Citation

Version: Corrected Publisher’s Version
License: Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from: https://hdl.handle.net/1887/29764

Note: To cite this publication please use the final published version (if applicable).
The handle http://hdl.handle.net/1887/29764 holds various files of this Leiden University dissertation.

**Author:** Takes, Frank Willem  
**Title:** Algorithms for analyzing and mining real-world graphs  
**Issue Date:** 2014-11-19
1.1 Introduction

We live in a connected world. In our social and digital lives, we are confronted with networks (or graphs) on a daily basis. When someone tells a story, it is likely that this story passed through various other people that together form a network of social interactions. Online social networks such as Facebook are based on gigantic networks in which people are connected through so-called friendship links. Browsing the internet means traversing a large network of pages that is connected via clickable (hyper)links. Accessing one webpage on a mobile phone creates a few dozen wired or wireless connections between devices in a matter of microseconds. Networks are everywhere around us, and influence the way in which we communicate, socialize, search, navigate and consume information.

When networks are stored in a digital format, they can produce an enormous amount of data. Such a large volume of data is sometimes called big data, not only because of its quantity, but also because the data may arrive at an enormous speed and because the data is usually diverse in terms of what type of information it represents. Data is used in many disciplines of science to verify hypotheses about a certain domain. Popularized under the name data science, large (network) datasets are being generated and investigated by commercial organizations as well as a number of research disciplines.

Within the field of computer science, we specifically consider tasks related to storing, retrieving, manipulating and understanding data in an automated and efficient way. The most simple type of data is called unstructured data, which may for example be the textual content of a news article or numeric measurements from a temperature sensor. On the other hand there is structured data, which refers to data that
is organized according to some data structure or model (note that other researchers sometimes use the term unstructured data for tabular data, and structured data for graphs). A common example of structured data (according to the first definition) is a database, which is made up of tabular structures consisting of different objects (rows) along with attributes (columns) that describe the objects. The majority of this text will however focus on graphs, a type of structured data which will be described in Section 1.2. Next, in Section 1.3 the focus will be on algorithms for computing certain properties of graph data. Given a dataset, one may also be interested in getting a better understanding of the knowledge incorporated in the data, a task broadly addressed by the field of data mining, which will be introduced in Section 1.4. This introductory chapter is concluded in Section 1.5 with an outline of how graphs, algorithms and data mining form the main topics of the following chapters of this thesis.

1.2 Graphs

A graph [135] is one of the most fundamental data structures used in computer science. Graphs are used to describe the relationship between objects within a certain domain. In a graph, the objects are commonly referred to as nodes (also called vertices, actors or entities) and the relationship between two vertices is called an edge (also called a link, an arc or a tie). An example of a small graph is shown in Figure 1.1.

This thesis primarily focuses on real-world graphs, often by other disciplines referred to as networks. Note that from a computer science perspective, and especially from an algorithmic point of view, the term “graph” is often preferred over “network”, as the latter is often interpreted as a structure of physical connections between multiple devices. A well-known example of a real-world graph is a social network, in which a node represents a person, and a link represents a social relationship between the two people that it connects. Throughout this chapter, online social networks are used as a running example to describe the various concepts that are relevant in (real-world) graphs.

Online social networks (OSNs) [25] are commonly accessed through a website or (mobile) application, and allow a user to create a profile, and then link this profile to other users, forming a network of social connections called the friendship graph. The profile can be enriched with user attributes such as the age, location and gender of the user. Furthermore, the OSN can be used to communicate with other users or to share information by means of for example text, images or video. The first online social networks were introduced around the year 2000, and roughly five years later Facebook, LinkedIn and Twitter were on their way to become online social networking
services with over a hundred million members each.

Moving back to the abstract concept of a graph, there are various ways to characterize a graph based on properties of both the nodes and the edges. If a graph contains different types of nodes, it is called heterogenic, whereas if all nodes are of the same type, it is called homogenic. A homogenic graph is also called a one-mode network. A heterogenic graph with two types of nodes is called a two-mode network or affiliation network if the set of nodes is bipartite, meaning that the node set can be split into two sets of nodes such that for every edge, the source and target node of that edge are in a different set. A two-mode network can be converted into a one-mode network consisting only of nodes of either one of the two types. In the resulting homogenic network, an edge between two nodes (of the one and only type) is present if both nodes linked to the same node of the other type in the original one-mode network. If the relationships in a graph are explicit, then this means that both actors explicitly form a connection (as is the case with a link in the friendship graph of for example Facebook). When links are implicit, it means that the link is based on some common activity or common property of the two actors, such as the fact that two users sent each other a message. A graph with implicit links is likely to be a projection of a two-mode network.

A node can have one or more attributes describing properties of the particular node. In the OSN example this could be the age and location of the person represented by the node. Similarly, an edge can have one or more attributes describing the type of relationship. Graphs with attributes on the nodes or edges are also called annotated graphs. If an edge has one numeric attribute, then this edge attribute is often called the weight of the edge, and the graph is called a weighted graph. In unweighted graphs, there is no edge weight (but for computational reasons, the weight of an edge is usually assumed to be equal to one).

In some cases, the direction of a link is relevant, and the graph is called a directed graph. This is for example the case in the online social network Twitter, where one user can follow another user, without this other user having to explicitly approve this connection. The term reciprocity is used to denote the extent to which links are mu-
tual. Clearly, in a directed graph such as Twitter, reciprocity is only partial. On the other hand, the friendships in Facebook obviously form an undirected graph, as two people are always each other’s friend, and the relationship is thus always symmetric, realizing full reciprocity by design. The number of incoming links of a node in a directed graph is called the indegree (e.g., the number of followers of a Twitter user), and similarly the number of outgoing links is called the outdegree of that node (e.g., the number of people a user follows). In an undirected graph, the indegree and outdegree are equal, and there is simply the notion of the degree of a node (e.g., the number of friends of a user on Facebook). The majority of this thesis deals with directed or undirected homogenic (one-mode) unweighted graphs.

A common property of the graphs that are investigated in this thesis, is that they are based on real-world data, meaning that the nodes of the graph represent for example actual people, physical objects, locations, organizations, digital information or written articles. An example is the so-called webgraph: the “graph of the internet”, representing the way in which millions of pages are connected by means of billions of clickable hyperlinks. Other examples are citation and collaboration networks, in which a node represents a scientist, and a link between two nodes indicates respectively a citation (a directed link) or collaboration (an undirected link).

An example of a collaboration network is given in Figure 1.2. In this figure, a node represents a staff member of the computer science department of Leiden University, and an undirected edge between two people indicates that they collaborated between 2005 and 2012 by writing a paper together. In this figure, the edge width is proportional to the number of co-authored papers, and a thinner gray edge indicates indirect collaboration through a common co-author not employed in Leiden. This one-mode collaboration network can be seen as a projection of a two-mode network consisting of authors and publications, with edges connecting publications to their authors.

Furthermore, Figure 1.2 explains the concept of connected components: groups of nodes where for any two nodes in this group, there exists a path (a sequence of nodes connected through edges) between these two nodes. The figure has three connected components. The distance between two nodes is defined as the minimum number of edges that has to be traversed to get from one node to the other, or alternatively, as the length of a shortest path between these two nodes. Obviously, this measure of distance has a different semantic meaning depending on the type of graph that is considered. In a collaboration network such as that of Figure 1.2, the distance between two people could be an indication of the similarity of these people’s research.

Noteworthy is the fact that many real-world graphs have similar structural properties, even though they are based on completely different data. First of all, real-world graphs are typically sparse, meaning that the number of edges is very low compared to the maximum number of edges that may possibly exist between all the nodes. Second,
Figure 1.2: A graph of 117 scientific collaborations (undirected edges) between 72 staff members (nodes) of the computer science institute of Leiden University. Data is based on staff publication lists from 2005 to 2012. Visualized using NodeXL (http://nodexl.codeplex.com).
these large graphs usually have one large connected component containing the vast majority of the nodes. For example, of a particular online social network considered in Chapter 6, over 99.9% of the in total eight million users is connected via friendship links. Third, most graphs are scale-free, meaning that they have a power-law degree distribution with a fat tail, i.e., there are a lot of nodes with a very low degree, and only a few nodes with a high degree. The fat tail implies that high degree nodes have a degree that is many times larger than the average degree over all nodes. Indeed, high degree nodes of the graph often serve as hubs in the network, realizing very low average node-to-node distances. This fourth commonly observed property is often referred to as the “small-world phenomenon” [71], which is closely related to the concept of “six degrees of separation”, a theory which says that the majority of the people in the world are connected via only six handshakes.

Graphs are studied in many different disciplines of science. Since the sixties, popularized under the name “social network analysis”, networks of social interaction have been extensively studied within the social sciences [139]. There, the goal is to get an understanding of the social interaction between the different actors in a network. Furthermore, physicists refer to large graphs as “complex networks”, and study for example the different models behind networks [7]. It has been shown that the interaction between proteins can be understood by modeling them as a graph [64], demonstrating the applicability of graphs as a model in bioinformatics. Within the field of public administration, large networks of corporations are also studied, for example to model and better understand the global network of corporate control [59]. The structure of such a corporate graph is shown in Figure 1.3, in which a node represents a company in the Netherlands and an edge between two nodes denotes the fact that these companies have a common senior level director.

Indeed, graphs are everywhere, and the interaction between objects that they model is relevant in many areas of research. Whereas other disciplines of science are usually interested in the domain-specific information incorporated in these graphs, for computer scientists, the emphasis is on creating efficient algorithms for storing, analyzing, understanding and computing certain aspects of the graph and addressing the complexity issues that arise when larger graphs are considered.

1.3 Graph algorithms

An algorithm [91] can be defined as a sequence of instructions to solve a particular problem. Computer scientists are generally interested in designing algorithms that solve a problem efficiently, both in terms of time and memory usage.

This thesis specifically considers algorithms for large graphs. This classification of
Figure 1.3: A graph of 3,711 board interlocks (undirected edges) between 1,422 companies (nodes) in the Netherlands. Data originates from the ORBIS database of Bureau van Dijk. Visualized using the Fruchterman-Reingold and ForceAtlas2 algorithms in Gephi (http://gephi.org).
1.3. Graph algorithms

size may seem rather vague when considering the seemingly ever-increasing amount of available storage, memory and computation power. A more precise way would be to say that large graphs cannot be stored in memory as an adjacency matrix, but only as an adjacency list, making certain operations (such as computing the distance between every pair of nodes) more complex. Algorithms for large graphs usually iterate over the nodes or edges of the graph a constant number of times: quadratic (or worse) complexity in the number of nodes or edges is prohibited, and to ensure practical use the complexity of algorithms should be somewhat linear in the number of nodes or edges. To make these “large” numbers a bit more concrete, large graphs today typically have hundreds of thousands or even millions of nodes, and possibly hundreds of millions or billions of links. Usually, it is very hard to properly visualize graphs once the number of nodes and edges increases. See for example Figure 1.3, which shows a graph consisting of “only” 1,422 nodes and 3,711 undirected edges. Standard tools for visualizing graphs are no longer suitable when the size of the graph exceeds say a hundred thousand nodes. Therefore, when computation or measurements on larger graphs have to be done, specialized frameworks that store and manipulate the graph (without worrying about visualization) are used. For most of the experiments presented in this thesis, a custom C++ framework was used.

A substantial number of graph algorithms proposed in the literature deals with modeling or generating graphs using a mathematical model. The focus of graph algorithms discussed in this thesis is on computing or measuring certain properties or characteristics of a given (real-world) graph. A well-known example of such a graph algorithm is Dijkstra’s shortest path algorithm [43], which computes the distance between two nodes of a graph. Graph properties can roughly be divided into local, global and subglobal properties.

Examples of global properties of the graph include the graph diameter (longest shortest path length, see Chapter 2), its average clustering coefficient (the degree to which nodes tend to cluster together on a global scale), or the number of connected components of the graph. On the other hand, local properties say something about individual nodes, with a commonly addressed issue being that of node centrality, the importance of a node in the graph. In the friendship graph of an online social network, the number of friends of a user (the degree of the node) is a typical centrality measure. The importance of pages in the webgraph of the internet is commonly assessed using the PageRank [107] centrality measure, which ranks webpages based on how many other high-ranked pages link to the considered page. These two measures are incorporated in Figure 1.3, where the size of a node is proportional to its degree (larger size means a higher degree), and the node color corresponds to its PageRank value (darker means a higher PageRank value).

A third type of graph algorithms deals with graphs on a subglobal level, computing
or deriving aspects of a group of nodes, with \textit{community detection} \cite{89} algorithms as a well-known example. These algorithms try to cluster the nodes in the graph so that groups of nodes that are more connected amongst each other than with the rest of the graph, form one community. Clustering is also a frequently addressed task in the field of data mining, which is the topic of the next section.

1.4 Data mining

\textit{Data mining} \cite{143} is the field of research that focuses on getting a better understanding of a (large) dataset in an automated way, for example by searching for patterns in the data. The goal is often to find “something new”, i.e., to discover knowledge that is not immediately visible by using common sense or by manually inspecting the data. Alternatively, one could say that data mining deals with converting information into knowledge, a process called \textit{knowledge discovery}. With this in mind, it is often said that data mining is somewhat related to the fields of artificial intelligence, machine learning and statistics. The remainder of this section describes several common data mining tasks, using the small over-simplified database (table) of online social network users from Table 1.1 as an example dataset.

\textit{Association} is the task of relating attributes of the objects, forming so-called association rules that describe the data. In Table 1.1, a possible association rule could be that if the age attribute has a high value, then the number of friends is low. Indeed, age seems somewhat associated with the number of friends in the example table. \textit{Clustering} refers to the task of grouping sets of objects together because they have certain attributes in common. Again considering the example dataset, a possibility would be to group the users into two clusters based on their gender and location: all female users happen to be from the United States, and all male users are not. \textit{Outlier detection} deals with finding single objects or small groups of objects that “stand out” because they do not comply with the patterns or constraints that the other objects do. A possible outlier in the example dataset could for example be Hugo, because he does

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
<th>Age</th>
<th>Location</th>
<th>Photos</th>
<th>Friends</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charlie</td>
<td>male</td>
<td>23</td>
<td>United Kingdom</td>
<td>4</td>
<td>416</td>
</tr>
<tr>
<td>Hugo</td>
<td>male</td>
<td>27</td>
<td>Mexico</td>
<td>0</td>
<td>238</td>
</tr>
<tr>
<td>Jack</td>
<td>male</td>
<td>42</td>
<td>Australia</td>
<td>8</td>
<td>164</td>
</tr>
<tr>
<td>Kate</td>
<td>female</td>
<td>31</td>
<td>United States</td>
<td>815</td>
<td>158</td>
</tr>
<tr>
<td>Rose</td>
<td>female</td>
<td>65</td>
<td>United States</td>
<td>39</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 1.1: A small database (table) containing users of an online social network.
not have any photos.

The three techniques described above are all descriptive: they attempt to describe or summarize the data, for example to discover the knowledge incorporated in the data, or to find interesting patterns that provide more insight in the considered domain. A more predictive task in data mining is that of classification: the process of determining, given an object and its attributes, some other (initially unknown) attribute, which is then referred to as the class of the object. In the example dataset, the class attribute could be the gender of the user, and one way of predicting this class using the data given in Table 1.1 could be to say that all users with more than 30 photos are female.

For the (too) simple example table, each of the traditional data mining tasks described in this section can be executed more or less perfectly. However, more often than not, a dependency or pattern is only true for a (hopefully large) percentage of the instances, and numeric measures have to be used to assess the accuracy of a derived result. When a set of association rules has been derived, a clustering has been made, or an outlier has been found, it can also be a challenging task to determine whether or not the results make sense within the given context of the data. A clustering might be based on a coincidence in the data, an association rule may be based on a trivial dependency in the attributes, and an outlier may just be an error in the dataset. Therefore in data mining it is important to have a ground truth that can be used to verify patterns. Alternatively, one can use separate datasets for training and validating the technique, so that the performance of a certain technique can objectively be measured. Obviously, carefully choosing a correct, suitable and fair ground truth is essential for the verification of results obtained by a data mining algorithm.

When data mining techniques are applied to graph data, we speak of link mining [49] or graph mining [33]. A well-known predictive task in this context is that of link prediction: given a graph of existing nodes and edges, which edges are likely to appear (or disappear) in the future? A common descriptive graph mining task is frequent subgraph detection: given a graph, which subgraph occurs more frequently than expected, and may indicate a pattern in the graph? For each of these tasks, it is important to keep the network aspect of the data in mind: it is not only the objects and their attributes, but also the relationships between the objects that may define the knowledge that is incorporated in the data.

1.5 Thesis outline

This thesis consists of two parts. Part I deals with efficient computation of distance-related measures and properties of graphs. The algorithms and techniques introduced
in this first part help answer questions such as:

- What is the diameter of a given real-world graph? (Chapter 2)
- How can we determine which nodes form the center of a large graph? (Chapter 3 and Chapter 4)
- How can we efficiently assess the distance between two pages on the internet? (Chapter 5)
- What measures and techniques are able to identify the prominent actors in an online social network? (Chapter 6)

In Chapter 2, an algorithm for efficiently computing the exact diameter of large graphs is introduced, and a similar technique is used in Chapter 3 to also compute the eccentricity distribution. Chapter 4 provides a generalized version of the algorithms presented in the previous two chapters to efficiently compute various other distance measures including the radius, center and periphery of a graph. In each of these chapters, shortest paths in graphs are exactly computed. To speed up this process at the cost of exactness, in Chapter 5 so-called landmark strategies are discussed, which can be used to approximate the distance between two nodes with high accuracy. Finally, Chapter 6 can be seen as a case study in which the (former) Dutch online social network Hyves is considered in the context of so-called centrality measures that determine the importance of a node in a graph.

Part II of this thesis is more oriented towards data mining in information networks, as both chapters are based on data from users that are navigating the well-known free online encyclopedia Wikipedia. The second part addresses issues related to the following questions:

- How difficult is it for humans to navigate through a network of Wikipedia articles? (Chapter 7)
- What are the differences between human search strategies and algorithmic search strategies? (Chapter 7)
- What can be learned from the patterns in human navigation paths in information networks? (Chapter 8)

In Chapter 7 and Chapter 8, the studied dataset of human traversal patterns originates from the Wiki Game, an online game in which the main task is to link two given random Wikipedia articles by means of clicking the hyperlinks between these pages.
Chapter 7 focuses on both failed and successful user-generated paths in order to assess the difficulty of this path finding task, whereas Chapter 8 considers mining the successful paths in an attempt to better understand the human search strategy.

Each of the seven chapters following this introduction ends with a conclusion, summarizing the results presented in that chapter and providing suggestions for future work.

Publications

The different chapters of this thesis are based on the following peer-reviewed publications:


A full list of publications by the author can be found on page 167 of this thesis.