Department of Computer Science
Series of Publications A
Report A-2014-3

Methods for Finding Interesting Nodes in Weighted Graphs

Laura Langohr

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium CK112, Exactum building, Kumpula, Helsinki, on June 30, 2014, at 12 o'clock noon.

University of Helsinki
Finland

Supervisor
Hannu Toivonen, University of Helsinki, Finland

Pre-examiners
Leman Akoglu, Stony Brook University, NY, USA
Tapio Salakoski, University of Turku, Finland

Opponent
Dino Pedreschi, University of Pisa, Italy

Custos
Hannu Toivonen, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/
Telephone: +358 9 1911, telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-9976-2 (paperback)
ISBN 978-952-10-9977-9 (PDF)
Computing Reviews (1998) Classification: E.1, H.2.8, H.3.3, J.3
Helsinki 2014
Unigrafia

Methods for Finding Interesting Nodes in Weighted Graphs

Laura Langohr

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
Laura.Langohr@cs.helsinki.fi
http://www.cs.helsinki.fi/people/langohr/

PhD Thesis, Series of Publications A, Report A-2014-3
Helsinki, June 2014, 78+54 pages
ISSN 1238-8645
ISBN 978-952-10-9976-2 (paperback)
ISBN 978-952-10-9977-9 (PDF)

Abstract

With the increasing amount of graph-structured data available, finding interesting objects, i.e., nodes in graphs, becomes more and more important. In this thesis we focus on finding interesting nodes and sets of nodes in graphs or networks. We propose several definitions of node interestingness as well as different methods to find such nodes.

Specifically, we propose to consider nodes as interesting based on their relevance and non-redundancy or representativeness w.r.t. the graph topology, as well as based on their characterisation for a class, such as a given node attribute value. Identifying nodes that are relevant, but non-redundant to each other is motivated by the need to get an overview of different pieces of information related to a set of given nodes. Finding representative nodes is of interest, e.g. when the user needs or wants to select a few nodes that abstract the large set of nodes. Discovering nodes characteristic for a class helps to understand the causes behind that class.

Next, four methods are proposed to find a representative set of interesting nodes. The first one incrementally picks one interesting node after another. The second iteratively changes the set of nodes to improve its overall interestingness. The third method clusters nodes and picks a medoid node as a representative for each cluster. Finally, the fourth method contrasts diverse sets of nodes in order to select nodes characteristic for their class, even if the classes are not identical across the selected nodes. The first three

methods are relatively simple and are based on the graph topology and a similarity or distance function for nodes. For the second and third, the user needs to specify one parameter, either an initial set of k nodes or k, the size of the set. The fourth method assumes attributes and class attributes for each node, a class-related interesting measure, and possible sets of nodes which the user wants to contrast, such as sets of nodes that represent different time points. All four methods are flexible and generic. They can, in principle, be applied on any weighted graph or network regardless of what nodes, edges, weights, or attributes represent.

Application areas for the methods developed in this thesis include word co-occurrence networks, biological networks, social networks, data traffic networks, and the World Wide Web. As an illustrating example, consider a word co-occurrence network. There, finding terms (nodes in the graph) that are relevant to some given nodes, e.g. *branch* and *root*, may help to identify different, shared contexts such as *botanics*, *mathematics*, and *linguistics*. A real life application lies in biology where finding nodes (biological entities, e.g. biological processes or pathways) that are relevant to other, given nodes (e.g. some genes or proteins) may help in identifying biological mechanisms that are possibly shared by both the genes and proteins.

Computing Reviews (1998) Categories and Subject Descriptors:
E.1 [Data Structures]: Graphs and Networks
H.2.8 [Database Management]: Database Applications — Data Mining
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval — Clustering, Relevance Feedback, Retrieval Models, Search Process, Selection Process
J.3 [Life and Medical Sciences]: Biology and Genetics

General Terms:
Algorithms, Experimentation

Additional Key Words and Phrases:
Graphs; Networks; Graph Mining; Knowledge Retrieval; Relevance; Irrelevance; Non-redundancy; Representatives; Submodularity; Greedy Algorithm; Iterative Algorithm; Clustering; Subgroup Discovery; Gene Set Enrichment

Acknowledgements

I would like to express my gratitude to all those who made this thesis possible. In the first place this is my supervisor, Hannu Toivonen, for his excellent supervision, continuous support and valuable hints.

Many thanks to all current and former members of the Discovery team at the Department of Computer Science, University of Helsinki, Finland, especially to Lauri Eronen, Petteri Hintsanen, Kimmo Kulovesi, Atte Hinkka, Aleksi Hartikainen, and Oskar Gross for various software and test data. My gratitude also goes to several other current and former colleagues including Fang Zhou, Esther Galbrun, Doris Entner, Mikko Pervilä, Panu Luosto, Niko Välimäki, Michael Gutmann, Antti Hyttinen, Luca Martino, Sotiris Tasoulis, and Jefrey Lijffit, for our scientific and non-scientific discussions.

I am grateful to Vid Podpečan, Nada Lavrač, and Igor Mozetič from the Jožef Stefan Institute, Ljubljana, Slovenia, and Marko Petek, Kristina Gruden, Kamil Witek, Ana Rotter, and Špela Baebler from the National Institute of Biology, Ljubljana, Slovenia, for the joint research and for making my stay in Ljubljana possible. I would also like to thank Biljana Mileva Boškoska, Dragana Miljković, Matjaž Juršič, Darko Čerepnalkoski, and Borut Sluban, for our discussions over coffee and on Friday afternoons.

Many thanks to my pre-examiners Leman Akoglu and Tapio Salakoski as well as Marina Kurtén for their comments that helped to improve this thesis.

This research has been supported by, at least, the Department of Computer Science at the University of Helsinki, the Finnish Doctoral Programme in Computational Sciences (FICS), the Graduate School in Computational Biology, Bioinformatics and Biometry (ComBi), the Doctoral Programme in Computer Science (DoCS), the European Comission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract BISON-211898, the Helsinki Institute for Information Technology (HIIT), and the Algorithmic Data Analysis (Algodan) Centre of the Academy of Finland.

Special thanks I would like to give to my family and friends, in particular to my husband Dimitri, for his invaluable support and patience.

Contents

List of publications and the author's contributions

The thesis consists of this introductory part (Chapters 1–4) and the following four original publications. The scientific content of the thesis is presented primarily in these original articles, reprinted at the end of the thesis, and not in the introductory part. Articles I–III have not been included in any other theses. Article IV has been included in Vid Podpečan's PhD thesis in Slovenia [110].

Article I — A Model for Mining Relevant and Non-redundant Information, Laura Langohr and Hannu Toivonen. In *Proceedings of the 27th ACM Symposium on Applied Computing (ACM SAC '12)*, pages 451–456, ACM, 2012.

I took part in conceiving the research problem, in developing the methods, in designing and implementing the experiments, as well as in analysing the results. I also co-wrote the article.

Article II — Retrieval of Relevant and Non-redundant Nodes, Laura Langohr and Hannu Toivonen. In *Proceedings of the Workshop on Dynamic Network Analysis (DNA), in conjunction with the 12th SIAM International Conference on Data Mining (SDM '12)*, SIAM, 2012.

I made major contributions to this article, which include the development of the methods, implementation and development of the tests, analysing the results, and writing the article.

Article III — Finding Representative Nodes in Probabilistic Graphs, Laura Langohr and Hannu Toivonen. In *Proceedings of the Workshop on Explorative Analytics of Information Networks (EIN), in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD '09)*, 2009.

I participated in the development of the methods, designed and implemented the experiments, analysed the results, and took part in writing the article.

Article IV — Contrasting Subgroup Discovery, Laura Langohr, Vid Podpečan, Marko Petek, Igor Mozetič, Kristina Gruden, Nada Lavrač, and Hannu Toivonen. In *The Computer Journal, Special Issue on Discovery Science*, vol. 56 (num. 3), pages 289–303, Oxford University Press, 2013. First published online 2012.

I had the main responsibility for formulating the computational representation of the research problem, developing the methods, designing and performing the experiments, as well as for writing the article.

Chapter 1

Introduction

This thesis is about graph mining methods that discover useful information from weighted graphs where nodes represent objects or concepts and edges relations between them. Specifically, this thesis addresses the following research questions:

1. *What kind of nodes are interesting for the user?*
2. *How to find such interesting nodes in weighted graphs?*

One first has to define what kind of nodes are considered to be interesting, or in other words, worth to explore by the user. How interestingness of nodes is defined and which method is used to find interesting nodes may depend on the application at hand. If no method exists yet for finding such nodes, an appropriate graph mining method has to be developed.

Graph mining refers to methods that search a graph in order to find some information [35]. It often denotes finding interesting patterns, such as frequent subgraphs, within a graph. We focus on mining nodes and sets of nodes from graphs or networks instead. The terms graph, weighted graph and network are considered to be synonyms in this thesis.

An application area for such methods is biology. Biological networks can represent such diverse information as protein-protein interactions [101], biological pathways [80], metabolic networks [46], food webs [145], biological neural networks [140], and they can span multiple types of relationships [43]. In general, in a biological network nodes represent biological entities (e.g. genes, proteins, or pathways) and edges represent biological relations between them (e.g. a gene codes a protein, or a protein is active in a pathway). Edge weights can state, e.g., the reliability of the relationship that the edge represents (e.g. how confident is the data source, from which one obtained the relation) [43].

Given a biological network, a user might want to understand how a set of genes (nodes in the graph) is related to a specific disease (another node). Finding biological processes or pathways (other nodes in the graph) that are relevant to both, the disease and the genes, may help in identifying possible shared biological mechanisms. Another user might need to select a few genes (nodes) from a large set of genes (nodes) for further study, or want to gain some insight about the set of genes without looking at all of them. Then, finding a few genes (nodes) that are representative for the large set of genes might meet his desire. A user who wants to discover mechanisms behind the course of a disease, might need to understand which gene expression patterns are characteristic for one time point, but not for the other time points. Contrasting different time points and finding genes characteristic for a class, such as a given time point, may reveal new research hypotheses.

Further application areas include social networks where nodes represent individuals, edges relationships between them, such as friendship or collaboration, and edge weights, e.g., the strength of social relationships [54]. Graphs can also represent data traffic, such as in peer-to-peer (P2P) networks, where nodes represent computers, edges communication channels, and edge weights, e.g., the available bandwith of a channel [91]. Even the World Wide Web can be modelled as a huge graph in which nodes represent web pages, edges hyperlinks between them, and edge weights, e.g., the rarity w.r.t. other edges orginating from the same node [39]. Throughout the introductory part of this thesis we will use, for illustrating purposes, examples of a word co-occurrence network where nodes represent terms, edges pairwise term co-occurrences, and edge weights the intensity of the relationship, i.e. how often two terms co-occur [129].

In this thesis, we are interested in finding interesting nodes among those present in the given graph. Nodes are considered to be interesting if they are either

- relevant and non-redundant based on the graph topology,
- representative based on the graph topology, or
- characteristic for a class, such as a given node attribute value.

Identifying nodes that are relevant, but non-redundant to each other is motivated by the need to get an overview of different pieces of information related to a set of given nodes. As an illustrating example, consider a word co-occurrence network. Suppose a user wants to understand how the terms *branch* and *root* (two nodes) are related to each other. Finding other terms (other nodes in the graph) that are relevant to both may help to identify their shared contexts. For instance, the terms *tree*, *equation*, and *indo-*

european language represent such contexts (such as *botanics*, *mathematics*, and *linguistics*, respectively). These terms are relevant as each of them co-occurs with both, *branch* and *root*. They are non-redundant, because they represent different contexts. Hence, these terms may be considered interesting.

Finding representative nodes is of interest, e.g. when the user needs to select a few nodes which abstract a large given set of nodes. For instance, in the word co-occurrence network, given the terms *maple*, *birch*, *aspen*, and *pine*, the term *birch* is representative as it is a typical example among them.

Discovering nodes characteristic for a class might give the user new insights about that class. Consider again nodes representing terms. It may be the case that nouns describing organisation units such as *branch*, *section*, and *division* are characteristic for texts written by managers. Further, nouns in general might be characteristic for different groups of authors as different groups of authors tend to use different sets of nouns.

This thesis defines the tasks of finding nodes that are relevant and non-redundant, or representative based on the graph topology (i.e. nodes, edges, and edge weights), or characteristic for a class based on their node attributes; proposes definitions for interestingness of such nodes; and gives relatively simple methods for identifying such nodes.

The graph mining methods presented here find interesting nodes in one of the following four ways:

- by incrementally picking one interesting node after another,
- by iteratively changing the set of nodes to improve its overall interestingness,
- by clustering nodes and picking a medoid node as representative for each cluster, or
- by contrasting diverse sets of nodes in order to find nodes characteristic for a class, even if that class is not identical across the selected nodes.

All four presented methods are flexible and generic. They can in principle be applied on a weighted graph or network regardless of what nodes, edges or weights represent. The first three methods are relatively simple and are based on a similarity or distance function for nodes. For the second and third, the user needs to specify one parameter, either an initial set of k nodes or k, the size of the set. The fourth method assumes more information about the nodes such as attributes for each of them.

This introductory part of the thesis is not a review of our research work. It rather gives the background and context of the research work presented in Articles I-IV by describing the scientific field and placing the articles'

contents within it. Further, it depicts the problems addressed in Articles I-IV and describes how they propose to solve these, i.e. summarises the scientific contributions of those articles.

The rest of this thesis is structured as follows. In Chapter 2, we describe graphs and how interestingness of nodes can be defined within them, followed by a review on methods for finding interesting nodes. Afterwards, we discuss our contributions in Chapter 3. Chapter 4 concludes the introductory part with answers to the research questions and some notes about future work, followed by the body of the thesis, the four original publications.

Chapter 2

Background

In this chapter we first give an overview of how graphs can be used to model various types of information, including basic notations as well as different measures for node similarity and distance based on the graph topology. We then describe different ways of defining what interesting nodes are. After that we briefly review existing methods for finding such nodes in graphs. A full review of graph mining, interestingness measures, and methods for finding interesting nodes is outside the scope of this thesis.

2.1 Graphs as an information model

Information is often modelled as a graph of objects: think of social networks, biological networks, word co-occurrence networks, or the World Wide Web, for instance. Next we give some basic notations of graphs and related properties. Afterwards, we show how node similarity and distance can be defined.

2.1.1 Basic graph notations

Formally a graph is denoted as $G = (V, E)$ where V is the set of *nodes* (or vertices) and $E \subseteq V \times V$ is the set of *edges*. Each edge links two nodes and is said to be *incident* to those two nodes.

Graphs can be *directed* or *undirected*. In a directed graph each edge $e \in E$ connecting two nodes u and v is an ordered pair (u, v), whereas in an undirected graph it is an unordered pair $\{u, v\}$. For example, the World Wide Web is often modelled as a directed graph as the hyperlinks point from one web page to another. Word co-occurrence networks again are in many models undirected, because if term u co-occurs with term v in the same sentence, then also term v co-occurs with term u in that sentence.

Biological networks can be directed (a gene codes a protein) or undirected (a protein interacts with another protein).

A *heterogeneous* graph is a graph where nodes and/or edges are of multiple types [124]. For example, biological networks often are heterogeneous: nodes can represent genes, proteins, pathways, etc. and edges different relations between them such as coding, participation, interactions, etc. A biological network can also be homogeneous as in the case of protein interaction networks [66], where nodes represent only proteins and edges only interactions between two proteins.

A graph is called *weighted* if a weight $w(e), w : E \to \mathbb{R}^+$ is assigned to each edge $e \in E$. Edge weights often represent the intensity of the relationship (e.g. how often two terms co-occur) [129], or the length of the edge (e.g. the length of a road between two intersections) [35]. They can also state the reliability of the relationship that the edge represents (e.g. how confident is the data source, from which one obtained the relation) [43].

A graph is called *probabilistic* if the edge weights $w(e), w : E \to [0, 1]$ represent probabilities (or uncertain relations), such as the probability that the relationship exists [120]. These can be used, e.g. to model the informativeness and reliability of the relationships in a biological network [43]. Further, such settings arise in probabilistic or uncertain databases [29, 37].

A graph $G' = (V', E')$ is a *subgraph* of graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.

Given a node $u \in V$ the set of its *neighbouring* or *adjacent* nodes is defined as $\Gamma(u) = \{v : \{u, v\} \in E\}$. The size of that set $|\Gamma(u)|$ is the *degree* of node u, which is equal to the number of edges connected to u. Similarly, on a directed graph the set of neighbouring nodes, which node u is pointing to and pointed by, can be defined as $\Gamma_{out}(u) = \{v : (u, v) \in E\}$ and $\Gamma_{in}(u) = \{v : (v, u) \in E\}$ respectively. The number of nodes in these sets are the *outdegree* $|\Gamma_{out}(u)|$ and the *indegree* $|\Gamma_{in}(u)|$ of node u.

A (undirected) *path* P between two nodes u_1 and u_k consists of subsequent nodes and edges and can be specified by its edges: $P = \{e_1, \ldots, e_k\}$ $= \{\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{k-1}, u_k\}\} \subseteq E$.

Two nodes are *connected* if there exists a path between them. A graph is connected if every two nodes of the graph are connected.

The *length* $len(P)$ of a path P is the number of edges along the path.

The *shortest path* $sp(u, v)$ between two nodes u and v is then the path with the minimum number of edges from node u to node v [35]. Alternatively, the path from node u to node v which minimises the sum of the edge lengths along that path can be considered as the shortest path [76].

Given a probabilistic graph, the probability $prob(P)$ of a path is the

product of the probabilistic edge weights $w : E \to [0, 1]$ along that path [88]:

$$prob(P) = \prod_{e \in P} w(e).$$

The *best path* $bp(u, v)$ between two nodes u and v is then the most probable path among all paths from node u to node v.

We can now provide definitions of typical node similarity and distance measures, which are based on the graph topology.

2.1.2 Node similarity and distance

Similarity and *distance* of nodes based on the graph topology can be defined in various ways. For example, the similarity between two nodes might be considered to be high and the distance between them to be low, if there is an edge adjacent to both of them; if both nodes occur in the same cluster; or if their neighbourhoods are identical, equivalent, or similar [21, 127].

Whether we consider distance or similarity between nodes is not a crucial issue. In general, if the similarity between two nodes is high, then the distance between them is low and vice versa. Hence, if a similarity function $s : V \times V \to \mathbb{R}^+$ or distance function $d : V \times V \to \mathbb{R}^+$ is given, the other one can be identified, e.g. with the inverse $s(u, v) = 1/d(u, v)$ for all $u, v \in V$, with the exception that $s(u, v) = \infty$ if $d(u, v) = 0$.

A simple definition of distance $d(u, v)$ between two nodes is the length $len(sp(u, v))$ of the shortest path $sp(u, v)$ between them:

$$d(u, v) = \begin{cases} 0 & \text{if } u = v \\ len(sp(u, v)) & \text{if } u \neq v \text{ and they are connected} \\ \infty & \text{else.} \end{cases} \tag{2.1}$$

Alternatively, in a probabilistic graph the similarity of two nodes u and v can be defined as the probability of the best path:

$$s(u, v) = \begin{cases} 1 & \text{if } u = v \\ prob(bp(u, v)) & \text{if } u \neq v \text{ and they are connected} \\ 0 & \text{else.} \end{cases} \tag{2.2}$$

This is a simple but relatively efficient lower bound of the probability that u and v are connected, a measure known as network reliability [34].

Node similarity measures can also be based on node neighbourhoods. Such similarity measures include e.g. the relative number of common neighbours [62, 87], Adamic/Adar [1, 87] and preferential attachment [13, 87].

More complex distance or similarity measures are based, e.g. on maximum flow between nodes [56, 35], or random walk [106, 131]. In particular, random walk with restart (RWR) provides a score for how closely two nodes are related to each other in a weighted graph [131]. The standard random walk starts from a node u and then iteratively moves from the current node to a neighbouring one. The (transition) probability of choosing any particular edge to follow is proportional to the edge weight [133]. In a RWR, the random walker will at each step return to the original node u with a probability d. The similarity of two nodes u and v can then be defined as the steady-state probability that the random walker who started from u will be at node v. Formally, it can be calculated by

$$s(u,v) = (1-d) \sum_{x \in \Gamma(u)} \frac{s(x,v)}{|\Gamma(x)|} + d\ p(u), \tag{2.3}$$

where d is the probability to return to node u, $p(v) = 1$, and $p(u) = 0$ for $u \in V, u \neq v$. This is a special case of the personalised PageRank [58, 133] which we will describe later (Equation 2.9 in Section 2.2.1).

Next we describe various ways of defining interestingness of nodes in graphs, which are often based on pairwise node similarities or distances. We also briefly comment on interestingness of edges and subgraphs.

2.2 Interesting nodes

Given data represented as a graph, a user might want to find interesting nodes or sets of interesting nodes. Interestingness of nodes can be defined in various ways. In this thesis we are interested in finding nodes that are *relevant* and *non-redundant*, or *representative* based on the graph topology, or *characteristic for a class*, e.g. a given node attribute value. To look in particular at these types of interestingness is a choice we made for this thesis.

In the following sections we review existing definitions for each of these separately. Thereby we unify the terminology and equations to make them easily comparable. In addition we briefly review related problems, where the aim is to find interesting objects, but which are not modelled as graph mining problems.

2.2.1 Relevance

Some nodes in a given graph might be more relevant than others. This is the case, e.g. when a user is interested only in some parts of the information

represented by the graph. Consider a word co-occurrence network of millions of terms (nodes) and a user who wants to understand how the terms *branch* and *root* are related to each other. Then, the terms *tree*, *maple* and *birch* represent a set of relevant terms as each of them may connect the two given terms, whereas other terms such as *cow*, *walk* and *fun* are not.

The definition of node, edge, or subgraph relevance might be based on the graph topology alone (*global relevance*), or alternatively, on the graph topology w.r.t. some given nodes (*relative relevance*).

Next we describe global relevance in more detail and present typical measures for it, followed by a review of relative relevance and its measures.

Global relevance of nodes. The global relevance of a node is defined by the topology of a given graph alone. For example, one might be interested in finding nodes in a graph which are relevant w.r.t. all other nodes [142].

Consider again the word co-occurrence network of terms. There, the terms *tree* and *indo-european language* might be connected to many other terms and hence be considered as relevant terms w.r.t. the whole graph. In contrast, the terms *birch* or *English* might be connected to few other terms and considered as less relevant.

In biological networks central genes or proteins are those that are evolutionarily well conserved [148] and whose removal or disruption causes lethality [44, 53]. Finding central proteins offers to identify which proteins serve as the evolutionary backbone within all proteins of an organism [148].

In general, the *centrality* of a node is a typical measure for its global relevance [48, 67] and various definitions of node centrality exist (for an overview see, e.g. [36]). Centrality can be viewed to measure the *influence* of the node [142]. Identifying influential persons in social networks is of interest, e.g. if a company wants to sell a new product, but can address the advertisement only to a limited amount of people.

The centrality of a node is often defined by its degree (*degree centrality*) [48]:

$$rel(u) = |\Gamma(u)|, \tag{2.4}$$

which can be normalised by the number of nodes in the graph not including u in order to rescale the centrality values such that $rel(u) \in [0, 1]$. Thus, nodes with many edges are considered more central.

Alternatively, one can define the centrality of a node as the number of the shortest paths between any other two nodes in the graph that go

through this node (*betweenness centrality*) [47, 20]:

$$rel(u) = \sum_{\substack{v,w \in V \\ u \neq v \neq w}} \frac{|\{sp(v,w) : sp(v,w) \text{ passes through } u\}|}{|sp(v,w)|}, \tag{2.5}$$

which can be normalised by the number of pairs of nodes not including u. That is, a node that lies on a high fraction of shortest paths connecting pairs of nodes is considered more central. A node which has few neighbours (low degree centrality) can still be crucial for many pairwise connections between nodes (high betweenness centrality).

The centrality of a node can also be defined as the inverse sum of shortest distances from the node to all other nodes (*closeness centrality*) [117, 20]:

$$rel(u) = (\sum_{v \in V} d(u,v))^{-1}, \tag{2.6}$$

which can be normalised by the number of nodes. That is, a node which in average is close to many other nodes in the graph is considered more central. A node can have a low degree centrality, but high closeness centrality, e.g. if its neighbours have high degree centrality. The betweenness centrality of such a node is high, if it is in a crucial position w.r.t. many shortest paths, and it is low, if many alternative shortest paths exist.

These centrality measures have been generalised for weighted and directed graphs [14, 20, 36, 103, 141].

The PageRank [22] of a node is a measure of importance in the setting of directed graphs such as the World Wide Web. A node u has a high PageRank if the sum of the ranks of the nodes $v, (v,u) \in E$ is high. That is, either u has many nodes that link to it, it has a few, but highly ranked nodes linking to it, or something between these two extreme cases [22, 106]. The PageRank of a node u is defined as

$$rel(u) = (1-d) \sum_{v \in \Gamma_{in}(u)} \frac{rel(v)}{|\Gamma_{out}(v)|} + d\ \frac{1}{|V|}, \tag{2.7}$$

where $d \in [0,1]$ is a damping factor. If PageRank is thought of as random walk, then the PageRank of a node u is the steady-state probability that the random walker will be at node u and the damping factor d is the probability that the random walker jumps from a node to another random node. Hence, the PageRank of a node is influenced by the PageRank of nodes which point to it. The PageRank has also been generalised for weighted graphs [109].

In the context of web page ranking, another well known approach is Hypertext Induced Topic Selection (HITS) [73]. There, the aim is to find

authority nodes (web pages) relevant to a specific (search) topic. Authorities are those nodes that are pointed to by several *hub* nodes, and hubs are those nodes that link to several authority nodes. Formally, the authority and hub scores of a node u can be defined as

$$\begin{aligned} rel_{aut}(u) &= \sum_{v \in \Gamma_{in}(u)} rel_{hub}(v) \\ rel_{hub}(u) &= \sum_{v \in \Gamma_{in}(u)} rel_{aut}(v), \end{aligned} \tag{2.8}$$

which are normalised such that their squares sum up to one, i.e. $\sum_{v \in V} rel_{hub}(v)^2 = 1$ and $\sum_{v \in V} rel_{aut}(v)^2 = 1$. Then, the web pages (nodes) with a high authority score can be defined to be the most relevant web pages (nodes). Authorities can also be of interest in the task of understanding information propagation in a social network [8]. There, one might distinguish between peers (e.g. friends) and authorities (e.g. movie stars) that influence an individual differently. In general, node rankings produced by PageRank as well as HITS correlate with ranking nodes by their in-degree [39]. Further, both methods can be generalised and combined into a unified framework.

In a heterogeneous network nodes of different types can be ranked differently [124]. This is of interest, e.g. in a co-authorship network, where authors (nodes) are relevant if they published many papers in highly ranked conferences (nodes of another type) or if they co-authored papers with many highly ranked authors. Conferences again are relevant if many papers from highly ranked authors are published there.

Related problems outside the scope of this thesis include finding all missing or future nodes and edges [87, 72], finding anomalous nodes [123, 7], edges [25] or subgraphs [105], or finding relevant edges or subgraphs [51, 130, 38] in a given graph. In contrast, we are interested in finding interesting nodes among those present in the given graph.

Relative relevance of nodes. The relative relevance of nodes is defined by the topology of the given graph as well as some given nodes. These nodes can be used to indicate subjective interestingness in the objects the nodes represent, the connections between them, and much more.

Consider again the word co-occurrence network as an example. Then, a single term such as *root* might be given by the user. The relevance of other terms might depend on whether they often co-occur with the term *root* or not. The terms *tree*, *maple* and *birch*, e.g. might constitute a set of relevant terms w.r.t. the term *root*.

In biology a common problem is that high-throughput techniques associate several genes with a disease or trait. Given a heterogeneous biological

network, finding biological processes or pathways (nodes in the graph) that are relevant both to the disease and the given genes (nodes obtained by the high-throughput techniques) helps to understand how they are related and may help identify possible shared biological mechanisms.

From now on, we will refer to the given nodes as *(positive) query nodes*, or depending on the example at hand, as query terms, query genes, and so on. Note that we refer to these as query nodes specified by the user though they might have come from somewhere else than a query, such as a data mining step or a program.

Formally, relative relevance of nodes can be based on node similarity. That is, one can be interested in finding nodes with high similarity to the given query nodes. (We discussed node similarities in Section 2.1.2.)

Further, relative relevance of a node u w.r.t. a (positive) query node q might be defined by a set of paths connecting u with q (e.g. all of them, or the k shortest paths) [142]. There, longer paths contribute less than shorter ones to the relevance. Formally, such a relevance can be defined by

$$rel_P(u,q) = \sum_{P \in \mathcal{P}(u,q)} \alpha^{-len(P)}.$$

That is, a node u is highly relevant to q if there exist many paths ($\alpha = 1$) or many short paths ($\alpha > 1$) between u and q. The relative relevance of a node u w.r.t. a set of (positive) query nodes Q_p can then be defined as the average relevance relative to that set Q_p

$$rel_P(u,Q_p) = \frac{1}{|Q_p|} \sum_{q \in Q_p} rel_P(u,q)$$

or, alternatively, in order to require node u to have high relevance to each query node $q \in Q_p$, as the minimum relevance

$$rel_P(u,Q_p) = min\{rel_P(u,q) \mid q \in Q_p\}.$$

Personalised PageRank (PPR) [58, 65] and personalised HITS [27, 142] rank Web pages not only according to the graph topology of the World Wide Web, but in addition take the query terms into account. Specifically, the personalised PageRank can be defined as:

$$rel(u) = (1-d) \sum_{v \in \Gamma(u)} \frac{rel(v)}{|\Gamma(v)|} + d\ p(u), \tag{2.9}$$

where $p(u) \geq 0$, $\sum_{u \in V} p(u) = 1$ is the personalised score of node u [58]. The damping factor d specifies how strongly the personalised PageRank is

biased towards the personalised scores. In the special case where $p(x) = 1$ for a node $x \in V$ and $p(u) = 0$ for all other nodes $u \in V, u \neq x$, the personalised PageRank $rel(u)$ gives the similarity between nodes u and x (as described before in Equation 2.3 in Section 2.1.2).

There are some settings which are closely related to, but slightly differ from our setting. For instance, in the setting of recommendation systems the aim is to identify new products that are either similar to products currently liked by the user (content-based recommendation) [64] or have been liked by similar users (collaborative recommendation) [12]. In information retrieval (IR) a classical problem is to identify a set of relevant objects (typically documents) w.r.t. some query entities (typically terms) [99, 93]. In both problems, the selection of objects is typically based on the information contents of objects. Hence, they differ from the setting we consider in this thesis, where relevance is based on the graph topology. Further, we specify queries by nodes, not by keywords.

Related problems outside the scope of this thesis include finding relevant edges or subgraphs w.r.t. some given nodes. For instance, one can be interested in finding unusual relationships [89], or in capturing the relationship or the connectivity between given nodes [45, 61, 6]. In contrast, we are interested in finding interesting nodes w.r.t. some given nodes.

2.2.2 Non-redundancy

Given a set of relevant nodes, these nodes can be very similar or *redundant* to each other based on the graph topology. To ensure diversity within the resulting set of nodes, these nodes should not only be relevant, but they should also be mutually *non-redundant* or complementary to each other.

Consider again the word co-occurrence network. Given the term *root*, the terms *tree*, *maple* and *birch* constitute a set of relevant terms w.r.t. *root*, but they are redundant to each other as they all represent trees. On the other hand, the terms *tree*, *equation*, and *indo-european language* constitute a set of terms that are relevant w.r.t. *root*, but represent a diverse set of contexts (*botany*, *mathematics*, and *linguistics*, respectively).

Given a biological network, finding biological processes or pathways that are non-redundant, i.e. finding a more varied result, is probably more useful for the biologist as it can represent several different hypotheses.

A variant of random walk with restart has been proposed to directly address non-redundancy [132]. There, relevance is measured by the personalised PageRank, and non-redundancy by the adjacency matrix (or the personalised adjacency matrix, which is biased towards the query vector) weighted by the personalised PageRank vector. Specifically, the overall rel-

evance and non-redundancy of a set of (retrieved) nodes $R \subseteq V$ is measured by the difference of its relevance and redundancy:

$$REL(R) = rel_P(R) - red(R), \tag{2.10}$$

where $rel_P(R)$ and $red(R)$ are defined as follows. The relevance of the set R of nodes is the sum of the individual relevances of nodes $u \in R$:

$$rel_P(R) = 2 \sum_{u \in R} rel_P(u, Q_p), \tag{2.11}$$

where the relevance $rel_p(u, Q_p)$ is the personalised PageRank of node u. That is, a set R of nodes has high relevance if the nodes within R have a high PageRank which was biased by Q_P. The redundancy of the set R is

$$red(R) = \sum_{u,v \in R} \alpha(u, v)\ rel_P(v, Q_p) \tag{2.12}$$

where $\alpha(u, v)$ is the element at the uth row and vth column of the row-normalised adjacency matrix of the graph. That is, the redundancy of a set R of nodes is low if edges (u, v) that link nodes $u, v \in R$ have low weight and the terminal nodes v of such edges have a low personalised PageRank.

In IR the aim is to retrieve documents from various categories where queries and documents may belong to more than one category [52, 19, 3]. Finding documents that are relevant (e.g. similar) w.r.t. query terms, but at the same time non-redundant to each other, can be achieved, e.g. by balancing the similarity (relevance) between documents and the query and the inverse similarity (non-redundancy) among documents [23]. These relevance and non-redundancy measure have been adapted for the setting of query nodes in a graph [96] as follows. The relevance and non-redundancy are balanced by their difference:

$$REL(u, Q_P, R) = rel_P(u, Q_P) - red(u, R), \tag{2.13}$$

where the relevance of a node u w.r.t. to the set of positive query nodes Q_P is measured by their similarity:

$$rel_P(u, Q_P) = \alpha\ s(u, Q_P)$$

and the redundancy of a node u w.r.t. the set of previously retrieved nodes R is measured by the similarity of the most similar node within R:

$$red(u, R) = (1 - \alpha)\ \max_{v \in R} s(u, v).$$

Several subsequent IR approaches find relevant and non-redundant documents by preferring documents that cover most categories of a query over those that cover only a few [150, 32, 3]; by generating related queries [114]; by learning an ordering of search results from a diverse set of orderings based on user feedback [115]; or by maximising the probability of retrieving at least one relevant document among the top k documents [28, 3].

Graph-based approaches for word sense disambiguation identify the different senses (or meanings) of a term based on a word co-occurrence network [100]. For example, the term *bank* can denote a building or a sloping land amongst other meanings. Typically terms (nodes) adjacent to each other are considered to be redundant [139, 2]. Non-redundancy can also be addressed indirectly when terms in the word co-occurrence network are clustered and terms within a cluster are considered to be redundant [42].

Non-redundancy has also been defined in recommendation systems [121]. As stated in Section 2.2.1, the setting differs from the one we consider here. Interestingness in these problems is defined on the information contents of objects (products) and not on the graph topology. We next briefly describe how non-redundancy is defined in this setting based on a pairwise similarity measure, an idea that can be used also in graphs. The aim is to find a set of recommendations that are relevant (e.g. similar) to a given product or set of products Q_P, but at the same time non-redundant to each other. Then, the overall relevance of a product can be defined as the product of its relevance to Q_P and its non-redundancy to a set of recommended products R [121]:

$$REL(u, Q_P, R) = rel_P(u, Q_P)\ (1 - red(u, R)), \tag{2.14}$$

as their difference:

$$REL(u, Q_P, R) = \alpha\ rel_P(u, Q_P) + (1 - \alpha)\ (1 - red(u, R)), \tag{2.15}$$

where the parameter α is again used to weight either the relevance or non-redundancy more strongly, or as their harmonic mean (i.e. the reciprocal of the arithmetic mean of the reciprocals):

$$REL(u, Q_P, R) = 2\ (rel_P(u, Q_P)^{-1} + (1 - red(u, R))^{-1})^{-1}. \tag{2.16}$$

The relevance of a node u w.r.t. to the set of given products Q_P is measured, independently of which overall relevance is chosen, by their similarity:

$$rel_P(u, Q_P) = s(u, Q_P).$$

The redundancy of a node u w.r.t. the set of previously retrieved nodes R is measured by the average similarity to the nodes in R:

$$red(u, R) = \frac{1}{|R|} \sum_{v \in R} s(u, v). \tag{2.17}$$

The redundancy of a set of products has also been defined as the sum of pairwise similarities among products in the set [153]:

$$red(R) = \frac{1}{2} \sum_{u \in R} \sum_{v \in R} s(u, v). \tag{2.18}$$

Further, non-redundancy has been addressed in various other settings. It can be used, e.g. in drug discovery in order to find a set of diverse chemical compounds [119], in automated planning in order to find a diverse set of solutions (or plans) [122], and for simplifying a given graph [130].

2.2.3 Irrelevance

Some nodes in a given graph might be uninteresting or *irrelevant* to the user. This is the case, e.g. when a user is interested only in some parts of the information represented by the graph. Remember that in the setting of relative relevance the user can specify positive query nodes to indicate what she is interested in and other nodes are then considered less relevant. Here we consider the case where the user can explicitly specify some *negative query nodes* to indicate what she is *not* interested in.

Consider again the word co-occurrence network and a user who is not interested in mathematics. When the term *mathematics* is specified as a negative query term the terms *equation* and *formula* are quite irrelevant.

Given a biological network, the user might specify well-known results as negative query nodes, and thus guide the mining process towards novel and therefore more interesting results.

In general, the desirable effect of irrelevance is similar to non-redundancy: a node close to a negative query node is irrelevant, just like a node close to another node is mutually redundant.

A variant of random walk considers positive as well as negative query nodes [133]. There, the user can specify one positive query node, a set of favourable nodes, and a set of negative query nodes. Then, the original graph structure is refined as follows. Edges are added from the positive query node to each favourable node, and their weights are determined by the number of neighbouring nodes as well as by the number of positive query nodes. Further, edges are added from each negative query node and k neighbouring nodes to a newly introduced sink node. The k neighbouring nodes as well as the edge weights for all edges incident to the sink node are determined by a random walk with restart on the original graph. Then, a random walk of restart is performed on the refined graph in order to determine the relevances from the positive query node to all other nodes in the graph. Such a random walk of restart on the refined graph will

more likely visit favourable nodes and their neighbours, and less likely visit negative query nodes and their neighbours, than a random walk of restart on the original graph. Hence, node relevances w.r.t. the positive query node are increased for favourable nodes and their neighbours, and decreased for negative query nodes and their neighbours.

Negative (query) objects have been addressed in various other settings as well. For instance, in IR documents containing a negative query term are often assumed to be least interesting for the user [79]. Though, remember that this problem differs from the problem considered in this thesis (as stated before in Sections 2.2.1 and 2.2.2). In IR, queries are specified by terms, whereas we assume queries are specified by query nodes in a graph.

In the setting of automated planning, a user might prefer to specify not only positive, but also negative query constraints in order to state what has to be rejected rather than what has to be accepted [136]. However, this is typically not viewed as a graph mining problem.

In the setting of link prediction, it has been proposed to assign edges a negative weight if they represent foes or distrust in social networks [85]. In contrast to link prediction which aims to predict missing or future edges, we are interested in finding interesting nodes present in the given graph. Further, we assume only positive edge weights.

2.2.4 Representativeness

A node is representative, e.g. if it is a typical example in a large set of nodes.

Consider again the word co-occurrence network. Given the terms *maple*, *birch*, *aspen* and *pine*, the term *birch* is representative as it is a typical example among them.

In a biological network a gene might be representative for a set of genes if it is a typical instance of those genes. It can give the biologist some insight about the set of genes without looking at all of them.

The representative for a subset V_i of nodes can be defined to be the *medoid*, i.e. the node of the subset for which the sum of distances to all other nodes in the subset is the smallest [68]. That is, it can be defined to be the node that maximises either the closeness centrality (see Equation 2.6) or, the sum of similarities to all other nodes in the subset:

$$rep(u, V_i) = \sum_{v \in V_i} s(u, v) = \sum_{v \in V_i} d(u, v)^{-1}. \tag{2.19}$$

Alternatively, the representativeness of a node for a given subset of nodes can be measured by its degree (Equation 2.4), betweenness centrality (Equation 2.5), its HITS score (Equation 2.8) [147], or by the relative

number of neighbours which also belong to that subset [77]:

$$rep(u, V_i) = \frac{|\{v : v \in \Gamma(u), v \in V_i\}|}{|\Gamma(u)|}. \tag{2.20}$$

Further, a node can be defined to be representative for a subset of nodes if it is close to that subset, but far away from other subsets on average [134]. Formally, given a partition $V_1 \cup \ldots \cup V_k = V$ of the nodes in a graph, the representativeness of node u for a subset V_i can be measured by

$$rep(u, V_i) = \frac{1}{|V_i|} \sum_{v \in V_i} s(u, v) \cdot \prod_{\substack{j=1\ldots n \\ i \neq j}} \Big(1 - \frac{1}{|V_j|} \sum_{v \in V_j} s(u, v)\Big), \tag{2.21}$$

where $s(\cdot) \in [0, 1]$. That is, the first multiplicand is the normalised variant of Equation 2.19 and measures how central node u is for the subset V_i. The second multiplicand measures how discentral node u is to all other subset V_j, and is large if u is not central to any other subset.

This representativeness measure (Equation 2.21) has been used, e.g. to find authors (nodes) and scientific articles (nodes of another type) representative for sets of years (sets of nodes of yet another type) [134]. Then, the titles of the representative articles of different periods of times may reveal that there was a topic shift in between.

Neither do typical settings of two related problems, finding representative objects and finding representative features, consider data in the form of graphs, but assume a data set of objects associated with a set of features [16, 108]. Then, representativeness may be measured based on information gain [108], or a correlation coefficient [55]. Alternatively, features are grouped into sets of similar features and for each set one feature (e.g. the medoid) is identified as a representative for that set [68].

Objects can also be selected as representatives such that they are uniformly distributed over the space [70]. In addition, various other measures exist that depend not only on features associated to the objects, but also assume that class attributes are specified [16, 63]. The quality of a subset of features is often measured by its performance on a classification task, i.e. representativeness is measured only indirectly [55]. Alternatively, the quality of a feature subset can be measured by a representativeness measure which is based on the features alone, and hence is independent of the method used for finding such subsets [90].

In both problems, finding representative objects or features, representatives are used to reduce the number of objects or features, respectively. One typical application for this is to eliminate irrelevant and redundant

objects, e.g. in order to reduce computational complexity or prevent overfitting when using other methods [90, 149]. Similarly, representative nodes can be used to reduce the number of nodes of a given graph [147]. This can be helpful, e.g. for providing quick approximate computations on the graph, for visualising the graph, or for identifying underlying patterns.

Notice that nodes can be representative as well as non-redundant, if they are representative for distinct subsets of nodes. In contrast, two nodes can be non-redundant, but *not* representative, if they are distant from each other, but there exist no subsets of nodes they represent.

Further, one can be interested in sampling a subgraph from a large graph, such that it maintains some topological properties (e.g. node degree distribution) of the original graph [36, 84]. Alternatively, finding representative graphs might provide, e.g. a summary of large ensembles of graphs [11]. In contrast, we are only interested in finding interesting nodes.

2.2.5 Characteristic for a class

A node is characteristic for a class if it is a member of a subset of nodes that share a combination of node attribute values and a class-related interestingness measure of that set lies above a given threshold. This is a typical problem in bioinformatics, where the aim is to find enriched gene sets (interesting subsets of nodes), e.g. for virus infected samples. Notice that here we take node attributes and class attributes into account, but not the graph topology. The results can then be visualised in a graph and used for other graph mining methods [111].

Consider again as an example nodes representing terms. The terms *branch*, *section*, and *division* are all nouns and may be used in the same sense: a unit of an organisation. Formally, assume terms described by two node attributes, word category (e.g. the term *branch* can be used as a noun or verb) and word sense (e.g. the noun *branch* can be used in the sense of a part of a tree or organisation unit), are given. Then, the set $T' = \{$*category=noun, sense=organisation unit*$\}$ of node attribute values might define the subset of terms $V_{T'} = \{$*branch, section, division*$\}$ as they share the attribute values within T'.

Let $classes : \mathcal{P}(V) \to \mathbb{Z}_+ \times \mathbb{Z}_+$ be a function that gives the distribution for a class $c \in T$ of a given set $V_{T'} \subseteq V$ of nodes, where $\mathcal{P}(V)$ is the powerset of the nodes V. That is, $classes(V_{T'})$ gives the number of nodes in $V_{T'}$ annotated by c and the number of nodes in $V_{T'}$ not annotated by c.

Consider again nodes representing terms described by the node attributes word category and word sense. Assume further, different texts are given, of which some are written by managers. A user might specify

the node attribute value *author=manager* as the class she is interested in. Then, $classes(V_{T'})$ gives the number of terms (nodes) in $V_{T'}$ which occur in texts written by managers and the number of terms (nodes) in $V_{T'}$ which do not occur in those texts. Similarly, $classes(V \setminus V_{T'})$ gives the numbers of other terms that occur in texts written by managers and not.

Then, the *class-related interestingness* of a set of node attribute values can be defined as

$$\begin{aligned} f_c : \quad & \mathcal{P}(T) \to \mathbb{R}, \\ & T' \mapsto g(\text{classes}(V_{T'}), \text{classes}(V \setminus V_{T'})) \end{aligned} \tag{2.22}$$

where $\mathcal{P}(T)$ is the powerset of node attribute values T and $V_{T'} = V_1 \cap \cdots \cap V_k \subseteq V$ is the subset of nodes which share the attribute values $T' = \{t_1, \ldots, t_k\} \subseteq T$.

In other words, f_c is a function $g(\cdot)$ of the class distributions within and outside of the subset of nodes. The exact definition of function $g(\cdot)$ varies from one problem variant to another, but the common denominator is that it is based on the class distributions alone. Often the subsets are analysed by statistical tests, like Fisher's exact test, the χ^2 test, or the Binomial probability. (For more information, see, e.g. [33]). Alternatively, heuristics such as the weighted relative accuracy [82] or the generalisation quotient [49] can be used. Without loss of generality, we assume small values of $f_c(\cdot)$ indicate high interestingness.

A subset of nodes defined by a set $T' \subseteq T$ of node attribute values is characteristic for the class $c \in T$ if the class-related interestingness $f_c(T')$ is small. Formally, this can be defined as $f_c(T') \leq \alpha$ for a given constant α, or one can identify the k best subsets of nodes instead of using a fixed threshold.

Consider again nodes representing terms which are described by the attributes word category, word sense and authorship. Assume $classes(V_{T'})$ gives the number of terms (nodes) in $V_{T'}$ which occur in texts written by managers and those which are not. Let the class-related interestingness of a set $V_{T'}$ be determined by Fisher's exact test based on the class distribution. Then, the interestingness of a set $T' = \{$*category=noun, sense=organisation unit*$\}$ of node attribute values that define the subset of terms $V_{T'} = \{$*branch, section, division*$\}$ is probably high if the terms *branch*, *section*, and *division* are frequently used in texts written by managers, occur infrequently in texts written by other authors, and other terms divide more or less evenly across the different texts. Hence, the terms *branch*, *section*, and *division* that share the attribute values *category=noun* and *sense=building* are characteristic for texts written by managers.

Finding such subsets of nodes which are characteristic for a class can be of interest in various domains. For instance, given a social network where nodes represent patients and node attributes their medical records, one can aim to find subsets of patients with a specific diagnosis that share a combination of medical examination results or living habits [49, 10]. The identified diagnostic patterns can, e.g. supplement medical consultation systems [10].

Given a graph where nodes represent traffic accidents, the aim can be to find subsets of accidents of a specific severity that share a combination of attribute values such as vehicles involved, number of people injured, road class, speed limit, or light conditions [82]. Finding such sets of traffic accidents may reveal unexpected relations, e.g. that serious and fatal accidents often involve only a single vehicle.

Other application domains include census data [75], vegetation data [94], telecommunications [83], and chess endgame positions [83], amongst others.

There are closely related topics where the interestingness of node attribute value combinations is defined in a slightly different manner. For instance, one might be interested in finding *frequent item sets*, i.e. frequent combinations of attribute values, such as *category* = *noun* $\wedge$ *sense* = *organisation unit* [4, 98]. One might also be interested in finding *emerging patterns*, i.e. item sets for which the support increases significantly from one class to another [41]. Further, one might be interested in finding *association rules*, such as $X \mapsto Y$, where the antecedent X and consequent Y are item sets [5]. In categorical data the antecedent and consequent are (attribute, attribute value) pairs such as *sense* = *organisation unit* $\mapsto$ *category* = *noun* [15, 18]. Alternatively, one might be interested in finding *exception rules*, i.e. unexpected association rules which differ from a highly frequent association rule [125]. That is, unexpected association rules are rules such as $X \wedge Z \mapsto Y$, where $X \mapsto Y'$ and $Z \not\mapsto Y'$. Here, X and Z are item sets or (attribute, attribute value) pairs, and Y and Y' are different (class attribute, class) pairs. Consider e.g. X to be *sense* = *organisation unit*, Z to be *category* = *noun*, Y to be *author* = *manager*, and Y' to be *author* = *contract worker*. Further, one might be interested in finding *contrast sets*, i.e. association rules whose frequency differs significantly across classes [15, 60]. One might also be interested in finding *classification rules*, such as $X \mapsto Y$, where the antecedent X consists of attribute value pairs and the consequent Y is a class [82]. There, the aim is to find a set of rules which allows to predict the class of any object. In contrast, we are interested in finding subsets of nodes where each subset itself is characteristic for a class.

Next we review different methods to find interesting nodes in graphs.

2.3 Methods for finding interesting nodes

Consider data represented as a graph and an interestingness measure of nodes given. Then, the most interesting nodes can be found by various methods. A review of all possible methods is out of the scope of this thesis. We focus on *incremental methods* that pick one interesting node after another, methods that *iteratively improve* the overall interestingness by changing the set of nodes, methods for *clustering* nodes and picking a *medoid as representative* for each cluster, and methods that *find diverse sets of nodes* in order to select nodes characteristic for their class.

In the following sections we describe each of these methods. Further, we briefly review areas where they have been applied.

2.3.1 Incremental methods

In each step, greedy methods make the choice that looks best at that step [35]. We are especially interested in greedy methods that incrementally pick one interesting node after another w.r.t. an interestingness measure $int(\cdot)$. Such an incremental greedy method produces a ranked list of nodes. In order to find k interesting nodes one can simply select the top k nodes of the ranked list.

Consider a word co-occurrence network is given as well as a similarity measure of nodes. Assume that the aim is to find terms (nodes) relevant w.r.t. the term *root*, but non-redundant to each other. Then, an incremental greedy method first finds the most relevant term w.r.t. the term *root*, which might be the term *tree*. The second most relevant term might be *maple*. However, the term *maple* is quite redundant w.r.t. the term *tree*. Instead, the term *equation* might be selected as it is relevant w.r.t. *root* but non-redundant w.r.t. *tree*. Next, the incremental greedy method finds a term relevant w.r.t. *root*, but non-redundant w.r.t. the terms *tree* and *equation*. That is, it might find *indo-european language* as the third most interesting term.

A greedy method is guaranteed to find a set of k objects (nodes) which achieves at least $1/k$ of the optimal score if objects are chosen w.r.t. a function that is *submodular* [102]. An interestingness measure $int(\cdot)$ is submodular if it satisfies the following diminishing returns property: the marginal gain of adding an object to a set A of objects is at least as big as adding it to any of its supersets $B \supseteq A$:

$$int(A \cup \{x\}) - int(A) \geq int(B \cup \{x\}) - int(B).$$

If the interestingness measure $int(\cdot)$ is not only submodular but also

nondecreasing (the marginal change is either positive or zero), then a greedy method is guaranteed to find a set of k objects (nodes) which achieves at least $(e-1)/e \approx 63\%$ of the optimal score [102].

Several variants of random walk have been proposed that rank nodes in an incremental, greedy fashion to capture their relevance and non-redundancy [152, 96, 132]. We have been describing how random walks can measure node similarities in Section 2.1.2, relevance by (personalised) PageRank in Section 2.2.1, non-redundancy in Section 2.2.2, and irrelevance in Section 2.2.3. Next, we describe three variants of random walk that find relevant and non-redundant nodes by incrementally picking one relevant node after another which are non-redundant to the previously selected nodes.

First, in an *absorbing random walk* nodes are selected in an incremental greedy fashion such that at each step a random walk is performed on the graph, after which the most relevant node is selected, which is then turned into an absorbing node, i.e. a node at which a random walk stops [152]. The absorbing nodes diminish the relevance of nodes with high similarity, because a random walk visiting them will be more likely absorbed than a random walk visiting more distant nodes. As a result, the diversity across the selected nodes is enhanced.

Second, in a *vertex-inforced random walk* nodes are selected in an incremental greedy fashion such that at each step, a random walk is performed on the graph, the most relevant node is selected, and the transition probabilities are reinforced as follows [96]. The transition probabilities to previously selected nodes increase, including those from a node to itself. (In this setting each node has an edge to itself.) Thus, a random walker visiting a node close to an already visited node will probably move to the already visited node and stay there for some time. As a result, a random walk will more often visit nodes distant from already visited nodes than nodes close to an already visited node. Hence, the relevance of nodes distant to already selected nodes increases and a diverse set of nodes is selected.

Finally, a variant of *random walk with restart* ranks nodes measuring their relevance and non-redundancy as follows [132]. Nodes are selected in an incremental greedy fashion such that at each step, the node is selected which adds the highest marginal contribution to the set of previously selected nodes w.r.t. the relevance and non-redundancy of Equation 2.10. This method finds a set of k nodes which achieves at least 63% of the optimal score, as the relevance and non-redundancy measure of Equation 2.10 is submodular and nondecreasing.

Further, incremental greedy methods have been used in IR to find documents that have high relevance to the query but contain minimal similarity

to previously selected documents, by selecting, at each step, the document with the highest marginal relevance and non-redundancy. Then, the greedy method finds at each iteration the most relevant node, outputs it and adds it to the set of retrieved nodes R.

Similarly nodes in a graph can be selected one by one w.r.t. the highest marginal relevance and non-redundancy of Equation 2.13 [96]. Several subsequent IR approaches incrementally find documents which are relevant w.r.t. the query, but mutually non-redundant to each other using other relevance and non-redundancy measures (see e.g. [28, 32, 3] and Section 2.2.2). Some of these relevance and non-redundancy measures are submodular (e.g. [3]).

Incremental greedy methods have also been used to rank recommendations [121]. For instance, such an incremental greedy method has been used to select, at each step, the recommendation with the highest relevance and non-redundancy (see Equations 2.14–2.16).

Two graph-based methods for word sense disambiguation also use incremental greedy methods to identify word senses [139, 2]. There, for each (target) term that should be disambiguated a word co-occurrence network is built where nodes represent terms co-occurring with the given word [139]. Then an incremental greedy method selects, at each step, the term with either the highest degree [139] or the highest PageRank [2]. Neighbours of previously selected nodes are no longer eligible to be selected. Then, for an instance of a term its sense is identified as follows. Terms surrounding it give scores to the selected nodes, from which the one with the highest sum of scores represents the sense (see [139, 2] for more details).

Further, incremental greedy methods have been used in the setting of viral marketing in order to find a set of k customers (nodes) of maximum influence [40, 69]. There, at each step a customer is selected if a marketing action offered for that customer increases the expected revenue, which can be measured by a submodular and nondecreasing function [69]. Hence, the performance is guaranteed to be at least 63% of the optimum.

Incremental greedy methods have also been used in other settings. Often, at each step the object is selected which has minimum similarity or maximum distance to previously selected objects. Such greedy methods have been used, e.g. to find a diverse set of chemical compounds [31], a diverse set of solutions [59], or a diverse set of reviews [81]. Alternatively, an incremental greedy method can at each step select the object, e.g. with the highest information gain w.r.t. previously selected objects [108].

2.3.2 Iterative improvement

In this section, we review methods that iteratively improve the overall interestingness of a set of nodes. Even though the greedy incremental method makes the best possible choice in each step, the set of top k nodes is not necessarily optimal for any k except $k = 1$.

The iterative method produces a non-redundant set of k relevant nodes, where k is given. Given an initial (e.g. random) set of k nodes, such a method iteratively replaces $l \leq k$ nodes by l new nodes from outside of the current k nodes if the interestingness of the node set improves. The swapping is then repeated until no improvements can be achieved anymore.

Consider again the word co-occurrence example. Assume the aim is to find terms which are relevant w.r.t. the term *root*, but non-redundant to each other. Given the terms *tree*, *maple*, and *equation* as the initial set of nodes, the terms *tree* and *maple* might be considered to be redundant as they represent the same context: *botanics*. Replacing the term *maple* by *indo-european language* produces a set consisting of the terms *tree*, *indo-european language*, and *equation*. These terms are relevant as each term co-occurs with the term *root*, and at the same time they are non-redundant as they represent different contexts: *botanics*, *linguistics*, and *mathematics*, respectively. Hence, the latter set might be more interesting for the user.

The method as such is deterministic (except when there are ties) and is guaranteed to stop: the number of possible configurations of k nodes is finite, and since the solution is changed only if it is improved, the method never returns to a previous solution. Unfortunately the number of possible solutions is exponential.

Incremental methods (Section 2.3.1) and iterative methods can be combined to find a set of interesting nodes. In a generalised method $l_- \leq k$ nodes are removed and $l_+ \leq k$ nodes are added at each step. Such a method with different values of l_- and l_+, where $l_- \leq l_+$, has been used, e.g. for feature selection [113]. The greedy incremental ($l_- = 0$ and $l_+ = 1$) and the iterative improvement ($l_- = l_+$) are instances of such a combined approach.

2.3.3 Clustering

The aim of graph clustering or partitioning is to assign nodes into groups (clusters) such that nodes within the same cluster are strongly connected, but there are sparse connections between clusters [104]. Alternatively, the aim can be to identify clusters of nodes such that nodes within the same cluster share some common characteristics [50].

Consider again the word co-occurrence network. There, the terms *maple*,

birch, *aspen* and *pine* might constitute one cluster, and the terms *English*, *German* and *Finnish* another cluster.

A typical class of clustering methods is **hierarchical clustering**. There, nodes are either grouped (*bottom-up* or *agglomerative* methods) or split (*top-down* or *divisive* methods) in an iterative manner [50, 118].

Agglomerative methods start with each node in a cluster of its own [104]. In each iteration, those two clusters are merged that give the best merged cluster as a result, measured by some linkage criteria. Typical linkage criteria include the highest similarity of two nodes of different clusters (*single* or *minimum linkage*), the highest similarity among those nodes farthest away from each other w.r.t. a pair of clusters (*maximum* or *complete linkage*) or the highest average similarity of nodes in two different clusters (*average* or *mean linkage*). Other linkage criteria can also be used (see e.g. [118, 151]).

Divisive methods (including *graph partitioning*) start with all nodes in one cluster [104, 26]. In each iteration, the cluster is split into (typically two) clusters [118]. Splitting criteria can be based on node betweenness [51], minimum cut [57], or low conductance [24], among others.

For both the agglomerative and divisive methods, the clustering can be finished when exactly k clusters are obtained, where k is a pre-specified number, or when a threshold of a cluster quality score is reached [118]. Alternatively one can cluster in the agglomerative case until all nodes are merged to the same cluster, or in the divisive case until each node belongs to a separate cluster. The clusters for each step can then be represented as a *dendrogram* (a hierarchical tree), where each leaf contains one node, and the root consists of all nodes [104]. Horizontal cuts through the tree at different levels represent the results for different numbers k of clusters. Cluster quality criteria can be used to determine where to cut the dendrogram [118].

The **k-medoid** clustering method [68] is similar to the better known k-means clustering method [92]. Given n objects and $k \leq n$, the number of clusters to be constructed, both methods start with k initial (e.g. random) cluster centres. Then, all objects are assigned to the cluster identified by the closest centre, after which the cluster centres are re-calculated. The object assignment and cluster centre calculation is iterated until the clusters stabilise. The two methods differ in their choice of cluster centre: whereas the k-means clustering method uses the mean value of the objects within a cluster, k-medoids uses the object with minimum average (or minimum total) dissimilarity to the objects within a cluster as cluster centre [68].

Choosing an object as cluster centre (medoid) is a practical necessity when working with graphs, since there is no well defined mean for a set of nodes. Given a graph and a node similarity measure, the medoid of a

cluster is then the node for which the average or total similarity to all other nodes in that cluster is maximal. Alternatively, the medoid node can also be chosen w.r.t. some other measure, e.g. the closeness centrality [116].

The k-medoids method also immediately gives a set the representatives. As each medoid is representative for its cluster [68], k clusters define a set of k medoid nodes. Then, these k medoid nodes can be defined as a set of representative (interesting) nodes in the graph. However, a possible problem of k-medoids is related to the use of medoids: it may discover star-shaped clusters, where cluster members are connected mainly through the medoid.

Consider again the word co-occurrence example. Given the terms *maple*, *birch*, *aspen* and *pine*, the term *birch* is representative for the set as it is a typical example among them. Identifying such representative terms gives, e.g. a quick and representative overview of the different clusters.

Various other methods exist for clustering nodes in graphs as well. For instance, *online clustering* methods process one object (node) at a time [118], *local clustering* methods compute one cluster at a time based on local information of the graph [26, 118], *spectral clustering* is based on eigenvectors [95], and *Markov clustering* is based on random walks [137].

Nodes in a graph have been clustered in various application areas [118]. Next, we briefly review such applications where nodes in a graph have been clustered based on their graph topology in order to find interesting nodes.

Finding a representative author (a node) and a representative scientific article (a node of another type) for a given subset of time stamp nodes might provide an interpretation for that subset [134]. There, subsets (clusters) of time stamp nodes are obtained by a spectral clustering method. Then, the representativeness measure of Equation 2.21 is used (as stated before in Section 2.2.4) to find representative nodes for each cluster.

Both hierarchical and k-medoids clustering have been used to cluster nodes in social networks [116]. However, medoids are only used for clustering with k-medoids, but are not considered to be representatives there.

Clustering has been used to identify different senses of an ambiguous word as follows [42]. For an ambiguous word a local co-occurrence network of similar words is built [144]. Then, terms (nodes) are clustered by an online Markov clustering and semantically close clusters are merged [42]. Each cluster is assigned a sense label by identifying the hypernym in WordNet which subsumes as many nouns as possible of that cluster [143].

Clustering nodes in a graph by k-medoids and identifying a representative node for each cluster based on degree, betweenness centrality and HITS has been used to approximate shortest path length computations, and for visualising large graphs [147]. A comparison of the different measures showed

that random and centrality-based selections of representatives performed equally well when approximating the shortest path length. However, when approximating the shortest path length to hubs, identifying the representatives based on degree and betweenness centrality performed better than random selection or identifying the representatives based on HITS.

2.3.4 Finding sets of nodes characteristic for class

In this section, we review methods that find subsets of nodes characteristic for a class, such as a given node attribute value. In particular, **subgroup discovery methods** find rules of the form $Condition \mapsto Subgroup$, where the antecedent $Condition$ is a conjunction of attribute values and the consequent $Subgroup$ is a set of objects which satisfy some class-related interestingness measure [74, 146, 83]. That is, given a graph where nodes represent objects, subgroup discovery methods find rules of the form $T' \mapsto V_{T'}$ where nodes within a subset $V_{T'} = V_1 \cap \cdots \cap V_k \subseteq V$ share the set of node attribute values $T' = \{t_1, \ldots, t_k\} \subseteq T$, the size of the subset $|V_{T'}|$ is large enough, and the set satisfies some class-related interestingness measure (Equation 2.22).

Subgroup discovery methods often find such rules in a top-down general-to-specific search [146, 94]. For instance, one can search for rules such that in each generation sets of node attribute values are expanded in all possible ways, the generated sets are evaluated for interestingness, and the interesting rules are expanded further in the next iteration [94]. This is continued until a prespecified iteration depth is achieved or no further interesting rules can be found. Subgroup discovery methods such as searching for enriched gene sets (SEGS) [135] and SDM-Aleph [138] make use of attribute value hierarchies. In general, the search can be performed breadth-first, depth-first, or best-first [146]. (For a review on subgroup discovery methods see [78].)

Some *classification methods* aim to find rules for predicting the class of any object (see Section 2.2.5), though classification methods can be adapted to serve as subgroup discovery methods [83]. For instance, CN2-SD [82] is a modified version of the CN2 [30]. Similarly, other classification methods could be adapted [82].

Next we will present the contributions of this thesis and consider the connections between the previous work just described and our work presented in this thesis.

Chapter 3

Contributions of this thesis

This thesis includes four original publications. In this chapter we will explain for each article how we define interestingness and describe the methods we used to find interesting nodes. We will motivate our choices and discuss how the definitions of interestingness and the methods used differ from those reviewed in Section 2.2 and Section 2.3, respectively. Finally, we will show some experimental results we obtained.

3.1 Finding relevant and non-redundant nodes

Article I addresses the problem of identifying relevant and non-redundant pieces of information. We assume a user specifies some positive, and possibly also some negative query objects. Then, the aim is to identify other objects that are relevant w.r.t. the positive query objects, irrelevant w.r.t. any negative query object, and non-redundant w.r.t. each other.

Here, objects can denote nodes in a graph, but they can denote any types of objects for which a distance or similarity function is defined, though in the following we will refer to those as nodes.

Next, we discuss our choice of relevance and non-redundancy measure. We then describe how an incremental greedy method as well as a method that iteratively improves the result find relevant and non-redundant nodes w.r.t. that measure. Experiments on word co-occurrence and co-authorship networks are reported. Given a word co-occurrence network, the terms selected by the incremental method can be subjectively identified as relevant and non-redundant. The results obtainted with a co-authorship network demonstrate that both methods produce a good set of relevant and non-redundant nodes when compared to random results.

3.1.1 Relevance and non-redundancy

Relevance. We define the relevance of a node $u \in V$ w.r.t. a single positive query node $q \in V$ directly as their similarity:

$$rel_P(u, q) = s(u, q) = d(u, q)^{-1}. \tag{3.1}$$

Given a set $Q_P \subset V$ of (positive) query nodes, we define the relevance of node u w.r.t. a set Q_P as

$$rel_P(u, Q_P) = (\sum_{q \in Q_P} d(u, q)^{\alpha})^{-\frac{1}{\alpha}} \tag{3.2}$$

where $\alpha \geq 1$. This is the inverse of the p-norm [97]. Since the p-norm is a distance but we want to measure relevance, we take the inverse of it. Equation 3.1 is a special case of this definition when $Q_P = \{q\}$ and $\alpha = 1$.

With $\alpha = 1$, the p-norm is the sum of the distances. Then, our relevance can be interpreted as a variant of closeness centrality (Equation 2.6), where the closeness is measured only w.r.t. the query nodes. That is, given two positive query nodes, all nodes on the shortest path between those two query nodes have an equal, highest relevance.

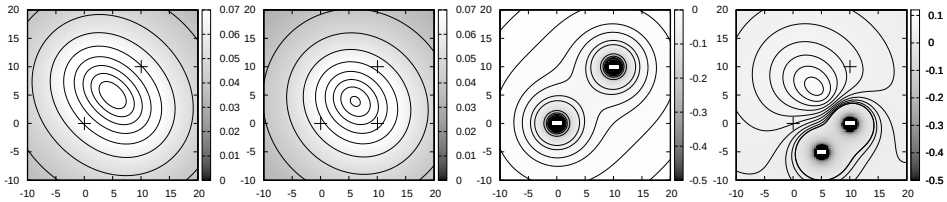
The relevance measure of Equation 3.2 meets our desire that a node is considered to be more relevant if it is close to all positive query nodes. By choosing the inverse of the p-norm distance, and not simply the closeness to positive query nodes, we give the user the possibility to choose that larger distances should dominate the function more by setting $\alpha > 1$. With $\alpha = \infty$ the p-norm is the maximum of the distances.

For the sake of illustration, consider a set V of points on a plane and the Euclidean distance $d(u, v)$ between points. Figure 3.1 (a)–(b) shows the relevance with $\alpha = 4$ and two or three positive query points. The panels illustrate how the relevance with $\alpha > 1$ emphasises larger distances and, in effect, favours points that are more equally distant to all query points.

Our definition of relevance (Equation 3.2) has some nice properties. It is monotone decreasing in the distance to each query node (with the exception of $\alpha = \infty$ when it is a function of the largest distance alone). Further, the formulation as a function of the set of distances guarantees certain simplicity as it rules out complex relevance functions that would depend on the inner structure of the set Q_P of positive query nodes.

Irrelevance. The irrelevance or negative relevance of node u w.r.t. a single negative query node $\bar{q}$ is measured with the given similarity or distance function, just like relevance to a single positive query node:

$$rel_N(u, \bar{q}) = s(u, \bar{q}) = d(u, \bar{q})^{-1}. \tag{3.3}$$



(a) $\alpha = 4$ **(b)** $\alpha = 4$ **(c)** $\beta = 2$ **(d)** $\alpha = 4, \beta = 2$

Figure 3.1: Altitude profiles of (a)–(b) relevance, (c) irrelevance, and (d) overall relevance, of points on a plane. Positive query points are denoted by pluses, negative ones by minuses. Lighter areas are more relevant.

Given a set $Q_N \subset V$ of negative query nodes, we define the negative relevance of node u w.r.t. $Q_N \subset V$ as

$$rel_N(u, Q_N) = \sum_{\bar{q} \in Q_N} d(u, \bar{q})^{-\beta} = \sum_{\bar{q} \in Q_N} s(u, \bar{q})^{\beta}, \tag{3.4}$$

where $\beta \geq 1$. That is, in the special case of $\beta = 1$ we measure irrelevance to negative query nodes in a similar way as redundancy to a set of nodes is measured in Equation 2.17, with the minor difference that we do not normalise our irrelevance. (Remember that the desirable effect of irrelevance is similar to redundancy.)

The irrelevance measure of Equation 3.4 has desirable properties, too. It is zero if there are no negative query nodes, the effect of a negative query node infinitely far away is zero, and the function is monotonically decreasing in each distance. That is, negative query nodes are treated as a disjunction: a node is considered to be less relevant if it is close to *any* negative query node. Consider again nodes representing terms. When the terms *mathematics* and *linguistics* are specified as negative query terms the terms *equation*, *formula*, *indo-european language* and *English* are quite irrelevant as each of them has a low distance to one of the query terms.

The situation is subtly different from positive query nodes, where the relevance of an node was defined to be highest when the node is relevant to *all* query nodes (as weighted by parameter α). Hence, p-norm would *not* be a good alternative here, as it would prefer nodes centred between all negative query nodes. That is, in the above example the terms *discipline* and *science* would be irrelevant as they have low distance to both, *mathematics* and *linguistics*. However, the user might still be interested in *disciplines* and *science* in general. In contrast, terms like *equation* and *formula* would be considered to be less irrelevant as they have low distance

only to one negative query node, though they are quite irrelevant for a user who specified *mathematics* as negative query term.

Setting $\beta > 1$ allows the user to give more weight to larger similarities, i.e. to more proximal negative query nodes. The higher its value is, the more dominant are the most proximal nodes.

For an illustration of the effects of negative query points, consider again a set V of points on a plane. Figure 3.1 (c) shows how the effects of negative query objects are concentrated locally around them.

Given positive and negative query nodes, i.e. sets Q_P and Q_N, respectively, the **overall relevance** of node u is defined as

$$REL(u, Q_P, Q_N) = rel_p(u, Q_P) - rel_N(u, Q_N). \tag{3.5}$$

In the special case of $\alpha = 1$ and $\beta = 1$ we balance relevance to positive and negative query nodes as relevance and redundancy have been balanced in Equation 2.13: by their difference.

As a result, our overall relevance measure favours nodes that are centred between the positive query nodes and that are not close to any negative one. Given a single positive and single negative query point, it simply measures which one is closer.

Figure 3.1 (d) illustrates the combined effect of two positive and two negative query points on a plane: the most relevant area is no longer exactly between the positive query points, but is pushed away by the negative query points.

Non-redundancy. We want to retrieve a list of relevant nodes, which are mutually non-redundant or complementary to each other. We define redundancy in a set $R \subseteq V$ of nodes in a similar way that we defined negative relevance:

$$red(R) = \sum_{\substack{u,v \in R \\ u \neq v}} d(u, v)^{-\gamma} = \sum_{\substack{u,v \in R \\ u \neq v}} s(u, v)^{\gamma}, \tag{3.6}$$

where $\gamma \geq 1$. In the special case of $\gamma = 1$ our non-redundancy measure is almost equivalent to the one of Equation 2.18 with the minor difference that we do not add similarities of nodes to themselves.

Overall relevance and non-redundancy. The overall goal is to find a diverse set of relevant nodes according to the user's query. Using the definitions above, we define the overall relevance and non-redundancy of a set of (retrieved) nodes $R \subseteq V$ as

$$REL(R, Q_P, Q_N) = \sum_{u \in R} rel_P(u, Q_P) - \sum_{u \in R} rel_N(u, Q_N) - red(R). \tag{3.7}$$

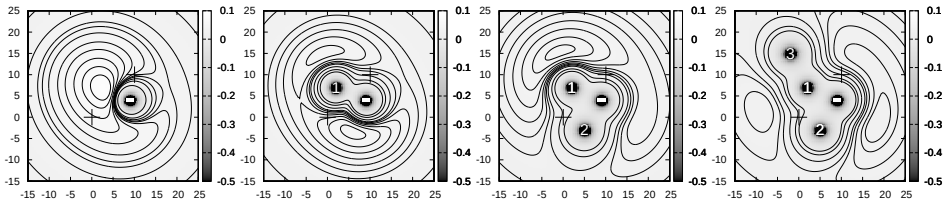


Figure 3.2: The greedy method applied on points on the plane, with $\alpha = 4$, $\beta = 2$, two positive (plusses), and one negative (minus) query point. The identified points are denoted by digits in their output order. Contour lines are displayed only for positive overall relevance values, a thick one where it is zero.

This gives a relatively simple but general objective function that tries to find a balance between relevance w.r.t. positive query nodes, avoidance of negative query nodes, and mutual non-redundancy of nodes in the result. Thereby, relevance, irrelevance and non-redundancy are treated in quite a uniform way. The measure is general in the sense that it is based on node distance or similarity, but independent of the actual choice of which distance or similarity measures are used.

Next, we will describe two approaches for finding a set of nodes that are relevant but non-redundant w.r.t. Equation 3.7.

3.1.2 Incremental method

In Article I we propose to adapt the incremental greedy methods described in Section 2.3.1 to produce a ranked list of relevant and non-redundant nodes. At each iteration such a greedy method finds the currently most relevant node w.r.t. the overall relevance $REL(u, Q_P, Q_N)$. We treat irrelevance and non-redundancy in the same way by setting $\gamma = \beta$. Then we can, at each iteration, add the most relevant node simply to the negative query nodes. As a result, the ith node output is non-redundant w.r.t. the first $i - 1$ nodes already output.

Figure 3.2 illustrates how the greedy method incrementally picks points from the plane given two positive and one negative query point.

In Article I we showed that $REL(R, Q_P, Q_N)$ of Equation 3.7 is submodular. Hence, an incremental greedy method finds a set of k nodes with an overall relevance and non-redundancy of at least $1/k$ of the optimal score.

3.1.3 Iterative improvement

In Article I we further proposed to use the iterative method described in Section 2.3.2 in order to improve an initial set of k nodes w.r.t. $REL(u, Q_P, Q_N)$.

In each step, one of the k nodes is replaced by the optimal one, given the $k-1$ other current nodes, until no improvements can be achieved.

The initial solution R clearly could have an important effect on the quality of the result as the iterative method may converge to a local optimum. We therefore propose the following alternatives to initialising it:

1. Run the greedy method first (for k iterations) and use the top k nodes from it as the initial solution to the iterative method.
2. Give k random nodes as the initial solution.

Next, we will show empirically that both methods indeed find relevant and non-redundant nodes.

3.1.4 Experiments and results

In Article I we performed experiments on a word co-occurrence and on a co-authorship network. Based on the results, both methods produce a good set of nodes, with high relevance and low redundancy, on both data sets. Next, we will show some exemplary results we obtained. (See Article I for all experiments and their detailed description.)

Word co-occurrence network. Let us illustrate with a word co-occurrence network that the generic model is able to perform a non-trivial task without being specifically tuned for it. The goal is to test how the proposed framework manages to find different senses of a given word. In Table 3.1 we present the results.

The two most relevant non-redundant words associated to *bank*, for instance, are *reserve* (which corresponds to sense #5 of bank in the WordNet[1] dictionary: "a supply or stock held in reserve for future use"), and *river* (sense #1: "sloping land [...] beside a body of water"). The third most relevant word is *gaza*, as in Gaza Strip, which occurs in the specific context of the West Bank of the Jordan river. The fourth most relevant word is *credit* (sense #2: "a credit card processing bank"). The fifth most relevant word is *international*, which does not correspond to any WordNet sense of bank (or banking), but is highly ranked, because it occurs often in the corpus in phrases like "international banking".

For *star*, we obtain several relevant words from the astronomical context, but also a name (Star Trek) and the sense of being a celebrity or movie star.

For *branch* and *root* as the positive query terms, the three first relevant words again represent different contexts: botanics (*tree*), linguistics (*indo*), and *mathematics*. The other two terms relate to *mathematics* as well.

[1] http://wordnet.princeton.edu/

bank	star	branch, root
reserve	planet	tree
river	trek	indo
gaza	cluster	mathematics
credit	sirius	line
international	movie	equation

Table 3.1: Top five words ranked as relevant and non-redundant by the greedy algorithm for $\alpha = 4$, $\beta = 2$ and different words and a word pair as positive query nodes.

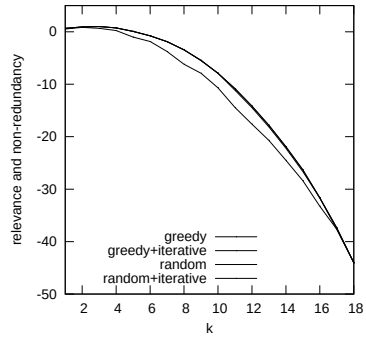


Figure 3.3: Overall relevance of set R_k of top k nodes obtained by different methods with $\alpha = 4$ and $\beta = 2$.

Comparison of the two methods. For comparing the two methods we retrieved a set of k nodes with the following four different variants:

1. finding relevant and non-redundant nodes with the greedy method and taking the top k nodes,
2. finding them initially with the greedy method and improving the results with the iterative method,
3. picking k nodes randomly initially and improving the results with the iterative method, and
4. simply picking k nodes randomly.

Figure 3.3 shows the overall relevance and non-redundancy of the four variants. It is slightly positive for small k, but eventually drops to negative values for larger k. The overall relevance becomes negative, e.g. when the mutual redundancy of the selected nodes to each other is larger than their relevance to the positive query nodes.

Comparing the four variants to each other indicates that the three first ones, using the greedy and iterative methods, are practically indistinguishable while the random results are systematically inferior. This indicates that the result of the greedy method is, in addition to being a ranking of the nodes, also a good choice for any given k. Another observation is that the iterative method performed equally well with random initialisation as it does with initial ranking obtained by the greedy method.

Hence, based on the results of Article I, both the incremental and the iterative method produce a good set of relevant and non-redundant nodes. An interesting result is that the method that produces a ranking also seems to work well in practice for any top k nodes.

3.2 Finding relevant and non-redundant nodes in probabilistic graphs

Article II extends the model of Article I (described in Section 3.1) to identify and retrieve relevant and non-redundant nodes in a probabilistic graph.

Next, we discuss how we adapted the relevance, irrelevance and non-redundancy measures of Article I to this special case in Article II. We again use an incremental greedy method as well as a method that iteratively improves the result to find such nodes. Experiments on co-authorship networks are reported, for experiments on a biological network see Article II. The results suggest that both measures, the standard and the probabilistic relevance and non-redundancy measure produce a good set of relevant and non-redundant nodes.

3.2.1 Relevance and non-redundancy

As we consider probabilistic graphs, the similarity $s(u, v)$ of two nodes u and v can be measured by a probability, such as the probability that the nodes are related or linked (Equation 2.2). To map probabilities to distances we then use the function $d(u, v) = -\log(s(u, v)) = -\log(prob(bp(u, v)))$. As a result the most probable paths can be reduced to shortest paths.

Relevance. Using Equation 3.1 and setting $\alpha = 1$, the relevance of a node $u \in V$ w.r.t. a single positive query node $q \in V$ is then

$$rel_P(u, q) = d(u, q)^{-1} = -\log(s(u, q))^{-1} \tag{3.8}$$

if $u \neq q$ and they are connected.

The relevance w.r.t. a set of positive query nodes Q_P is measured by

$$rel_P(u, Q_P) = (\sum_{q \in Q_P} d(u, q))^{-1} = -\log(\prod_{q \in Q_P} s(u, q))^{-1} \tag{3.9}$$

That is, $rel_P(u, Q_P)$ is a lower bound of the sum of network reliabilities [34] (see Section 2.1.2). Equation 3.9 is approximate also for another reason: it does not take into account possible overlaps in the best paths. The probabilities of any shared edges will be counted several times. This could be circumvented by considering the union of all edges, but we anticipate that this additional complexity is not significant in practice.

Irrelevance. For the irrelevance we propose to use the maximum inverse distance or maximum similarity

$$rel_N(u, Q_N) = \max_{\bar{q} \in Q_N} d(u, \bar{q})^{-\beta} = \max_{\bar{q} \in Q_N} s(u, \bar{q})^{\beta} \tag{3.10}$$

instead of the sum of similarities (or inverse distances). Clearly, $rel_N(\cdot)$ is a approximation of Equation 3.4 as it is a lower bound of it and it is the highest lower bound we can obtain using just one negative query node.

This simple optimisation reduces the run time. Recall that the irrelevance of a node u usually depends on its distance to all negative query nodes. However, it is mostly dependent on the nearest negative query node. Hence, a reasonably good approximation can be obtained without computing all of these. Thus, this optimisation does not change the worst case complexity, but can give a practical advantage.

Given a probabilistic graph and $\beta = 1$, Equation 3.10 translates to

$$rel_N(u, Q_N) = \max_{\bar{q} \in Q_N} ((-log(s(u, \bar{q})))^{-1}) = (-\log \max_{\bar{q} \in Q_N} s(u, \bar{q}))^{-1}. \quad (3.11)$$

Non-redundancy. Similarly to the definition of irrelevance, the non-redundancy can be defined as the maximum inverse distance or similarity:

$$red(R) = \max_{\substack{u,v \in R \\ u \neq v}} d(u, v)^{-\beta} = \max_{\substack{u,v \in R \\ u \neq v}} s(u, v)^{\beta} \quad (3.12)$$

which translates given a probabilistic graph and $\beta = 1$ to

$$red(R) = \max_{\substack{u,v \in R \\ u \neq v}} ((-log(s(u, \bar{q})))^{-1}) = (-\log \max_{\substack{u,v \in R \\ u \neq v}} s(u, v))^{-1}. \quad (3.13)$$

Overall relevance and non-redundancy. The overall relevance and non-redundancy can be defined as before (Equation 3.7) as the difference of relevance, irrelevance and non-redundancy:

$$REL(R, Q_P, Q_N) = \sum_{u \in R} rel_P(u, Q_P) - \sum_{u \in R} rel_N(u, Q_N) - red(R). \quad (3.14)$$

3.2.2 Incremental method

We again use a greedy method to produce a ranked list of nodes in an incremental fashion w.r.t. the overall relevance $REL(u, Q_P, Q_N)$. In each iteration, it finds the currently most relevant node and outputs it.

3.2.3 Iterative improvement

We also use an iterative method to produce a non-redundant set of k relevant nodes, where k is given as a parameter. Given k initial nodes as input (again, either obtained from the greedy method or randomly chosen), the method replaces, in each iteration, one of the k nodes by the optimal one, given the $k - 1$ other nodes. The method stops when no improvements can be achieved.

3.2.4 Experiments and results

We performed experiments on social and biological networks, using different distance measures. Based on the results, both methods produce a good set of nodes, with high relevance and low redundancy with a standard as well as with a probabilistic distance measure. Next, we will show some exemplary results we obtained. (See Article II for all experiments and their detailed description.)

Let us demonstrate the effect of the proposed relevance and non-redundancy measures with a co-authorship network connecting *C. Faloutsos* and *J. Han*. We used four different pairwise similarity measures:

- LEN-SP: the reciprocal of the length of the shortest path (Equation 2.1) on boolean edge weights,
- LEN-SP-RWR: Random walk with restart with transition probabilities proportional to LEN-SP,
- CUM: a similarity measure proportional to a cumulative distribution function [112] in the range $[0, 1]$, where the similarity of any two authors, especially when not co-authors, is defined using the best path between them, taking the product of pairwise similarities along the path as the final similarity, and
- CUM-RWR: Random walk with restart with transition probabilities proportional to CUM.

When using LEN-SP or CUM as similarity, and either the relevance and non-redundancy measure of Section 3.1.1 or the probabilistic relevance and non-redundance measure of Section 3.2.1, the top eight authors obtained are all prominent researchers that are relatively closely related to Faloutsos and Han by direct or indirect co-authorship relations (see Article II for their names and countries of affiliation). In both cases, the first four of the chosen authors have never published together according to DBLP, so they are likely to represent different communities or areas relevant to Faloutsos and Han. The spread of the results is also illustrated by the fact that many of the first eight authors come from different countries.

In contrast, if redundancy is ignored and the computation is based only on relevance, a redundant set of authors is obtained, regardless of which similarity measure is used (see Article II for their names and countries of affiliation). The eight most relevant authors are highly connected to each other in the co-authorships network, and come from either the US or Canada, with a few exceptions. A redundant set of authors is also obtained when LEN-SP-RWR or CUM-RWR are used as similarity measure.

Hence, the results of Article II show that both the standard and probabilistic relevance and non-redundancy measures based on different similarity measures produce a good set of relevant and non-redundant nodes.

3.3 Finding representative nodes in probabilistic graphs

Article III examines how to find a small representative subset of nodes out of a larger set of nodes.

Next, we discuss our choice of representativeness measure. We then describe how we use clustering methods to find clusters of nodes and select a representative node for each cluster. The results suggest that clustering-based approaches are capable of finding a representative set of nodes.

3.3.1 Representativeness

We assume a probabilistic graph is given and use Equation 2.2 to measure pairwise node similarities.

Then, given a set of nodes, we find a node that represents the nodes within the set by adapting the representativeness measure of Equation 2.19:

$$rep(u, R) = \prod_{v \in R} s(u, v) = \prod_{v \in R} prob(bp(u, v)) \tag{3.15}$$

in case all nodes within R are connected. That is, the node that has a maximal product of similarities between each other node in the cluster and itself is chosen as the new medoid.

3.3.2 Clustering

In Article III we used k-medoids as well as hierarchical clustering as described in Section 2.3.3 to find sets (clusters) of nodes.

For the k-medoids clustering we randomly chose k nodes as initial medoids. Then, the k-medoids method iteratively assigns the other nodes to the closest medoids based on the similarity measure of Equation 2.2. If the pairwise similarity between a node and all medoids equals zero, the node will be considered an outlier and is not assigned to any medoid in this iteration. Next, the medoids are re-calculated for each cluster R by finding the node $u \in R$ that maximises Equation 3.15. This is repeated until the clustering converges or the maximum number of iterations is reached. Hence, k-medoids directly produces representatives based on the representativeness measure (Equation 3.15), but not necessarily the optimal ones.

In addition, we use agglomerative hierarchical clustering to find k clusters. Starting with each node as a separate cluster, at each step it merges the two clusters that are the closest based on average linkage and the similarity measure of Equation 2.2. We used average linkage to give more weight on cluster coherence. The hierarchical clustering stops when k clusters were obtained. For each cluster we selected one representative based on the representativeness measure of Equation 3.15.

3.3.3 Experiments and results

In Article III we performed experiments on biological networks. Based on the results, both clusterings produce a good set of representatives. Next, we will show some exemplary results we obtained. (See Article III for all experiments and their detailed description.)

Example. Let us illustrate with the following example that k-medoids finds a representative set of $k = 3$ genes out of a set of nine genes. The nine genes belong to three known groups, each group associated with a phenotype. The k-medoids clustering converged in this case after two iterations. Figure 3.4 presents the result. Only one gene (EntrezGene:1627) was assigned to another cluster than it should w.r.t. the phenotypes. Apart from this, the clustering produced the expected partitioning: each gene was assigned to a cluster close to its corresponding phenotype. The three representatives (medoids) are genes assigned to different phenotypes. Hence, the medoids can be considered representative for the nine genes.

Biological networks. Let us next look at experiments performed on 100 biological networks. For each network there were $k = 10$ sets of genes given, where each set consisted of 3 to 41 genes. (See Article III for further experiments with $k = 3$.) Genes within a set related to the same gene family (or disease). We used k-medoids as well as hierarchical clustering to cluster the genes and selected a representative for each cluster obtained. As k-medoids is sensitive to the randomly selected first medoids, we applied k-medoids five times in each run and selected the best result.

For comparison, we also considered a method that selects representatives randomly. We randomly select k medoids and clustered the remaining nodes of S to the most similar medoid. If the pairwise similarity between a node and all medoids equals zero, the node will be considered an outlier, as in k-medoids. We applied the random selection of representatives 20 times in each run and used average values of the measures in order to compensate the random variation.

We evaluated how successful the methods are in finding representative nodes based on four measures. First, we simply measure the *average simi-*

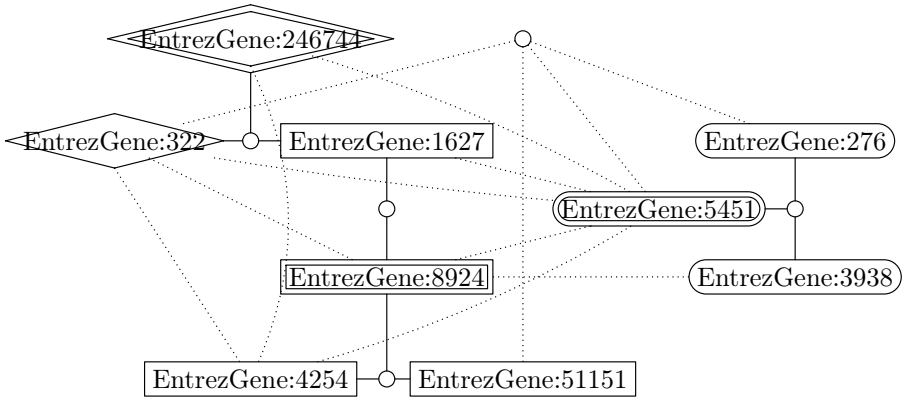


Figure 3.4: Clusters (diamonds, boxes, ellipses) and representatives (double borders) of nine given nodes, and some connecting nodes (circles) on best paths between them. Lines represent edges between two nodes, dotted lines represent best paths with several nodes.

larity of nodes to their closest representative (ASR):

$$ASR = \frac{1}{|V'| - k} \sum_{\substack{u \in V' \\ u \neq m(u)}} s(u, m(u)) \tag{3.16}$$

where V' is the set of given nodes to be clustered, k is the number of clusters, $m(u)$ is the medoid most similar to u, and $s(x, m(u))$ denotes the similarity (probability of best path) between node u and medoid $m(u)$.

In terms of average similarity of nodes to their representative, the k-medoids method slightly outperforms the hierarchical method (Figure 3.5, left panel). The hierarchical method, in turn, is clearly superior to the random selection of representatives (Figure 3.5, right panel).

Second, we take advantage of our knowledge that the genes belong to gene families. Specifically, we calculate the *fraction of non-represented classes* (NRC):

$$NRC = \frac{1}{k} |\{i \mid \nexists j : m_j \in H_i,\ j = 1..k\}|, \tag{3.17}$$

where k is the number of classes and clusters (equal in our current test setting), m_j is the medoid of the jth cluster, and H_i is the ith original class.

Table 3.2: Fraction of non-represented classes (NRC).

k-medoids	29 %
hierarchical	21 %
random	39 %

Table 3.3: Wrongly assigned objects, i.e. nodes (WAO).

k-medoids	44 %
hierarchical	25 %
random	46 %

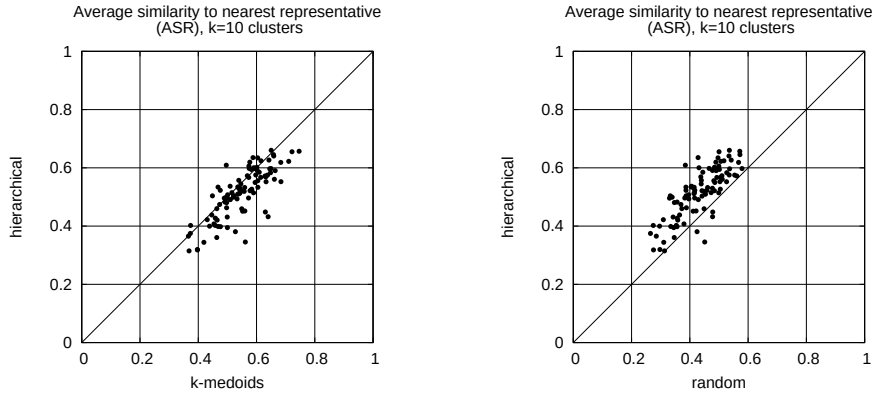


Figure 3.5: Average similarity of nodes to their nearest representative (ASR). In each panel 100 runs are visualised. Each point represents one run, thereby comparing ASR values of two variants (see x- and y-axis).

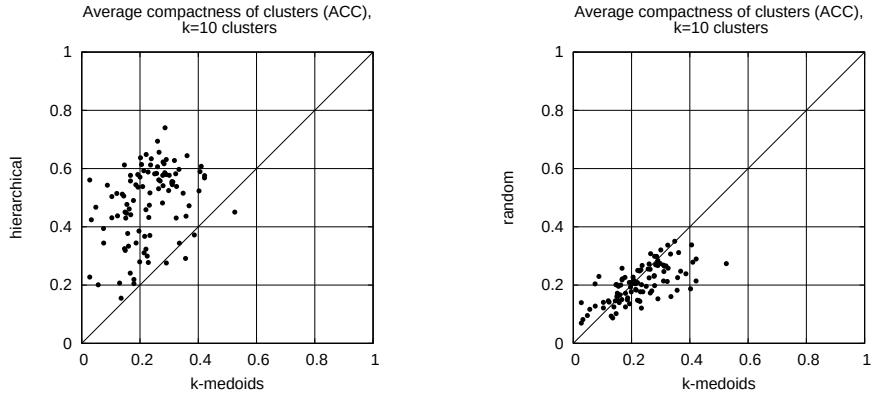


Figure 3.6: Average compactness of nontrivial clusters (ACC). In each panel 100 runs are visualised. Each point represents one run, thereby comparing ACC values of two variants (see x- and y-axis).

The fraction of non-represented classes is a more neutral measure of performance since neither variant directly maximises this. The results indicate that the hierarchical variant is clearly superior to the k-medoids variant (Table 3.2). Both methods clearly outperform the random selection of representatives.

Third, we measured how good the underlying clusterings are based on the *compactness of clusters* (ACC). Specifically, we calculated the minimum similarity of two nodes in each cluster and averaged it across clusters with at least two members:

$$ACC = \frac{1}{k'} \sum_{k=1}^{K} \min_{x,y \in C_k} s(x, y), \tag{3.18}$$

where $k' = |\{k : |C_k| > 1, k = 1..K\}|$, i.e. k' is the number of non-trivial clusters. This measure is sensitive to outliers, and thus may favour the k-medoids variant.

It is not surprising that clusters produced by the hierarchical method are more compact on average than those produced by the k-medoids method (Figure 3.6), as the hierarchical method more directly optimises this measure. It is however somewhat surprising that k-medoids performs only slightly better than the random variant.

Fourth, we measured how good the underlying clusterings are by measuring the difference of the clustering to the known classes. That is, we first identify the class best represented by each cluster, and then calculate the fraction of *"wrongly assigned objects (nodes)"* (WAO):

$$WAO = \frac{1}{|S|} \sum_{k=1}^{K} \min_{k'=1..K} |C_k \backslash H_{k'}|. \tag{3.19}$$

Rand index could have been used here just as well.

In terms of wrongly assigned nodes, the hierarchical variant clearly outperforms k-medoids (Table 3.3). The k-medoids variant outperforms the random selection of representatives, but only by a small difference.

Hence, based on the results of Article III both the k-medoids-based variant and the hierarchical-based method reliably identify a high quality set of representatives, though the hierarchical method seem to be more robust. Further, the success of the methods in identifying the underlying clusters depends on the evaluation measure used, and may also depend on the number of clusters to be constructed.

3.4 Finding sets of nodes characteristic for contrast classes

Article IV addresses the problem to identify sets of nodes that are worth to explore by the user, when the user is interested in the comparison of two or more classes, such as given node attribute values, at the same time. This is a typical problem in bioinformatics, where the aim is to find enriched gene sets that are specific for virus-infected samples at a specific time point.

Next we describe the interestingness measure we proposed in Article IV which allows us to find such subsets of nodes. We then discuss how to find such node sets by extending subgroup discovery by a second subgroup discovery step. Notice that here we take node attributes and class attributes into account, but not the graph topology. The results can then be visualised in a graph and used for other graph mining methods [111]. Experimental results on a time-series data set of virus-infected *S. tuberosum* (potato) plants revealed subsets of genes that were unexpected and useful for a plant biologist.

3.4.1 Characteristic for contrast classes

In Article IV we propose to replace the direct dependency on the class distribution of the classical subgroup discovery by a contrasting, indirect one. In the classical, direct case, one is interested in sets of node attribute values that are characteristic for a class. Our aim is to understand phenomena in a setting where several different classes are given, such as different time points. That is, in the contrasting case, we want to find sets of node attribute values that indicate nodes which are characteristic for their class, but not necessarily the same one.

In order to formally define the task, we first introduce a notation P for the set of nodes characteristic for a class:

$$P = \{v \in V \mid \exists\ T' \subset T \text{ for which } f_c(T') \leq \alpha \text{ and } v \in V_{T'}\}, \qquad (3.20)$$

where (as before) T denotes the set of node attribute values, V the set of nodes, $V_{T'}$ the set of nodes that share the attribute values $T' \subset T$, $f_c(\cdot)$ the class-related interestingness measure, and α a given constant. That is, set P consists of nodes which belong to subsets which are characteristic for a specific class.

Now the user can define two contrast classes $P_c, \overline{P_c} \subset P$. The selection of these two contrast classes depends on the objective and is left to the user. They can, e.g., take several classes into account. Given m different class

attributes $c_1, \ldots, c_m$ such as different time points, one can, e.g., contrast one specific time point c_k against all other time points.

Let $P_1, \ldots, P_m$ be the sets of nodes characteristic for each of the m class attributes. Then, P_c can be defined, e.g., as the set of nodes *only* characteristic for the kth given class attribute:

$$P_c = P_k \setminus \bigcup_{\substack{i \in \{1,\ldots,m\}, \\ i \neq k}} P_i \ . \tag{3.21}$$

This definition can be used to find interesting subsets which are specific for one class attribute in contrast to all the other class attributes. Alternatively, it could be defined, e.g., as their intersection (see Article IV) if one wants to find interesting subsets that are common to all class attributes.

The contrast class $\overline{P_c}$ can be defined as the complement of P_c, i.e.

$$\overline{P_c} = P \setminus P_c \ , \tag{3.22}$$

when one is interested in subsets specific for the nodes in P_c compared to all other nodes of P. Or, if a user is interested in contrasting two specific time points even in a case were more time points exist, then P_c would be defined as one of those time points and $\overline{P_c}$ as the other time point.

Let us then define the function $characteristic(\cdot)$ that gives the number of nodes characteristic for their class in the two contrasting classes P_c and $\overline{P_c}$ for a given subset $V_{T'}$:

$$\begin{aligned} characteristic: \ & \mathcal{P}(V) \to \mathbb{Z}_+ \times \mathbb{Z}_+, \\ & V_{T'} \mapsto (|V_{T'} \cap P_c|, |V_{T'} \cap \overline{P_c}|). \end{aligned} \tag{3.23}$$

Then the *contrasting interestingness* of a set of node attribute values can be defined as

$$\begin{aligned} f_i: \ & \mathcal{P}(T) \to \mathbb{R}, \\ & T' \mapsto g'(characteristic(V_{T'}), characteristic(P \setminus V_{T'})) \end{aligned} \tag{3.24}$$

for some function g' that measures the class distributions within and outside of the subset of nodes (see Section 2.2.5).

That is, the contrasting interestingness measure analyses whether a subset is interesting w.r.t. the two contrast classes, which both consist only of nodes that are characteristic for their own class. This is in contrast to the classical class-related interestingness measure, which analyses whether a subset of nodes is interesting w.r.t. the node's classes.

Then the *contrasting subgroup discovery problem* is to output all sets $T' \subset T$ of attribute values for which $f_i(T') \leq \alpha'$ for some given constant α' or, to identify the k best subsets of nodes instead of using a fixed threshold.

In other words, while classical subgroup discovery is related to the question of how to find subsets of nodes that are characteristic for a specific class, the problem of contrasting subgroup discovery is related to asking if subsets of nodes characteristic for (any) class can be found.

The relationship between the classical and contrasting subgroup discovery immediately implies that for any subset found for the contrasting subgroup discovery problem, its nodes are characteristic for their class. On the other hand, a set of attribute values may be a valid answer to the contrasting problem even if it is not for the classical problem.

That is exactly where the main conceptual contribution of Article IV is. Contrast subgroup discovery allows finding subsets of nodes that could not be found with classical subgroup discovery.

Next, we describe how to find subsets of nodes characteristic for contrast classes.

3.4.2 Contrasting subgroup discovery

Given a set of nodes described by node attribute values and different classes that a user has specified she is interested in, our goal is to find interesting subsets of nodes characteristic for their class. Thereby we allow the user to specify not only one, but several different classes as interesting.

To find such subsets we propose an approach that consists of three steps: First, interesting subsets are found by a classical subgroup discovery method. Second, contrast classes on those subsets are defined by set theoretic functions. Third, contrasting subgroup discovery finds interesting subsets for the contrast classes. Next, we will describe each step in detail.

Classical subgroup discovery (Step 1). A subgroup discovery method is applied on some given nodes that are annotated by node attributes and assigned a class. Thereby, we consider only one class attribute the user is interested in at a time, and apply a subgroup discovery method separately for each class attribute. The subgroups are then analysed by a statistical test, like Fisher's exact test followed by a permutation test.

Construction of contrast classes (Step 2). Let $P_1, \ldots, P_m$ denote the nodes characteristic for their class of the m class attributes specified by the user (e.g. each representing one time point in a gene expression experiment). Then, the two contrast classes P_c and $\overline{P_c}$ are defined by set theoretic functions, e.g. by Equations 3.21 and 3.22. (As stated before, the selection of a particular set theoretic function depends on the objective and is left to the user.)

Contrasting subgroup discovery (Step 3). In this step we apply a second subgroup discovery instance in order to analyse subsets w.r.t. the

constructed contrast classes. Given the subsets in the two contrast classes P_c and $\overline{P_c}$, we find interesting subsets of these nodes by a second subgroup discovery instance. Again, p-values are calculated, using a permutation test.

Assuming that both subgroup discovery instances (Step 1 and 3) find all subsets for which the classical interesting measures hold (Equation 2.22), then the proposed method does find all subsets that satisfy the indirect interestingness measure (Equation 3.24).

Similar to the approach presented here, some frequent item set mining methods intersect transactions to find closed frequent item sets [98, 107, 17]. Further, supervised pattern mining approaches take a class labelled data set as input (emerging pattern mining, exception rule mining and contrast set mining), but they can take multiple classes into account by comparing two classes where one is a union of several (sub)classes [86]. Especially, in contrast set mining two contrast classes are defined, and in a setting where several different class attributes exist, these methods can be applied in a pairwise manner. Similarly, we also aim to understand the differences between several contrasting groups. However, in contrast to these approaches we aim to find different types of patterns (described in Section 2.2.5). Further, our aim is to find interesting subsets of nodes which are characteristic for their class, regardless of their class.

Next, we will show that our proposed method indeed finds subsets of nodes interesting for the user.

3.4.3 Experiments and results

We performed experiments on a *Solanum tuberosum* (potato) time-labelled gene expression data set for virus-infected and non-infected plants. (See Article IV for a detailed description and all experimental results.)

Our interest is in assisting biologists to generate new research hypotheses. Therefore, a plant biologist evaluated our results by counting the quantities of gene sets which are unexpected as well as those which are useful to him (as in [126]). In this context, *unexpected* means that the knowledge was contained in the biological ontologies GO[2] (Gene Ontology) [71], KO[3] (Kyoto Encyclopedia of Genes and Genomes (KEGG) Orthology) [9] or GoMapMan[4], an extension of the MapMan [128] ontology, for plants, but it was not shown previously to be related to *S. tuberosum*'s response to viral infection. A gene set is *useful* if it is of interest for the plant biologist, i.e. the gene set description tells him something about the virus response,

[2] `http://www.geneontology.org/`

[3] `http://www.genome.jp/kegg/ko.html`

[4] `http://www.gomapman.org/`

and/or he might want to have a closer look at the genes of that gene set. We compare the results obtained by our proposed method (Step 1 to 3) to those results obtained with a classical subgroup discovery method (Step 1 only). The proposed contrasting subgroup discovery method identified several gene sets in Step 3, which were unexpected and useful for the biologist which were not identified in Step 1, i.e. the classical subgroup discovery method. (See Article IV for quantities of all, unexpected and useful gene sets found with each method.)

Gene sets that are unexpected, useful or both may contain genes that are interesting for further (tough, time-consuming) wet-lab experiments. Let us give a few examples of such identified gene sets. A gene set that the plant biologist considered as unexpected was

`unidimensional cell growth (GO:0009826)`

which covers 7 genes with a p-value of 0.0001. This gene set was novel when compared to the classical subgroup discovery method (Step 1).

An example of a useful gene set found by the contrasting subgroup discovery method (Step 3) is

`enoyl-CoA hydratase activity (GO:0004300)`

which covers 7 genes with a p-value $\leq 10^{-6}$. This gene set was novel when compared to the classical subgroup discovery method (Step 1).

An exemplary gene set found by the contrasting subgroup discovery method (Step 3) that was considered as unexpected as well as useful is

`ER to Golgi vesicle-mediated transport (GO:0006888)`
$\wedge$ `vesicle coat (GO:0030120)`

which covers 14 genes with a p-value of 0.0001. This gene set was more specific compared to the classical subgroup discovery method (Step 1).

Hence, the results of Article IV show that our proposed method finds subsets of nodes that are characteristic for a time point, a class specified by a biologist. These subsets can be unexpected and useful for a biologist and thus direct him where to look for the causes of the differences between the time points. Other methods (and possibly data) are needed to find those causes. For example, studying the genes (nodes) of such subsets may reveal new research hypotheses for biologists.

Next we will answer the research questions and discuss open questions and future work.

Chapter 4

Conclusions

In this thesis we addressed the problem of discovering interesting nodes in weighted graphs where nodes represent objects and edges relations between them. We reviewed how interestingness of nodes within a graph has been defined previously, and which methods have been used to find such interesting nodes. We then discussed the main contributions of Articles I-IV.

4.1 Answers to the research questions

Each article of this thesis gives slightly different answers to the research questions: *What kind of nodes are interesting for the user?* and *How to find such interesting nodes in weighted graphs?* Which nodes are considered to be interesting or worth to explore by the user, differs within Articles I-IV. Also the methods to find such nodes differ across the articles.

Specifically, we proposed four different measures for node interestingness. In Article I we proposed that nodes are interesting if they are relevant and non-redundant, given positive and negative query nodes. We based our definitions of relevance, irrelevance and non-redundancy only on node distance or similarity. In Article II we adapted them for probabilistic graphs. In Article III a node is interesting if it is representative, i.e. if it is a typical example of a set of nodes. We based our definition of representativeness on probabilistic node similarity. In Article IV we proposed to consider nodes to be interesting if they are characteristic for a class, even if the classes are not identical.

We then gave methods for identifying such nodes: In Article I we used a greedy method that incrementally picks one interesting node after another and a method that improves the set of nodes when the size of the set is fixed. Article II shows that the same methods can be used to find relevant and

non-redundant nodes in a probabilistic graph. In Article III representative nodes are found by clustering the given nodes and picking a medoid node for each cluster. In Article IV nodes characteristic for a class are found by contrasting diverse sets of nodes and by combining well-known subgroup discovery methods. The method is not specific to graphs, but finding nodes characteristic for a class is one possible instance of the presented problem.

While the measures and methods are relatively simple, the experiments on word co-occurrence networks (Article I), co-authorship networks (Article I–II), and biological networks (Article II–IV) show that they indeed identify interesting nodes. We believe that these new measures and methods allow experts and practitioners from various fields to identify a small number of interesting nodes within a large set of nodes.

4.2 Outlook

This work is preliminary in several aspects and it would be interesting to address at least the following aspects in future work.

For instance, it would be interesting to apply and compare other, possibly new interestingness measures. For theoretical guarantees, it would be nice to have a nondecreasing, nonnegative relevance and non-redundancy function. Interestingness measures based on more expressive node similarities, e.g. on network reliability, could prove more powerful, but are also computationally more demanding. On the other hand, for complex and large applications faster methods would be useful. For example, the incremental and iterative methods are simple but efficient if the similarity measure is readily available. More efficient definitions or approximations of node similarity are needed for better scalability to large graphs.

It would also be interesting to investigate whether graph topology-based measures for finding nodes characteristic for a given node attribute prove more powerful. On the other hand, one could also examine whether relevance and non-redundancy measures or representativeness measures that take node attributes into account improve the results.

Further, one could seek to understand the reasons for the differences of various approaches, such as the incremental versus the iterative method and the k-medoids- versus the hierarchical-based method, or whether other methods such as a combined approach of the incremental and iterative approaches prove more powerful.

Finally, it would be interesting to validate the performance of the presented measures and methods on further, real applications and in domains not considered so far. In particular, it would be interesting if users could

evaluate the results in depth, e.g. a biologist could evaluate the interesting gene sets at the gene level, including a selection of genes for wet-lab experiments, investigating whether the identified subsets will affect the understanding of the biological mechanisms of virus response.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] E. Agirre, D. Martínez, O. L. de Lacalle, and A. Soroa. Two graph-based algorithms for state-of-the-art WSD. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, pages 585–593, Sydney, Australia, 22–23 July 2006.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining (WSDM '09)*, pages 5–14, Barcelona, Spain, 9–12 February 2009.

[4] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD '93)*, pages 207–216, Washington, D.C., USA, 26–28 May 1993.

[5] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, Menlo Park, CA, USA, 1996.

[6] L. Akoglu, D. Chau, J. Vreeken, N. Tatti, H. Tong, and C. Faloutsos. Mining connection pathways for marked nodes in large graphs. In *Proceedings of the 13th SIAM International Conference on Data Mining (SDM '13)*, pages 37–45, Austin, TX, USA, 2–4 May 2013.

[7] L. Akoglu, M. McGlohon, and C. Faloutsos. OddBall: Spotting anomalies in weighted graphs. In *Proceedings of the 14th Pacific-Asia Conference (PAKDD '10)*, pages 410–421, Hyderabad, India, 21–24 June 2010.

[8] A. Anagnostopoulos, G. Brova, and E. Terzi. Peer and authority pressure in information-propagation models. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD '11)*, pages 76–91, Athens, Greece, 5–9 September 2011.

[9] K. F. Aoki-Kinoshita and M. Kanehisa. Gene annotation and pathway mapping in KEGG. In J.M. Walker and Nicholas H. Bergman, editors, *Comparative Genomics*, volume 396, pages 71–91. Humana Press, New York City, NY, USA, 2007.

[10] M. Atzmüller, F. Puppe, and H.-P. Buscher. Exploiting background knowledge for knowledge-intensive subgroup discovery. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pages 647–652, Edinburgh, Scotland, 30 July – 5 August 2005.

[11] B. Bakker and T. Heskes. Clustering ensembles of neural network models. *Neural Networks*, 16(2):261–269, 2003.

[12] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[13] A.-L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3–4):590–614, 2002.

[14] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 101(11):3747–3752, 2004.

[15] S. D. Bay and M. J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5:213–246, 2001.

[16] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[17] C. Borgelt, X. Yang, R. Nogales-Cadenas, P. Carmona-Saez, and A. Pascual-Montano. Finding closed frequent item sets by intersecting transactions. In *Proceedings of 14th International Conference on Extending Database Technology (EEDBT/ICDT '11)*, pages 367–376, Uppsala, Sweden, 21–25 March 2011.

[18] M. Böttcher. Contrast and change mining. *Data Mining and Knowledge Discovery*, 1(3):215–230, 2011.

[19] B. Boyce. Beyond topicality: A two stage view of relevance and the retrieval process. *Information Processing & Management*, 18(3):105–109, 1982.

[20] U. Brandes. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[21] U. Brandes and T. Erlebach. *Network Analysis: Methodological Foundations*. Springer-Verlag, Secaucus, NJ, USA, 2005.

[22] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[23] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pages 335–336, Melbourne, Australia, 24–28 August 1998.

[24] J. J. Carrasco, D. C. Fain, K. J. Lang, and L. Zhukov. Clustering of bipartite advertiser-keyword graph. In *Proceedings of the Workshop on Clustering Large Data Sets, In conjunction with the 3rd IEEE International Conference on Data Mining (ICDM '03)*, Melbourne, FL, USA, 19–22 December 2003.

[25] D. Chakrabarti. AutoPart: Parameter-free graph partitioning and outlier detection. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD '04)*, pages 112–124, Pisa, Italy, 20–24 September 2004.

[26] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys*, 38(1), 2006.

[27] H. Chang, D. Cohn, and A. McCallum. Learning to create customized authority lists. In *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pages 127–134, Stanford, CA, USA, 29 June – 2 July 2000.

[28] H. Chen and D. R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pages 429–436, Seattle, WA, USA, 6–11 August 2006.

[29] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data (SIGMOD '03)*, pages 551–562, San Diego, CA, USA, 9–12 June 2003.

[30] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.

[31] R. D. Clark. OptiSim: An extended dissimilarity selection method for finding diverse representative subsets. *Journal of Chemical Information and Computer Sciences*, 37(6):1181–1188, 1997.

[32] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, pages 659–666, Singapore, Singapore, 20–24 July 2008.

[33] L. Cohen, L. Manion, and K. Morrison. *Research Methods in Education.* Taylor and Francis, Oxford, UK, 2003.

[34] C. J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, Oxford, UK, 1987.

[35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, Cambridge, MA, USA, and McGraw-Hill Book Company, New York City, NY, USA, 1990.

[36] E. Costenbader and T. W. Valente. The stability of centrality measures when networks are sampled. *Social Networks*, 25(4):283–307, 2003.

[37] N. N. Dalvi and Suciu D. Efficient query evaluation on probabilistic databases. *VLDB Journal*, 16(4):523–544, 2007.

[38] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. In *Proceedings of the 4th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining (KDD '98), pages 30–36, New York City, NY, USA, 27–31 August 1998.

[39] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. PageRank, HITS and a unified framework for link analysis. In *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM '03)*, pages 249–253, San Francisco, CA, USA, 1–3 May 2003.

[40] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pages 57–66, San Francisco, CA, USA, 26–29 August 2001.

[41] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '99)*, pages 43–52, San Diego, CA, USA, 15–18 August 1999.

[42] B. Dorow and D. Widdows. Discovering corpus-specific word senses. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics (EACL '03)*, pages 79–82, Budapest, Hungary, 12–17 April 2003.

[43] L. Eronen and H. Toivonen. Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC Bioinformatics*, 13(1):119, 2012.

[44] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71:056103, 2005.

[45] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 118–127, Seattle, WA, USA, 22–25 August 2004.

[46] D. A. Fell and A. Wagner. The small world of metabolism. *Nature Biotechnology*, 18:1121–1122, 2000.

[47] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[48] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1978–1979.

[49] D. Gamberger and N. Lavrač. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17(1):501–527, 2002.

[50] L. Getoor and C. P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.

[51] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[52] W. Goffman. A searching procedure for information retrieval. *Information Storage and Retrieval*, 2(2):73–78, 1964.

[53] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[54] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[55] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pages 359–366, Stanford, CA, USA, 29 June – 2 July, 2000.

[56] T. E. Harris and F. S. Ross. *Fundamentals of a Method for Evaluating Rail Net Capacities*. The RAND Corporation, Santa Monica, California, 1955. Research Memorandum RM-1573.

[57] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4–6):175–181, 2000.

[58] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*, pages 517–526, Honolulu, HI, USA, 9–16 November 2002.

[59] E. Hebrard, B. Hnich, B. O'Sullivan, and T. Walsh. Finding diverse and similar solutions in constraint programming. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI '05)*, pages 372–377, Pittsburgh, PA, USA, 9–13 July 2005.

[60] R. J. Hilderman and T. Peckham. Statistical methodologies for mining potentially interesting contrast sets. In F. Guillet and H.J. Hamilton, editors, *Quality Measures in Data Mining*, pages 153–177. Springer-Verlag, Berlin/Heidelberg, Germany, 2007.

[61] P. Hintsanen and H. Toivonen. Finding reliable subgraphs from large probabilistic graphs. *Data Mining and Knowledge Discovery*, 17(1):3–23, 2008.

[62] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[63] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[64] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, Cambridge, UK, 2010.

[65] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pages 271–279, Budapest, Hungary, 20–24 May 2003.

[66] H. Jeong, S. P. Mason, A.-L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

[67] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

[68] L. Kaufmann and P. Rousseeuw. Clustering by means of medoids. In Y. Dodge, editor, *Statistical Data Analysis based on the L1 Norm*. Elsevier, Amsterdam, Netherlands, 1987.

[69] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 137–146, Washington, DC, USA, 24–27 August 2003.

[70] R. W. Kennard and L. A. Stone. Computer aided design of experiments. *Technometrics*, 11(1):137–148, 1969.

[71] P. Khatri and S. Drăghici. Ontological analysis of gene expression data: Current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595, 2005.

[72] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Proceedings of the 2011 SIAM International Conference on Data Mining (SDM '11)*, pages 47–58, Mesa, AZ, USA, 28–30 April 2011.

[73] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[74] W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 249–271. MIT Press, Cambridge, MA, USA, 1996.

[75] W. Klösgen, M. May, and J. Petch. Mining census data for spatial effects on mortality. *Intelligent Data Analysis*, 7(6):521–540, 2003.

[76] Y. Koren, S. C. North, and C. Volinsky. Measuring and extracting proximity in networks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pages 245–255, Philadelphia, PA, USA, 20–23 August 2006.

[77] T. Kötter and M. R. Berthold. (Missing) concept discovery in heterogeneous information networks. In *Proceedings of the 2nd International Conference on Computational Creativity (ICCC '11)*, pages 135–140, Mexico City, Mexico, 27–29 April 2011.

[78] P. Kralj Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403, 2009.

[79] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Core algorithms in the CLEVER system. *ACM Transactions on Internet Technology*, 6(2):131–152, 2006.

[80] C. K. Kwoh and P. Y. Ng. Network analysis approach for biology. *Cellular and Molecular Life Sciences*, 64:1739–1751, 2007.

[81] T. Lappas, M. Crovella, and E. Terzi. Selecting a characteristic set of reviews. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pages 832–840, Beijing, China, 12–16 August 2012.

[82] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[83] N. Lavrač, F. Železný, and P. A. Flach. RSD: Relational subgroup discovery through first-order feature construction. In *Proceedings of the 12th International Conference (ILP '02)*, pages 149–165, Sydney, Australia, 9–11 July 2003.

[84] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pages 631–636, Philadelphia, PA, USA, 20–23 August 2006.

[85] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*, pages 641–650, Raleigh, NC, USA, 26–30 April 2010.

[86] J. Li, G. Liu, and L. Wong. Mining statistically important equivalence classes and delta-discriminative emerging patterns. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pages 430–439, San Jose, CA, USA, 12–15 August 2007.

[87] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[88] P.-M. Lin, B. J. Leon, and T. C. Huang. A new algorithm for symbolic system reliability analysis. *IEEE Transactions on Reliability*, R-25(1):2–15, 1976.

[89] S. Lin and H. Chalupsky. Unsupervised link discovery in multi-relational data via rarity analysis. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pages 171–178, Melbourne, FL, USA, 19–22 December 2003.

[90] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[91] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7(2):72–93, 2005.

[92] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium*

on Mathematical Statistics and Probability, pages 281–297, Berkeley, CA, USA, 21 June – 18 July 1965 and 27 December 1965 – 7 January 1966. Published 1967.

[93] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[94] M. May and L. Ragia. Spatial subgroup discovery applied to the analysis of vegetation data. In *Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management (PAKM '02)*, pages 49–61, Vienna, Austria, 2–3 December 2002.

[95] F. McSherry. *Spectral methods for data analysis*. PhD thesis, University of Washington, Seattle, WA, USA, 2004. AAI3118856.

[96] Q. Mei, J. Guo, and D. R. Radev. DivRank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pages 1009–1018, Washington, D.C., USA, 25–28 July 2010.

[97] C. D. Meyer. *Matrix Ananlysis and Applied Linear Algebra*. SIAM, Philadelphia, PA, USA, 2000.

[98] T. Mielikäinen. Intersecting data to closed sets with constraints. In *Proceedings of the Workshop on Frequent Itemset Mining Implementations (FIMI '03)*, Melbourne, FL, USA, 19 November 2003.

[99] M. Mitra and B. B. Chaudhuri. Information retrieval from documents: A survey. *Information Retrieval*, 2(2–3):141–163, 2000.

[100] R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, 2009.

[101] S. Navlakha and C. Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.

[102] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions — I. *Mathematical Programming*, 14:265–294, 1978.

[103] M. E. J. Newman. Scientific collaboration networks. II. shortest paths, weighted networks, and centrality. *Physical Review E*, 64:016132, 2001.

[104] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B — Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[105] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 631–636, Washington, D.C., USA, 24–27 August 2003.

[106] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1999.

[107] F. Pan, G. Cong, A.K.H. Tung, J. Yang, and M.J. Zaki. Carpenter: Finding closed patterns in long biological datasets. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 637–642, Washington, D.C., USA, 24–27 August 2003.

[108] F. Pan, W. Wang, A. K. H. Tung, and J. Yang. Finding representative set from massive data. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM '05)*, pages 338–345, Houston, TX, USA, 27–30 November 2005.

[109] P. Papapetrou, T. Chistiakova, J. Hollmén, V. Kalogeraki, and D. Gunopulos. Finding representative objects using link analysis ranking. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '12)*, pages 6:1–6:4, Heraklion, Greece, 6–8 June 2012.

[110] V. Podpečan. *Knowledge Discovery in a Service-Oriented Data Mining Environment*. PhD thesis, Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia, 2013. XII.

[111] V. Podpečan, N. Lavrač, I. Mozetič, P. Kralj Novak, I. Trajkovski, L. Langohr, K. Kulovesi, H. Toivonen, M. Petek, H. Motaln, and K. Gruden. SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics*, 12(1):416, 2011.

[112] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. k-nearest neighbors in uncertain graphs. In *Proceedings of the 36th International Conference on Very Large Data Bases (VLDB '10)*, Singapore, Singapore, 13–17 September 2010.

[113] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[114] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pages 691–692, Seattle, WA, USA, 6–11 August 2006.

[115] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, pages 784–791, Helsinki, Finland, 5–9 June 2008.

[116] M. J. Rattigan, M. Maier, and D. Jensen. Graph clustering with network structure indices. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pages 783–790, Corvalis, Oregon, 20–24 June 2007.

[117] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31:581–603, 1966.

[118] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 2007.

[119] G. Schneider, P. Schneider, and S. Renner. Scaffold-hopping: How far can you jump? *QSAR & Combinatorial Science, Special Issue on Challenges in Virtual Screening*, 25(12):1162–1171, 2006.

[120] P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen. Link discovery in graphs derived from biological databases. In *Proceedings of the 3rd International Workshop on Data Integration in the Life Sciences (DILS '06)*, pages 35–49, Hinxton, UK, 20–22 July 2006.

[121] B. Smyth and P. McClave. Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR '01)*, pages 347–361, Vancouver, BC, Canada, 30 July – 2 August 2001.

[122] B. Srivastava, T. A. Nguyen, A. Gerevini, S. Kambhampati, M. B. Do, and I. Serina. Domain independent approaches for finding diverse plans. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI '07)*, pages 2016–2022, Hyderabad, India, 6–12 January 2007.

[123] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM '05)*, pages 418–425, Houston, TX, USA, 27–30 November 2005.

[124] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. RankClus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '09)*, pages 565–576, Saint Petersburg, Russia, 23–26 March 2009.

[125] E. Suzuki. Autonomous discovery of reliable exception rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD '97)*, pages 259–262, Newport Beach, CA, USA, 14–17 August 1997.

[126] E. Suzuki and S. Tsumoto. Evaluating hypothesis-driven exception-rule discovery with medical data sets. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications (PADKK '00)*, pages 208–211, Kyoto, Japan, 18–20 April 2000.

[127] K. Thiel and M. R. Berthold. Node similarities from spreading activation. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM '2010)*, pages 1085–1090, Sydney, Australia, 14–17 December 2010.

[128] O. Thimm, O. Bläsing, Y. Gibon, A. Nagel, S. Meyer, P. Krüger, J. Selbig, L.A. Müller, S.Y. Rhee, and M. Stitt. MapMan: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *The Plant Journal*, 37(6):914–939, 2004.

[129] J. Toivanen, H. Toivonen, A. Valitutti, and O. Gross. Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity (ICCC '12)*, pages 175–179, Dublin, Ireland, 30 May – 1 June 2012.

[130] H. Toivonen, S. Mahler, and F. Zhou. A framework for path-oriented network simplification. In *Proceedings of the 9th International Symposium on Intelligent Data Analysis (IDA '10)*, pages 220–231, Tucson, AZ, USA, 19–21 May 2010.

[131] H. Tong, C. Faloutsos, and J.-Y. Pan. Random walk with restart: Fast solutions and applications. *Knowledge and Information Systems: An International Journal (KAIS)*, 14:327–346, 2008.

[132] H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pages 1028–1036, San Diego, CA, USA, 21–24 August 2011.

[133] H. Tong, H. Qu, and H. Jamjoom. Measuring proximity on graphs with side information. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pages 598–607, Pisa, Italy, 15–19 December 2008.

[134] H. Tong, Y. Sakurai, T. Eliassi-Rad, and C. Faloutsos. Fast mining of complex time-stamped events. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pages 759–768, Napa Valley, CA, USA, 26–30 October 2008.

[135] I. Trajkovski, N. Lavrač, and J. Tolar. SEGS: Search for enriched gene sets in microarray data. *Journal of Biomedical Informatics*, 41(4):588–601, 2008.

[136] G. Tré, S. Zadrożny, T. Matthé, J. Kacprzyk, and A. Bronselaer. Dealing with positive and negative query criteria in fuzzy database querying. In *Proceedings of the 8th International Conference on Flexible Query Answering Systems (FQAS '09)*, pages 593–604, Roskilde, Denmark, 26–28 October 2009.

[137] S. M. van Dongen. *Graph clustering by flow simulation.* PhD thesis, Utrecht University, Utrecht, The Netherlands, 2000.

[138] A. Vavpetič and N. Lavrač. Semantic subgroup discovery systems and workflows in the SDM-toolkit. *The Computer Journal*, 56(3):304–320, 2013.

[139] J. Véronis. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252, 2004.

[140] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[141] D. R. White and S. P. Borgatti. Betweenness centrality measures for directed graphs. *Social Networks*, 16(4):335–346, 1994.

[142] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 266–275, Washington, D.C., USA, 24–27 August 2003.

[143] D. Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03)*, pages 197–204, Edmonton, AB, Canada, 27 May – 1 June 2003.

[144] D. Widdows and B. Dorow. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING '02)*, pages 1–7, Taipei, Taiwan, 24 August – 1 September 2002.

[145] R. J. Williams and N. D. Martinez. Simple rules yield complex food webs. *Nature*, 404:180–183, 2000.

[146] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery from Databases (PKDD '97)*, pages 78–87, Trondheim, Norway, 24–27 June 1997.

[147] A. Y. Wu, M. Garland, and J. Han. Mining scale-free networks using geodesic clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 719–724, Seattle, WA, USA, 22–25 August 2004.

[148] S. Wuchty and E. Almaas. Peeling the yeast protein network. *Proteomics*, 5(2):444–449, 2005.

[149] D. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pages 907–916, Paris, France, 28 June – 1 July 2009.

[150] Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, pages 1224–1231, Helsinki, Finland, 5–9 July 2008.

[151] W. Zhang, X. Wang, D. Zhao, and X. Tang. Graph degree linkage: Agglomerative clustering on a directed graph. In *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, pages 428–441, Florence, Italy, 7–13 October 2012.

[152] X. Zhu, A. B. Goldberg, J. V. Gael, and D. Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT '07)*, Rochester, NY, USA, 22–27 April 2007.

[153] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*, pages 22–32, Chiba, Japan, 20–14 May 2005.