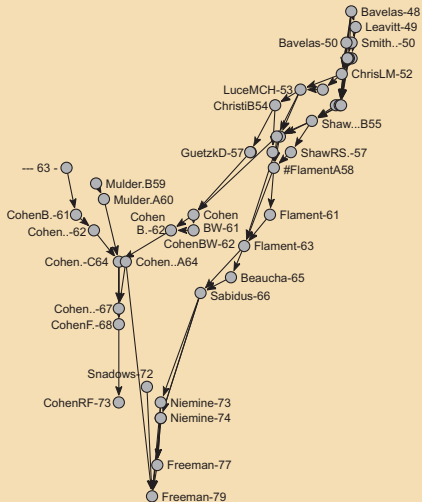
Andrej Mrvar

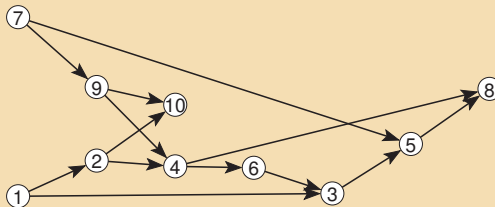***An acyclic network*** is a special kind of directed network.
We say: A directed network is *acyclic* if it contains no cycles:
If we start a walk from any vertex of an acyclic network and follow
directions of arcs there is no way to return to the initial vertex.
Examples of acyclic networks: Critical Path Method, Citation networks,
Genealogies...

In an acyclic network ***first (source) vertices*** exist – vertices with input
degree equal to 0 (e. g. vertices 1 and 7 in the figure). There exist ***last
(sink) vertices*** too – vertices with output degree equal to 0 (e. g.
vertices 8 and 10 in the figure above).

For every vertex of an acyclic network the ***depth*** can be computed: We assign depth 1 to all first vertices. Then we remove first vertices and lines attached to them from the network. In this way we obtain a new acyclic network. We assign depth 2 to new first vertices in the obtained network, remove first vertices and attached lines and continue the procedure.
Example: Vertices in the acyclic network
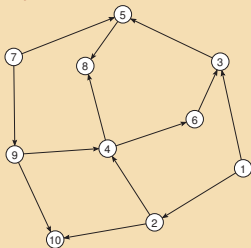
have the following **depths**

vertex	1	2	3	4	5	6	7	8	9	10
depth	1	2	5	3	6	4	1	7	2	3

Depths can be used for drawing an acyclic network in **layers**. If we use layers in x direction we get:
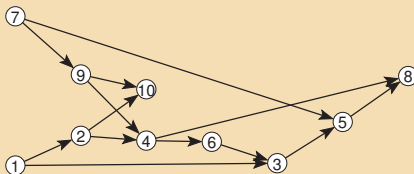
In Pajek depths are computed using:
Network / **Acyclic Network** / **Depth Partition** / **Acyclic**
If the command **Draw** / **Network + First Partition** is used afterwards, each vertex is colored using a color that corresponds to its depth.
Colors used are shown in the color table of Draw window:
Options / **Colors** / **Partition Colors** / **for Vertices**.
Layout in layers is obtained using **Layers** / **in y Direction**. If we want to improve the picture by hand, but we want to keep the layers, we can fix y coordinate using **Move** / **Fix y**
To obtain layers in x direction (from left to right) we need to rotate layout for 90 degrees: **Options** / **Transform** / **Rotate 2D** / **90**

Input: Acyclic network which represents citations among papers.
Result: Relative importance of papers and relative importance of citations. Example:
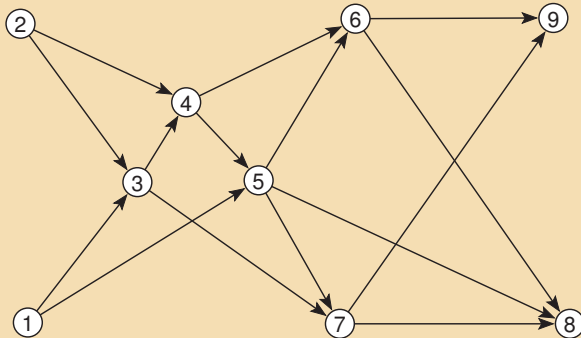
In citation networks vertices represent papers, directed link from paper *x* to paper *y* means, that *y* cites *x* (*x* is cited in *y*).

We want to find out which paper (vertex) is the most important and which citation (link) is the most important in such network.

The simplest such measure which could give as answer to this question is output degree (number of arcs going out of the vertex).

Network output degree can be computed easily:

Network / **Create Partition** / **Degree** / **Output**
or
Network / **Create Vector** / **Centrality** / **Degree** / **Output**

In the example network paper 5 is the most important with 3 citations.

_____

A much better measure is composed by taking number of paths passing through each vertex into account.

Traversal weights algorithm

- Unique first (source) and last (sink) vertices are added to acyclic network.

- We make two passes through this network:

  1 In first pass we start in unique source vertex and travel in direction of arcs until we reach the sink vertex. By doing this we remember number of paths coming to each vertex we reach (this can be done efficiently by summing counts for vertices from which we reach the selected vertex).

  2 In second pass we start in unique sink vertex and travel in the opposite direction of acs until we reach the source vertex. In this second pass we remember number of paths leading from this vertex to sink vertex.

Pajek

...Acyclic networks...

...Citation networks...

Citation networks

Acyclic Networks
Citation weights
Algorithm
Variants
Examples
Pajek
Main path
  Local searches
  Global searches
  Key Routes
Assignments

From the two counts obtained for each vertex we can compute number of paths from the source to the sink vertex that go through selected vertex.

We use the fundamental theorem of combinatorics:

If there exist M paths from the source vertex to selected vertex and if there exist N paths from selected vertex to the sink vertex then number of paths from source vertex to sink vertex that pass through selected vertex is equal to MN.

In a similar way we find out how many paths use selected link. Lets selected link goes from vertex i to vertex j.

If there exist M paths from the source vertex to vertex i and if there exist N paths from vertex j to the sink vertex then number of paths from source vertex to sink vertex that pass through selected link $i \rightarrow j$ is equal to MN.

If we want to compare these counts across different networks we can normalize them to obtain a measure in interval [0, 1].

Three variants of this algorithm exist:

- **Search Paths Count (SPC) method** – this is exactly the method we described (counting all paths from source to sink vertex);

- **Search Path Link Count (SPLC) method** – we consider each vertex in acyclic network as a source vertex: we add links from source vertex to each other vertex in the network. As result vertices and arcs at the end will receive higher traversal weights.

- **Search Path Node Pair (SPNP) method** – each vertex is considered as a source and as a sink vertex: we add links from source vertex to each other vertex in the network and from each vertex to the sink. As a result, vertices and arcs in the middle will receive higher traversal weights.

Citation networks

Acyclic Networks
Citation weights
Algorithm
Variants
Examples
Pajek
Main path
Local searches
Global searches
Key Routes
Assignments

Absolute and relativne importances of papers using SPC, SPLC and SPNP

pap.	*SPC*	*SPC*/30	*SPLC*	*SPLC*/58	*SPNP*	*SPNP*/99
1	14	0.47	14	0.24	25	0.25
2	16	0.53	16	0.28	29	0.29
3	18	0.60	27	0.47	48	0.48
4	21	0.70	35	0.60	60	0.61
5	20	0.67	35	0.60	56	0.57
6	14	0.47	26	0.45	39	0.39
7	12	0.40	22	0.38	33	0.33
8	17	0.57	32	0.55	32	0.32
9	13	0.43	25	0.43	25	0.25

Absolute and relative importances of citations using SPC, SPLC and SPNP

from	to	SPC	SPC/30	SPLC	SPLC/58	SPNP	SPNP/99
1	3	9	0.30	9	0.16	16	0.16
2	3	9	0.30	9	0.16	16	0.16
1	5	5	0.17	5	0.09	8	0.08
2	4	7	0.23	7	0.12	12	0.12
3	7	4	0.13	6	0.10	9	0.09
3	4	14	0.47	21	0.36	36	0.36
4	5	15	0.50	25	0.43	40	0.40
4	6	6	0.20	10	0.17	15	0.15
5	7	8	0.27	14	0.24	21	0.21
5	6	8	0.27	14	0.24	21	0.21
5	8	4	0.13	7	0.12	7	0.07
6	8	7	0.23	13	0.22	13	0.13
6	9	7	0.23	13	0.22	13	0.13
7	8	6	0.20	11	0.19	11	0.11
7	9	6	0.20	11	0.19	11	0.11

Two more small examples: citem.net and Citem2a.net
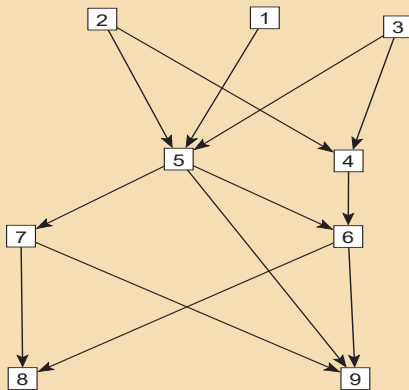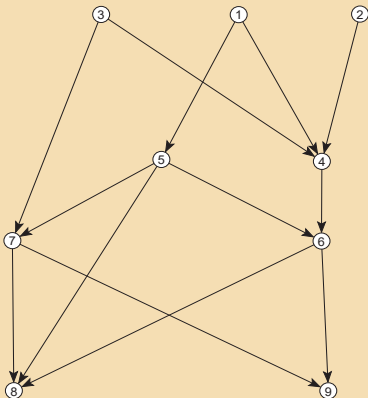
Traversal weights (importances of papers and citations):

Network / **Acyclic Network** / **Create Weighted Network + Vector** / **Traversal Weights** / **SPC** or **SPLC** or **SPNP**

Pajek returns the following results:

1. Vector with number of paths starting in source and finishing in selected vertex (*Number of Different Incoming Paths*).

2. Vector with number of paths from selected vertex to sink vertex (*Number of Different Outgoing Paths*).

3. Vector containing importances of vertices/papers (*Citation weights SPC/SPLC...*).

4. New network where values of lines represent importances of citations (*Citation weights*).
   If we want to see the counts in **Draw** window, we must check **Options** / **Lines** / **Mark Lines** / **with Values**.
   To see top citations select **Network** / **Info** / **General** and type e.g. 10 if we want to list top 10 important citations.

Determining the Main Path from Source to Sink

Traversal weights are used not only to find the most important papers and citations but also the **main path**. In a citation network, a main path is the path from a source vertex to a sink vertex with the highest traversal weights on its arcs. Several methods have been proposed to extract main paths from the network of traversal weights. Methods for extracting main paths can be **local** or **global**.

Local Searches – Searching step by step: In each step of the search select the arc(s) with the highest weight that are incident to the current arc.

Global Searches – Search for path from source to sink vertices with the overall highest sum of traversal weights on the path.

Network / **Acyclic Network** / **Create (Sub)Network** / **Main Paths**

Local Searches

1 **Local** / **Forward**
This approach consists of choosing the source vertex (or vertices) incident with the arc(s) with the highest weight, selecting the arc(s) and the second vertex(s) of the arc(s), and repeating this step until a sink vertex is reached.

2 **Local** / **Backward**
Instead of starting with one or more source vertices, we start with one or more sink vertices incident with the arc(s) with the highest weight and travel against the direction of the arcs.

Tolerance (in case of normalization *flow* it is a number between 0 and 1) can be used. Tolerance 0 means that in each step only arcs with the highest weights are taken into account, while larger tolerance means that also a bit smaller weights are considered.

Example: If tolerance is set to 0.1, and the highest arc value is 0.5, all arcs with values larger than 0.45 are selected (values that are not more than 10% smaller than 0.5).

Global Search – search for main path(s) with the highest overall sum of weights.

1 **Global / Standard**
Standard global main path is the path from source to sink vertices with the overall highest sum of traversal weights on the path. This method is widely used across scientific disciplines. In project planning, for example, it is called Critical Path Method (CPM).

Key Routes

We can search for local or global main paths that contain specific citations (arcs). These are usually the arcs with the highest traversal weights. Selected arcs are called **key-routes**. To search for main paths that must contain top 5 citations we enter 1-5 to the dialog box asking for rank numbers of key-routes.

_____

Instead of searching for main paths containing selected key-routes (these are arcs representing citations) we can also search for main paths containing selected articles (represented by vertices). This can be done in the following way: first select one or more interesting articles (vertices numbers) in a Cluster and run the command: **Through Vertices in Cluster**.

Analyze the citations among articles that discuss the topic of network centrality (`centrality_literature.net`). In 1979, Linton Freeman published an article that defined several kinds of centrality. His typology has become the standard for network analysis. Freeman, however, was not the first to publish on centrality in networks. His article is part of a discussion that dates back to the 1940s. Arcs represent citations; they point from the cited article to the citing article.

Some hints:

1. For traversal weights normalization use: *flow*
2. In local searches look what happens if you use tolerance 0 or 0.07.
3. As key-routes take top 1 and top 2 citations.
4. Search for local an global main paths containing papers by Leavitt (1949) and GilchSW (1954). In network file these are vertices with numbers 2 and 21 respectively.

Top: *forward local* and *key-route local* (only one key-route).
Middle: *standard global* and *key-route global* (only one key-route).
Bottom: *backward local*.

Pajek

Publications and citations pass on scientific knowledge and traditions; so do advisors to their students. The file `PhD.net` contains the ties between Ph.D. students and their advisors in theoretical computer science; each arc points from an advisor to a student. The partition `PhD_year.clu` contains the (estimated) year in which the Ph.D. was obtained. Search for separate research traditions in this network and describe how they evolve.