GRAPH MINING ALGORITHMS FOR THE ANALYSIS OF PATENT CITATION NETWORKS

by

ANDREW DAVID RODRÍGUEZ

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Myong K. Jeong

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2015

ABSTRACT OF THE DISSERTATION

Graph Mining Algorithms for the Analysis of Patent Citation Networks

By ANDREW DAVID RODRÍGUEZ

Dissertation Director:

Myong K. Jeong

Patent and patent citation networks are rich datasets. In this dissertation we develop graph mining algorithms for the analysis of patent citation networks. First we develop a measure of patent influence within a patent citation network. Identifying influential or important patents helps in decision making, including focusing investment. We propose algorithms based on the powerful graph kernels for the ranking of patents in influence, and we demonstrate how the von Neumann graph kernel is well suited for influence analysis in patent citation networks.

Secondly, we present new similarity measures between patents in a patent citation network. In the past, techniques such as text mining and keyword analysis have been applied for patent similarity calculation. The drawback of these approaches is that they depend on word choice and writing styles of authors. In this work we develop new similarity measures for patents in a patent citation network using only the patent citation network structure. The proposed similarity measures use multi-stage co-citation and bibliographic coupling links. Applications of the similarity measures include outlier scoring of patents in patent citation networks.

Finally, we propose new methods for scoring and ranking patents in outlierness within a patent citation dataset. A distinguishing characteristic of patent datasets is

that they contain both attribute data describing patents, as well as graph structure data in the citation network. Traditional outlier ranking techniques usually focus on either homogeneous vector data or on graph structure data. In this work we propose new outlier ranking methods developed specifically for patents in an attributed patent citation network. One challenge is how outlier ranking should handle these two different data types in an integrated fashion. To address this challenge, we first develop a new patent subspace clustering algorithm that considers both types of data. Based on the patent clustering result, we then develop methods for the scoring and ranking of patents in outlierness within patent citation networks. Proposed outlier score functions consider both patent attribute data and graph structure data. We compare the performance of our developed approaches with existing approaches using synthetic data and real-life U.S. patent data.

Acknowledgements

I acknowledge and thank my advisor, Dr. Myong K. Jeong, for much patience, guidance, and encouragement through the research and writing processes. I appreciate his great contributions of time and ideas. His encouragement to achieve a deep understanding of a concept, and develop it further made this accomplishment possible.

With much gratitude I acknowledge my dissertation committee members, Professors Susan Albin, Kang Li, and Jie Gong for their support, time, and efforts. Additionally, Dr. Susan Albin has been a kind and supportive advisor since the beginning of my graduate study, when she served as the Graduate Student Advisor. I thank my fellow graduate students for their engaging discussions and feedback in this research.

I would like to thank the faculty and staff of the Department of Industrial and Systems Engineering for all of their support during my time at Rutgers. Additionally, I am grateful to the faculty and staff at The University of Texas at San Antonio.

Saint John Bosco Catholic Church and The Catholic Center at Rutgers have been a sources of great encouragement, friendship, and life for many, including me. I thank those who so faithfully serve and lead the Roman Catholic Community at Rutgers University: The Brotherhood of Hope, The Sisters of Jesus our Hope, The Oratory of Saint Philip Neri, Saint Peter the Apostle Parish, and The Diocese of Metuchen.

I thank my loving Family for their care and encouragement. Our Mother and Father raised us with such love and with a belief that almost anything is possible through hard work. My older Brother has taught me so much and been so generous. My younger Sister is a light and a joy. I love, and in a very special way, I thank my strong, beautiful, loving, encouraging, and patient Wife, whose faithful support is so apparent and appreciated. I love and thank the Holy Family of Nazareth: Jesus, Mary, and Joseph. Above all, I give thanks and praise to my loving God: Father, Son, and Holy Spirit.

Dedication

To my loving Family – The Rodriguez and The Obaya Families,

to my beautiful Wife and Children, who fill my life with great joy.

Table of Contents

Abstract ii

Acknowledgements iv

Dedication v

List of Tables x

List of Figures xiii

1. Introduction 1

1.1. Overview 1

1.2. Dissertation outline 3

2. Graph Kernel-Based Centrality Measure for Evaluating the Influence of Patents in a Patent Citation Network 5

2.1. Introduction 5

2.2. Background 8

2.2.1. Graph kernels 8

Exponential diffusion kernel 9

von Neumann kernel 9

Laplacian diffusion kernel 9

2.2.2. Matrix norms 10

Entry-wise norms 10

Schatten norm 10

2.3. Graph kernel-based SVC 11

2.3.1. Introduction of GKB-SVC 11

2.3.2. Properties of GKB-SVC for patent citation networks 14

2.4. Experimental results 16
2.4.1. Data description: Example patent citation networks 16
Artificial dataset 1: 10-node patent citation network 16
Artificial dataset 2: Six 10-node patent citation networks 17
Artificial dataset 3: 15-node patent citation network 19
2.4.2. Parameter selection for GKB-SVC 20
2.4.3. Exponential diffusion graph kernel 21
Artificial dataset 1 21
Artificial dataset 2 22
Artificial dataset 3 22
2.4.4. von Neumann graph kernel 22
Artificial dataset 1 22
Artificial dataset 2 22
Artificial dataset 3 24
2.4.5. Guideline for the selection of a graph kernel and its parameter . 24
2.4.6. Comparison of out-degree centrality, original SVC, and GKB-SVC 26
Artificial dataset 1 27
Artificial dataset 2 27
2.5. Case study 28
2.6. Conclusion 30

3. Multi-stage Similarity Measure for Calculation of Pairwise Patent Similarity in a Patent Citation Network 35
3.1. Introduction 35
3.2. Background 39
3.2.1. Existing methods for patent similarity analysis 39
3.2.2. Classification codes for U.S. patents 42
3.3. Proposed multi-stage similarity measures 44
3.3.1. Multi-stage co-citation similarity measure 44

3.3.2. Multi-stage bibliographic coupling similarity measure 48
3.3.3. Normalized multi-stage co-citation and bibliographic coupling similarity measures 51
3.4. Experimental results 53
3.4.1. Data description 53
3.4.2. Parameter optimization for multi-stage co-citation 54
3.4.3. Parameter optimization for multi-stage bibliographic coupling . . 54
3.4.4. Validation of similarity scores 56
3.5. Conclusion 60

4. Patent Clustering and Outlier Ranking Methodologies for Attributed Patent Citation Networks 63
4.1. Introduction 63
4.2. Background 67
4.2.1. Graph structure-based node outlier ranking methods 68
4.2.2. Characteristics of patent citation networks 71
4.2.3. Graph structure and node attribute-based node outlier ranking methods 72
4.2.4. Subspace clustering for outlier detection 73
4.3. New subspace clustering algorithm for patents in a patent citation network 83
4.3.1. Attribute similarity criterion 84
4.3.2. Graph connectivity criterion 84
4.3.3. Subspace clustering numerical example 85
4.4. New node outlier ranking methods for attributed graphs 91
4.4.1. Integrated graph structure-based and node attribute model . . . 91
4.4.2. Weighted subspace clustering 92
4.4.3. Graph structure-based methods 95
4.5. Experimental results 96
4.5.1. Data description: Example patent citation networks 97

Artificial dataset 1: 6-node attributed patent citation network . 97
Artificial dataset 2: 14-node attributed patent citation network . 99
Artificial dataset 3: 14-node patent citation network 100
Real-life patent citation network 100
4.5.2. Artificial dataset 1: 6-node attributed patent citation network . 101
4.5.3. Artificial dataset 2: 14-node attributed patent citation network . 103
4.5.4. Artificial dataset 3: 14-node patent citation network 105
4.5.5. Real-life patent citation network 107
4.6. Conclusion and future work 108

5. Concluding Remarks and Future Research 111
5.1. Concluding remarks 111
5.2. Future research 112

Appendix A. Proof of Proposition 1 114

Appendix B. Proof of Proposition 2 115

References 116

List of Tables

2.1. Expected node influence rank for example citation Graph 1 18

2.2. Expected Node 1 influence rank for six example citation networks from dataset 2 19

2.3. Node rankings of Graph 1 using original SVC approach 20

2.4. Artificial dataset 1: effect of α parameter and p norm on ranking nodes of Graph 1 when using exponential diffusion GKB-SVC 21

2.5. Artificial dataset 2: effect of α parameter and p norm on ranking Node 1s of dataset 2 when using exponential diffusion GKB-SVC 22

2.6. Artificial dataset 3: effect of α parameter and p norms on ranking nodes of dataset 3 when using exponential diffusion GKB-SVC 23

2.7. Artificial dataset 1: effect of α parameter and p norm on ranking nodes of Graph 1 when using von Neumann GKB-SVC 23

2.8. Artificial dataset 2: effect of α parameter and p norm on ranking Node 1s of dataset 2 when using von Neumann GKB-SVC 24

2.9. Artificial dataset 3: effect of α parameter and p norm on ranking node of dataset 3 when using von Neumann GKB-SVC 25

2.10. Comparison of centrality measures for ranking 10 nodes from dataset 1 . 27

2.11. Comparison of centrality measures for ranking Node 1s from dataset 2 . 28

2.12. Comparison of ranking of U.S. patents using different centrality measures 'US-' prefix omitted) 29

2.13. Coefficient of variation for centrality scores of top 20 ranked nodes . . . 30

2.14. Comparison of ranking of U.S. patents using different centrality measures ('US-' prefix omitted) 32

2.15. Coefficient of variation for centrality scores of top 50 ranked nodes . . . 33

3.1. A sample of current U.S. patent classes 42
3.2. Number of subclasses within U.S. class codes that are associated with the selected patents 43
3.3. Guidelines for parameters of proposed co-citation and bibliographic coupling similarity measures 53
3.4. Pair-wise patent similarity scores using proposed normalized multi-stage co-citation similarity measure (CC score) and existing Jaccard similarity index for U.S. class codes of patents (US- prefix omitted in patent number) 55
3.5. Detailed information on the US-6240185, US-6389402 patent pair 56
3.6. Spearman correlation performance of proposed co-citation similarity methods when compared to Jaccard similarity using U.S. Class codes for 100 U.S. patents 60
3.7. Improvement factor of Spearman correlation over baseline: performance of proposed co-citation similarity methods when compared to Jaccard similarity using U.S. class codes for 100 U.S. patents 61
3.8. Spearman correlation performance of proposed bibliographic coupling similarity methods when compared to Jaccard similarity using U.S. Class codes for 100 U.S. patents 61
4.1. Sample attributes for U.S. patent data 66
4.2. Sample U.S. Class codes for U.S. patent data 82
4.3. Example patent attributes: number of subclasses within U.S. class codes for the five U.S. patents from Table 4.2 82
4.4. Node attributes example 83
4.5. Subspace clustering algorithm parameters 88
4.6. Resulting subspace clusters using existing and proposed algorithms, with attribute set, S, specified 90
4.7. Attribute values for the 14 nodes of the artificial dataset 2, with attributes a_1 and a_2 describing each node 100

4.8. Comparison of outlier scoring and ranking for artificial dataset 1 using proposed cluster-based method 102
4.9. Comparison of outlier ranking for artificial dataset 1 using existing and proposed methods (rank 1st is the greatest outlier) 103
4.10. Given subspace clustering result for artificial dataset 2 103
4.11. Weights for the 11 nodes that are in cluster C_1 104
4.12. Comparison of outlier score and ranking based only on subspace clustering attribute term for 14 nodes of artificial dataset 2 105
4.13. Comparison of outlier ranking based only on graph structure term for 14 nodes of artificial dataset 3 106
4.14. Top five ranked outlier patents from the real-life PCN dataset 108

List of Figures

2.1. Node j cites node i, and corresponding adjacency matrix, $\mathbf{A}$ 8

2.2. Process for ranking nodes of a citation network using proposed GKB-SVC 13

2.3. Artificial dataset 1 - example patent citation network called Graph 1 . . 16

2.4. Artificial dataset 2 - six example patent citation networks with labels . . 18

2.5. Artificial dataset 3 - example citation graph with five additional nodes citing Node 7 20

2.6. Patent citation network of the top 20 ranked important nodes using proposed von Neumann GKB-SVC with α=0.85 ('US-' prefix omitted) . 31

3.1. Example patent citation networks highlighting citation of x by i, indicating *influence* of patent x (left); and co-citation of x and y by i, indicating *relatedness* (or *similarity*) of patents x and y (right) 36

3.2. Example patent citation network highlighting bibliographic coupling and co-citation 37

3.3. Node j cites node i, and corresponding adjacency matrix, $\mathbf{A}$ 39

3.4. Level-1 co-citations in example patent citation networks to show $C_0(x, y)$ and $C_1(x, y)$ co-citation for node pair (x, y) in two cases: case 1: $i = j$ (left) and case 2: $i \neq j$ (right) 45

3.5. Four possible level-2 co-citations for node pair (l, m) 46

3.6. Example of simple bibliographic coupling of nodes x and y at level 0, indicating relatedness of patents x and y 49

3.7. Flowchart showing co-citation similarity calculation with normalization 51

3.8. Parameter evaluation for multi-stage co-citation similarity, normalized and not normalized, using weighting coefficient α^{r+1}, where r is the stage. Spearman rank correlation performance when compared to Jaccard similarity between patent pairs using U.S. Class codes for 100 patents shows best performance when applying our approaches: normalized multi-stage co-citation. 58

3.9. Parameter evaluation for multi-stage bibliographic coupling similarity, normalized and not normalized, using weighting coefficient α^{r+1}, where r is the stage. Spearman rank correlation performance when compared to Jaccard similarity between patent pairs using U.S. Class codes for 100 patents shows best performance when greater weight is given to the higher levels of bibliographic coupling. 59

4.1. Representation of 4,241 patents in a real-life patent citation network . . 64

4.2. Sample SCAN result showing two clusters, one hub, and two outliers . . 69

4.3. Attributed graph representation of patent citation network – combined graph structure and node attribute data 71

4.4. Example of the ability of subspace clustering to identify node clusters based on various attribute combinations and graph connectivity over the network 75

4.5. Patent citation network with combination of attribute and graph data, demonstrating one potential attribute subspace cluster, considering the attribute subspace *(Country, Class)* 76

4.6. Example attributed PCN to demonstrate subspace clustering algorithm 87

4.7. Weight function for different γ values 93

4.8. Example of four clusterings for different attribute subspaces, and point cluster distance for object o and Cluster C_i 94

4.9. Proposed patent outlier ranking flowchart 96

4.10. Artificial dataset 1 - example attributed PCN for node outlier ranking based on subspace clustering, where attributes indicate patent class counts 98

4.11. Artificial dataset 2 - example attributed graph for outlier ranking based on weighted subspace clustering; two attributes describe each node . . . 98
4.12. Artificial dataset 2 - given clustering of nodes using two attributes to describe nodes . . . 99
4.13. Artificial dataset 3 - example PCN graph data . . . 101
4.14. Artificial dataset 1 - example attributed patent citation network outlier ranking results using existing (left) and proposed (right) methods, where shaded node indicates outlier . . . 102
4.15. Artificial dataset 3 - example patent citation network and graph-based outlier ranking results for existing (left) and proposed (right) methods, where darker shade indicates greater outlierness . . . 107
4.16. Real-life patent citation network outlier ranking results using score function that combines cluster-based and graph-based methods, where red nodes indicate patent outliers . . . 109

Chapter 1

Introduction

1.1 Overview

Patent and patent citation data are rich datasets. Technological developments are captured in the creation of patents, thus patent analysis is considered a critical tool for company strategy formulation and planning [53, 54]. In addition, the growth of patent bibliometrics requires the development of new methodologies, different from traditional bibliometric methods [50]. Some of the patent citation network analysis methodologies that have been developed include ones for measuring inventive progress [80], identifying influential patents [40], measuring patent similarity [12, 81], identifying patent outliers [84], the visualization of technological progress [42].

In this dissertation we present new algorithms for the analysis of patent citation networks. This work is motivated by the goal of mining valuable information from complex networks. Through patent citation network analysis, core technologies may be identified, research and development investment can be focused, technological trends over time are followed, and new technology opportunities are identified.

First we propose a measure of patent influence in a patent citation network leveraging graph kernels. Identifying important patents helps in decision making, including focusing investment. In the past, centrality measures such as degree centrality and betweenness centrality have been applied to identify influential or important patents in patent citation networks [32]. How such a complex notion like technological influence can be analyzed is an important research topic. However, no existing centrality measure leverages the powerful graph kernels for this purpose. Graph kernels are well suited for this purpose since some consider the direct and indirect citations that a patent receives by considering powers of the citation network adjacency matrix. We consider the

change in the matrix norm from the inclusion to the exclusion of a patent in the patent citation network. The proposed approach provides a more robust understanding of the identification of influential nodes, since it focuses on graph structure information by considering both direct and indirect patent citations. Studies have shown that highly cited patents are of greater technical importance than less frequently cited patents, in the opinion of knowledgeable peer researchers and inventors [2]. This study leverages the premise that the change of similarity matrix that results from removing a given node indicates the importance of the node within its network, since each node makes a contribution to the similarity matrix among nodes. The node resulting in the largest change (*i.e.*, decrease) in the similarity matrix norm is considered to be the most influential node. We compare the performance of our proposed approach with other widely-used centrality measures using artificial data and real-life U.S. patent data. Experimental results show that our proposed approach performs better than existing methods.

We next present new similarity measures between patents in a patent citation network. In the past, techniques such as text mining and keyword analysis have been applied for the evaluation of patent similarity or dissimilarity [84]. The drawback of these approaches is that they depend on word choice and writing style of authors. Most existing graph-based approaches use common neighbor-based measures, which only consider direct adjacency. In this work we propose new similarity measures for patents in a patent citation network using only the patent citation network structure. The proposed similarity measures use co-citation and bibliographic coupling links. A challenge is when some patents are involved in a disproportionately large number of citations, thus are considered more similar to many other patents in the patent citation network. To overcome this challenge, we propose a normalization technique to account for the case where some pairs are ranked very similar to each other because they both are cited by, or cite, many other patents. The nature of patents in patent citation networks means that classification codes describing technology are available for each patent. We validate our proposed similarity measures using U.S. class codes for U.S. patents and the well-known Jacquard similarity index. Experiments show that the proposed methods perform well when compared to the Jaccard similarity index.

Finally, we present methods for scoring and ranking outliers in patent citation datasets. Unlike some datasets, patent data is meticulously assembled. Additionally, patent data contains both attribute data as well as graph structure data. Patent attributes such as classification codes are specifically assigned based on the nature of the technology. Similarly, graph structure data in the form of citations make and received are carefully considered by patent writers. Traditional outlier ranking techniques focus on either homogeneous vector data or on graph structures [30]. However, many of todays complex applications contain both types of data: multi-dimensional numeric information and relations between objects in attributed graphs. An open challenge is how outlier ranking should handle these different data types in a unified or integrated fashion. There is currently no work on patent citation network outlier or anomaly detection that considers attributes in patent citation network, in addition to the citation network structure. In this work we propose new outlier ranking methods developed specifically for patents in an attributed patent citation network. An open challenge is how outlier ranking should handle these different data types in an integrated fashion. To address this challenge, we first develop a new patent subspace clustering algorithm. Based on the patent clustering algorithm we then propose methods for the scoring and ranking of outlier patents within patent citation networks. Outlier score functions consider both patent attribute data and graph structure data. We compare the performance of our developed approaches with existing approaches using synthetic data and real-life U.S. patent data.

1.2 Dissertation outline

The chapters of this dissertation build upon each other and form a body of patent citation network analysis research. The rest of the dissertation is organized as follows. Following the introduction to patent citation network research in Chapter1, Chapter 2 presents a graph kernel-based approach for identifying important or influential patents within a patent citation network. Chapter 3 extends the patent citation analysis to patent co-citation and bibliographic coupling analysis, and presents new pairwise

patent similarity measures based on the citation structure of the patent citation network. Chapter 4 builds on the graph-based co-citation similarity measure, and proposes techniques for the scoring and ranking of outlier patents and patent citations in patent citation networks. Additionally, in Chapter 4, a new subspace clustering algorithm is presented for the clustering of patents in attributed patent citation networks. In each of Chapters 2 through 4, numerical examples and experimental results are presented. Finally, Chapter 5 summarizes the research results and presents future research opportunities.

Chapter 2

Graph Kernel-Based Centrality Measure for Evaluating the Influence of Patents in a Patent Citation Network

2.1 Introduction

In the creation of a new patent, it is typical for the new patent to refer to one or more previous patents in a bibliography. These citations highlight information that may be useful to the reader, explain how the current work relates to prior work, and indicates influences on the current work [21, 51, 59]. Patent citation data has long been known to be a source of information on technology innovation. Studies have shown that highly cited patents are of greater technical importance than less frequently cited patents, in the opinion of knowledgeable peer researchers and inventors [2]. Understanding technological evolution is vital for business and drives growth, and an increasing number of decision makers use patent citation analysis as a tool to survey and understand the activities of their competitors [38, 39, 74].

Citing a patent implies that the contents of the cited patent are relevant to those of the citing patent in some way. Extending this idea then, patent citation networks explain the relationship among some set of patents that cite, and are cited by, each other, where patents are the nodes of the network and an edge exists between the two nodes if one patent cites the other. Citation networks have the distinguishing characteristic of being *acyclic*, meaning that there are no closed loops of directed edges in the network [59]. This characteristic results from all directed edges (citations) going from an older patent to a newer patent, and never in the other direction. This type of network is different from networks such as the World Wide Web (WWW) and social networks, in which cycles in the networks are common.

In the general citation network, there are two kinds of nodes of particular interest:

authorities and *hubs*. Authorities are nodes that contain especially useful information on a topic of interest. Hubs, such as review papers in the scholarly article domain, are nodes that tell where the best authorities can be found [43, 59]. Kleinberg proposed a hyperlink structural analysis algorithm to determine authorizes and hubs in the World Wide Web [43]. Discovering *authoritative sources* in the WWW is similar to finding these *important nodes* in a patent citation network. Based only on the structural analysis of the patent citation network, we aim score and rank nodes in importance in order to identify the most important patents.

A great deal of research has been conducted with the goal of detecting influential or important nodes in a variety of networks [8, 9, 22, 24, 57, 63, 70]. For this objective, a variety of importance measures, called *centrality measures*, have been developed. Degree centrality is defined as the number of edges incident to a node [59]. Degree centrality can be made more specific to consider *out-degree* and *in-degree* centrality by counting the number of directed edges that are directed out from a node or are directed into a node, respectively. Another centrality measure is called *closeness*. In connected graphs there is a natural distance metric between all pairs of nodes, given by the length of the node-pair's shortest path. The *farness* of a node i is defined as the sum of its shortest path distances to all other nodes, and its *closeness* is defined as the inverse of its farness. The *betweenness* score for a node i is equal to the number of shortest paths (over all node pairs) that pass through node i. The random walk closeness centrality [62] measures the speed with which a randomly walking message reaches a node from elsewhere in the network, thus resulting in a random-walk version of closeness centrality. Kwon *et al.* [45] propose the weighted reachability (WR) measure, which is applied specifically to directed citation networks. The main idea of the WR measure is to consider both adjacent nodes (direct citations) and non-adjacent nodes (indirect citations). In this measure, direct citations are given a greater weight than indirect citations, where indirect citations are weighted inversely proportional to the length of the path between two nodes. Most of existing centrality measures do not consider the non-adjacent nodes (indirect citations). Although some centrality measures, such as WR, consider direct and indirect citations, its weighting system is not robust and offers

an opportunity for improvement.

This chapter proposes a new centrality measure focused on directed patent citation networks with unweighted edges between pairs of nodes. We do so in two new ways: (1) applying various graph kernels, which have not yet been applied for patent citation network analysis (2) leveraging the direction of citations. As a whole, we call this proposed approach graph kernel-based singular values-based centrality measure, or GKB-SVC. We are able to quantify the importance of a node using the patent citation information, specifically the patent citation network structure. The idea is to weight the adjacency matrix and the higher orders (*i.e.*, powers) of the adjacency matrix so that we capture direct and indirect citations with a varying and flexible weighting scheme, providing a centrality measure that is more robust than any existing measure. This chapter works on the assumption that the change of the similarity matrix that results from the removing of a particular node reflects the importance of that node to the network to which it belongs. We assume this relationship since each node contributes to the similarity matrix of the network, when it is included in the network. Combined with the *matrix norms*, which are a measure the size of a matrix, based on singular values, the proposed measure computes the change of similarity matrix that results from removing a node. The largest change in the matrix norm identifies the most influential node in the network. Generally, the larger the change in the matrix norm, the more influential the patent is in the patent citation network.

Our proposed centrality measure considers both paths of adjacent nodes and the nodes that are reachable, but not adjacent, as opposed to many other centrality measures that only consider nodes that are adjacent. Furthermore, our procedure allows for robust scoring and ranking of nodes in importance in order to identify influential nodes of a directed citation network so that the key technology areas are clearly identified. To evaluate the quality of the ranking produced by the proposed centrality measure applied to a patent citation network networks, we compare our results to out-degree centrality, WR [45], and original singular values-based centrality (SVC) [40] using artificial patent citation data and real-life patent citation data. Ultimately, we show that our proposed approach provides an improvement to the original SVC.

The remainder of the chapter is organized as follows. First, Section 4.2 presents some background on patent citation network research and gives an overview on various graph kernels and matrix norms. Section 2.3 presents the details of the proposed centrality measure, which leverages graph kernels. Section 2.4 presents the computational results obtained using artificial patent citation networks. Section 2.5 provides a case study on a real-life patent citation network dataset. Finally, Section 4.6 presents the conclusions reached as a result of the experimental results.

2.2 Background

In this section we provide background on patent citation networks, centrality measures, graph kernels, and matrix norms.

2.2.1 Graph kernels

We begin by describing the adjacency matrix. The adjacency matrix, denoted $\mathbf{A}$, is a matrix representation of which nodes are incident to which other nodes in a network, such as a patent citation network. For a given graph $G := (N, E)$ with $|N|$ nodes and $|E|$ edges, let $\mathbf{A} = [a_{ij}]$ be the adjacency matrix. If node j cites node i (*i.e.*, there is a directed edge from node i to node j), then $a_{ij} = 1$, otherwise $a_{ij} = 0$. That is, as $a_{ij} = 1$ if there is an edge between node i and j, $a_{ij} = 0$ otherwise. Note that matrix element a_{ij} may also be represented A_{ij}. A simple example of node j cites node i is shown in Figure 2.1.

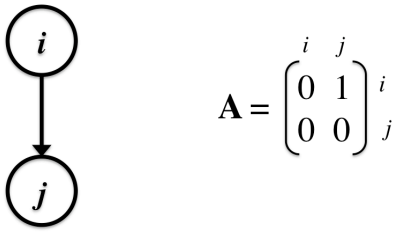

Figure 2.1: Node j cites node i, and corresponding adjacency matrix, $\mathbf{A}$

Graph kernels help to compute implicit similarities between patents in a high-dimensional feature space. In this application, they are leveraged to consider citations that are indirect citations by applying the graph kernel to the adjacency matrix, **A**. Graph kernels use an adjacency matrix of the original citation network (or a similarity matrix of choice) as input [23, 35].

In this section, we introduce graph kernels that may be applied to patent citation networks.

Exponential diffusion kernel

The exponential diffusion kernel is defined as:

$$\mathbf{K}_{ED} = \sum_{k=0}^{\infty} \frac{\alpha^k \mathbf{A}^k}{k!} = \exp(\alpha \mathbf{A}),$$

where the elements of $\mathbf{A}^k$, a_{ij}^k, represent the number of paths from node i to node j of length k. Note that this graph kernel has the α parameter that can be use to adjust the weight given to powers of **A**.

von Neumann kernel

The von Neumann diffusion kernel [23] differs from the exponential diffusion kernel by the discounting scheme. The von Neumann diffusion kernel has an exponential discounting rate and is defined as:

$$\mathbf{K}_{VN} = \sum_{k=0}^{\infty} \alpha^k \mathbf{A}^k = (\mathbf{I} - \alpha \mathbf{A})^{-1},$$

where the discounting factor is α^k. The von Neumann kernel is well defined for $0 < \alpha < ||\mathbf{A}||_2^{-1}$, where $||\mathbf{A}||_2$ is the spectral radius of **A**.

Laplacian diffusion kernel

The Laplacian exponential diffusion is defined as [23]:

$$\mathbf{K}_{LED} = \sum_{k=0}^{\infty} \frac{\alpha^k (-\mathbf{L})^k}{k!} = \exp(-\alpha \mathbf{L}),$$

where **L** = **D** - **A**, where **D** = Diag(a_i) is the diagonal degree matrix, with diagonal entries $d_{ii} = [\mathbf{D}]_{ii} = a_i = \sum_{j=1}^{n} a_{ij}$.

2.2.2 Matrix norms

In this section we describe how to quantify the change in the similarity matrix when removing a node from the network. This quantity will be used to measure the node's centrality score and to rank nodes in importance. The matrix norm quantifies the size of a matrix using some operation on the elements of the matrix [65]. In this work, we use the matrix norm to quantify the similarity matrix both with the inclusion and the exclusion of each node. In this way, we can calculate the difference in the two values and assign the difference to be the centrality (or importance) of the node to the network, since it quantifies the value of the absence of the node. The key idea is that the larger the difference in these two cases, the more important the node is to the network. We systematically calculate the matrix norm with the exclusion of each node in the network, one at a time, and are thus are able to rank all nodes in importance. In the following paragraphs we describe the different matrix norms used.

Entry-wise norms

The entry wise matrix norms treat the $n \times n$ matrix as an vector of size n^2. For example, using the p-norm for vectors, we get:

$$||\mathbf{A}||_p \quad = \quad (\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^p)^{1/p}.$$

The special case when $p = 2$ yields the Frobenius norm, and when $p = \infty$ yields the maximum norm. The max norm is the entry-wise norm with $p = \infty$, written as $||A||_\infty = \max\{|a_{ij}|\}$.

Schatten norm

The Schatten norm is based on the singular values from the singular value decomposition. In other words, the Schatten p-norms arise when applying the p-norm to the vector of singular values of a matrix. The Schatten norm is defined as follows:

$$||\mathbf{A}||_p \quad = \quad \sum_{i=1}^{n} \sigma_{ii}^p,$$

where σ_{ii} is the ith singular value of $\mathbf{A}$ for $i = 1, 2, \ldots, n$.

2.3 Graph kernel-based SVC

2.3.1 Introduction of GKB-SVC

The motivation of graph kernel-based singular values-based centrality (GKB-SVC) is to build on and improve the singular values-based centrality (SVC) presented in [40]. In doing so, we generalize the centrality measure by allowing for weighting of indirect citations of different lengths, thus making a more robust measure in order to better identify influential patents (*i.e.*, the core technologies) using the graph similarity matrix to explain the relationship of the nodes (patents), and graph kernels to draw intrinsic information from the citation network structure. In particular, the graph kernel parameter in the developed methods allow the weighting of direct and indirect citations to be tuned based on the application. For example, in some applications, it may be desirable to have the indirect citations substantially weighted, while in other applications it may be desirable to only substantially weight direct citations and citations of lesser lengths. We are able to achieve very similar node centrality ranking as both out-degree centrality and SVC, while adding the flexibility offered by graph kernel parameters.

The proposed GKB-SVC consists of 1) composition and graph kernelization of the similarity matrix of nodes 2) executing the singular value decomposition with the resulting modified similarity matrix 3) calculating the matrix norm using the obtained singular values, when removing each node from the singular matrix, one at a time. Figure 2.2 shows the process flow of our proposed centrality measure for identifying the most influential patent in the directed patent citation network. While the method of computing the difference of the similarity matrix norm with the existence and nonexistence of node in the network works as one way to rank nodes in a patent citation network, we propose improving original SVC in two key ways. The contribution of our approach is to:

1. Leverage the graph kernelization of adjacency matrix $\mathbf{A}$.

2. Remove effect of ith row only, corresponding to patent i, from kernelized matrix (do not change ith column, but leave it as it is in the kernelized matrix).

First, the similarity scores between all pairs of nodes are calculated. In this chapter, we adopt graph kernels such as: the exponential diffusion graph kernel and the von Neumann graph kernel applied to the graph adjacency matrix.

We calculate the effect of each node by comparing the inclusion of the node to the exclusion of the node in the network, and hence the similarity matrix. The node resulting in the largest difference is considered the most important node. We denote the SVD of the kernelized adjacency matrix after removing the effect of node t by:

$$\mathbf{K}_{(-t)} \quad = \quad \mathbf{U}_{(-t)}\Sigma_{(-t)}\mathbf{V}^{\mathbf{T}}_{(-t)},$$

where the Schatten norms for the kernelized adjacency matrix after removing the effect of node t are given as:

$$||\mathbf{K}_{(-t)}||_p \quad = \quad \sum_{i=1}^{l} \sigma^p_{(-t)_{ii}}.$$

The GKB-SVC then is formulated as the difference of the two norms:

$$p_p(t) = ||\mathbf{K}||_p - ||\mathbf{K}_{(-t)}||_p = \sum_{i=1}^{l}(\sigma_{ii})^p - \sum_{i=1}^{l}(\sigma_{(-t)_{ii}})^p,$$

where $\sigma_{(-t)_{ii}}$ is the ith singular value when node t is removed, and l is the predetermined number of elements to get the best lower rank approximation of the matrix $\mathbf{K}$. In a small network $l = n$, while in a large network, l is determined by experiment. Reducing the original matrix, $\mathbf{K}$, to a thin matrix using singular values can be interpreted as a form of noise suppression and makes it possible to more efficiently calculate the updates of the singular values for an extremely large network [10].

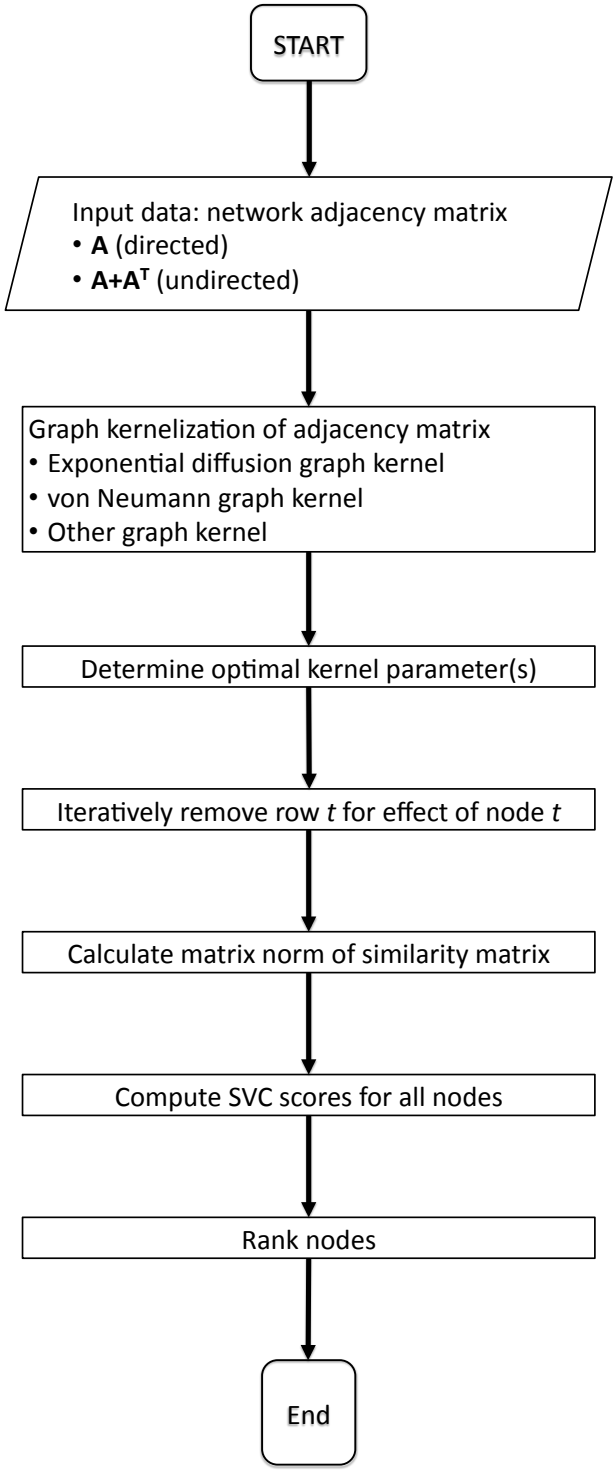

Figure 2.2: Process for ranking nodes of a citation network using proposed GKB-SVC

When using $p = 1$, the nuclear norm, Equation 2.1 can be written as:

$$p_1(t) = ||\mathbf{K}||_1 - ||\mathbf{K}_{(-t)}||_1 = \sum_{i=1}^{l}(\sigma_{ii}) - \sum_{i=1}^{l}(\sigma_{(-t)_{ii}}).$$

Similarly, when using $p = 2$, the Frobenius norm, Equation 2.1 can be written as:

$$p_2(t) = ||\mathbf{K}||_2 - ||\mathbf{K}_{(-t)}||_2 = \sum_{i=1}^{l}(\sigma_{ii})^2 - \sum_{i=1}^{l}(\sigma_{(-t)_{ii}})^2.$$

Finally, the spectral norm ($p = \infty$) makes for the simplest norm, as it uses only the largest singular value rather than a summation. The centrality measure based on spectral norm is the difference of the largest singular values between the case of the existence and the nonexistence of each node, and is given by:

$$p_\infty(t) = \sigma_{11} - \sigma_{(-t)_{11}}.$$

2.3.2 Properties of GKB-SVC for patent citation networks

To emphasize citation direction, in this work we propose only removing the row of the graph kernelization of the adjacency matrix, $\mathbf{A}$, corresponding to a given node. The original SVC does not use this approach and as a result, has a drawback that ranks some leaf nodes too highly. In order to rank node, SVC removes ith row and ith column of the similarity matrix when calculating the difference in the existence and nonexistence of a node i in the network. In this way, nodes that have many paths leading to them, such as leaf nodes, are ranked too highly as compared to nodes that are not leaf nodes (*i.e.*, nodes that are cited by other nodes). We propose modifying SVC by only removing the ith rows of the similarity matrix, and not the ith column, since the row of the matrix indicates how a patent is cited, while the column indicates how many other patents the particular patent cites. In this way, we acknowledge that the *cited* patent is more important than the *citing* patent. In this work, we remove the effect of node i by setting values in row i of matrix $\mathbf{K}$ to zero, while leaving column values of the matrix unchanged, when comparing the existence to the nonexistence of node i in a patent citation network. In Proposition 1 we relate the effect of removing

only the row in our proposed method to the existing out-degree centrality measure. See Appendix A for proof.

Proposition 1 If $\mathbf{A}$ is the adjacency matrix, then the proposed remove row modification to SVC, using the Frobenius norm (p=2) for the directed patent citation network, is equal to out-degree centrality given by:

$$\begin{aligned} p_2(t) &= \sum_{i=1}^{n}(\sigma_{ii})^2 - \sum_{i=1}^{n}(\sigma_{(-t)_{ii}})^2 \\ &= \text{out-degree of node } t, \end{aligned}$$

where σ_{ii} is the ith singular value of $\mathbf{A}$ and $\sigma_{(-t)_{ii}}$ is the ith singular value having removed the effect of node t.

One of the advantages our GKB-SVC measure provides over other methods is the flexibility allowed by way of choosing the α parameter value. For example, in the von Neumann graph kernel, a smaller α ($\alpha < 1$) means that smaller degrees of separation are given more weight. A larger α ($\alpha > 1$) means that higher degrees of separation are given more weight. A special case is $\alpha = 1$, where paths of all lengths have the same weight. This results follows from the fact that the kth power of α will be the coefficient for kth power of the adjacency matrix, $\mathbf{A}$, as can be seen in Equation 2.1. The weights will be applied to all decedents of a node when calculating the nodes centrality score, depending on how many levels from the node each decedent is located. The *descendants* of a node are all the nodes along the path from that node to a terminal node. A *terminal node* or a *leaf* is a node with out-degree of zero. Proposition 2 shows that when the same weight for direct and indirect citations are applied (*i.e.* $\alpha = 1$) the total number of paths to all decedents can be calculated using von Neumann GKB-SVC. Specifically, Proposition 2 shows that our proposed centrality measure counts the total number of paths to decedents of a node when the graph kernel parameter $\alpha = 1$. See Appendix B for proof.

Proposition 2 The total number of paths, of length 1 to m, where m is the longest path of any indirect citation for a given node t, is equal to one less than the remove row von Neumann graph kernel score using the entry-wise matrix norm and graph kernel

parameter $\alpha = 1$.

We can specifically leverage the flexibility of GKB-SVC applied to patent citation networks. In particular, for the exponential diffusion graph kernel and the von Neumann graph kernel, we can choose α parameter for best performance in any given network, as we will explore in the next section.

2.4 Experimental results

In this section we experiment with different graph kernel parameters and compare our proposed GKB-SVC measure with existing centrality measures using artificial datasets. The centrality measures considered for comparison purposes are out-degree centrality, original SVC, and GKB-SVC.

2.4.1 Data description: Example patent citation networks

Artificial dataset 1: 10-node patent citation network

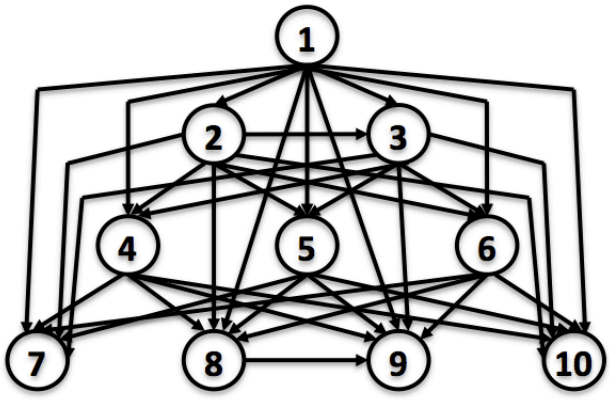

Figure 2.3: Artificial dataset 1 - example patent citation network called Graph 1

We first use the 10-node example citation network seen in Figure 2.3. Note that Node 1 is the root node of this acyclic graph. Additionally note that all other patents in this network either directly cite Node 1, or indirectly cite Node 1, by citing a node that itself cites Node 1, either directly or indirectly. Also, it is possible for there to

exist more than one citation path between any pair of nodes. Take Node 1 and Node 10 for example. Between these two nodes there are the paths 1→3→6→10 and 1→6→10 and 1→5→10, and others. In total, there are 16 paths between Node 1 and Node 10, of lengths 1, 2, 3, or 4. The adjacency matrix for the patent citation network in dataset 1 is given by,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Next we discuss the expected ranking of all 10 nodes for the network in Figure 2.3. The advantage of this example network is that we can systematically determine how we expect the nodes of the network to be ranked. We expect that Nodes 7, 9 and 10 are the least important nodes and scored similarly since they have similar citation structure, and are the least cited of all patents, being cited zero times each. Note the zero entries in the 7th, 9th, and 10th row of the adjacency matrix, **A** In a similar way, Nodes 8 is the next-most least important node since it is only cited by one other node, Node 9. Thus, we expect Node 8's score to be greater than that of Nodes 7, 9, and 10. We expect that Nodes 2 and 3 are not as highly ranked as Node 1, but are more highly ranked than Nodes 4, 5, and 6, which we expect to be scored the same, due to them having the same citation structure and citation count. To summarize then, our expected ranking of the nodes for Graph 1 in descending order, from most important to least important are shown in Table 2.1.

Artificial dataset 2: Six 10-node patent citation networks

Next we continue our comparison of centrality measures by scoring Node 1 (*i.e.*, the *root* node) of six example graphs using existing methods and our proposed method.

Table 2.1: Expected node influence rank for example citation Graph 1

Expected rank	Node
1	1
2	2
3	3
4	4, 5, 6 (tie)
7	8
8	7, 9, 10 (tie)

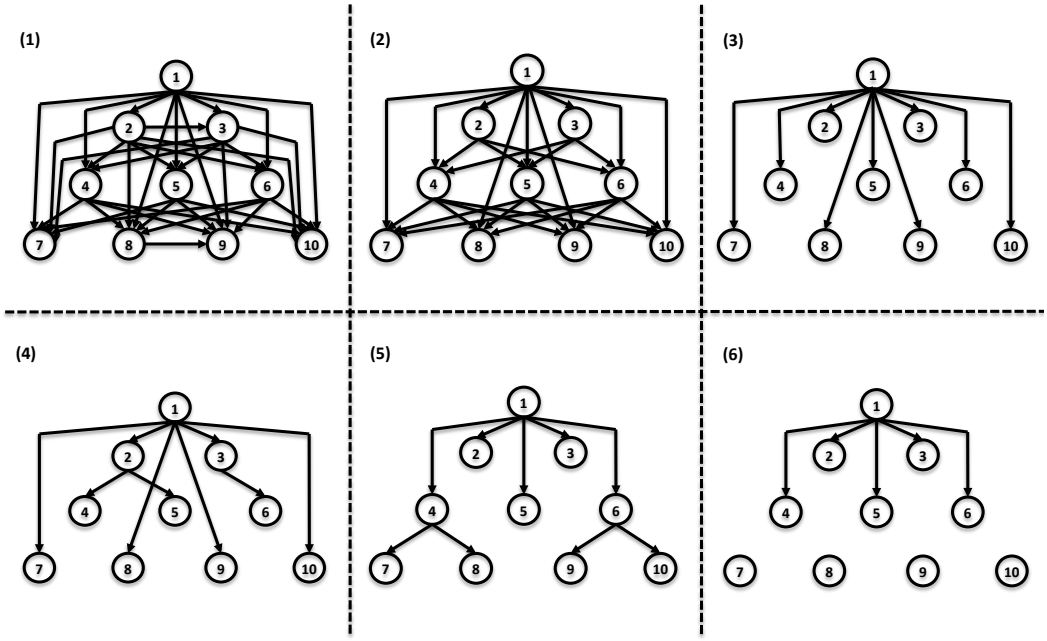

Figure 2.4: Artificial dataset 2 - six example patent citation networks with labels

We compare the scorings of Node 1 for the different graphs and use those root node scorings as an indicator of the importance of the graphs themselves. Note that this is a scoring of *root nodes over six different graphs*, seen in Figure 2.4, and not the usually performed ranking of all nodes within a single graph [45].

We consider the direct and indirect citations for all Node 1s of the six citation networks in dataset 2. We expect Node 1 of Graph 1 to be the highest ranking of all the six Node 1s of the dataset since it has the most direct and indirect citations. Generally, we expect the ranking to decrease from Graph 1 to Graph 6, since the total number of direct and indirect citations decreases. We summarize our expected ranking of the Node 1s for the six example networks in descending order, from most important to least important are shown in Table 2.2.

Table 2.2: Expected Node 1 influence rank for six example citation networks from dataset 2

Expected rank	Node 1 from network
1	1
2	2
3	3
4	4
5	5
6	6

Artificial dataset 3: 15-node patent citation network

The network in artificial dataset 3 is like Graph 1, but has five additional nodes that each cite Node 7, as seen in Figure 2.5. In other words Node 7 now has out-degree of five in this patent citation network. Clearly, this change will cause Node 7 to be considered more important than it previously had been in dataset 1. If we compare Node 7 to Node 4, we see that Node 7 has a higher out-degree. However, Node 4 has more indirect citations than Node 7. We expect that with these additional citations, the ranking relationship between Node 4 and Node 7 will depend on the graph kernel parameter used.

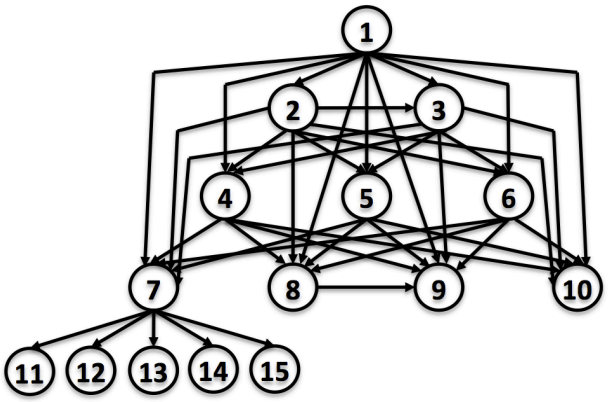

Figure 2.5: Artificial dataset 3 - example citation graph with five additional nodes citing Node 7

2.4.2 Parameter selection for GKB-SVC

We use the example citation network in Figure 2.3 to compare our proposed approach with existing methods and find that our method outperforms original SVC that was proposed by Kim *et al.*. Notice in Table 2.3 that Node 9 is ranked second, thus is ranked too highly. Compare that to Node 9's expected ranking of eighth in Table 2.1. This is because it is a leaf node and has many indirect paths to it. When Kim's SVC is performed, it removes both the ith row and ith column, meaning that the node's incoming incitation and outgoing citations are considered for importance rankings, resulting in Node 9 being ranked important because its exclusion has a large effect on the similarity matrix.

Table 2.3: Node rankings of Graph 1 using original SVC approach

Rank	Node
1	1
2	9
3	2
4	7, 10
6	8
7	3
8	4,5,6

We score and rank all nodes of Graph 1 using our proposed graph kernel-based SVC and the exponential diffusion and von Neumann graph kernel with $\alpha = \{0.01, 0.5, 0.85, 2.0\}$. In this experiment, we find that $\alpha >= 0.5$ (from this set of values) yields the best node scoring and ranking results. The reason that lesser α values, such as $\alpha = 0.01$, do not perform as well is because the indirect citations are weighted sufficiently small that the GKB-SVC ranking becomes similar to out-degree centrality. Note that the concept of GKB-SVC ranking nodes similarly to out-degree centrality for small α values is also seen for the real-life data experimental results.

2.4.3 Exponential diffusion graph kernel

Artificial dataset 1

Exponential diffusion graph kernel parameter $\alpha = 0.85$ and p=2 norm has the best ranking performance.

Table 2.4: Artificial dataset 1: effect of α parameter and p norm on ranking nodes of Graph 1 when using exponential diffusion GKB-SVC

	$p=1$				$p=2$				$p=\infty$				
	α				α				α				Expected
Node	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	rank
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4	4	4	4
7	8	8	8	8	8	8	8	8	7	9	9	9	8
8	7	7	7	7	7	7	7	7	9	7	7	7	7
9	10	10	10	10	8	8	8	8	10	8	8	8	8
10	8	8	8	8	8	8	8	8	7	9	9	9	8

In Table we observe $\alpha = 0.5$ produce a better ranking, since Node 8, is now higher than Node 10 for all tree p norms. Nodes 7, 9, and 10, do not all have the same score, as is expected, when using the $p = 1$ norm. Our best observed ranking results for $\alpha = 0.85$ is using the $p = 2$ norm.

Artificial dataset 2

Exponential diffusion graph kernel parameter $\alpha = 0.85$ and p=2 norm has the best ranking performance.

Table 2.5: Artificial dataset 2: effect of α parameter and p norm on ranking Node 1s of dataset 2 when using exponential diffusion GKB-SVC

	$p=1$				$p=2$				$p=\infty$				
	α				α				α				Expected
Node	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	rank
1	1	1	1	1	1	1	1	1	4	1	1	1	1
2	2	2	2	2	2	2	2	2	3	2	2	2	2
3	3	3	3	3	3	3	3	3	1	3	3	3	3
4	4	4	4	4	4	4	4	3	5	5	5	6	4
5	5	5	5	5	5	5	5	3	6	6	6	5	5
6	5	6	6	6	6	6	6	6	2	4	4	4	6

Artificial dataset 3

Depending on α parameter, Node 7 may be ranked fourth or seventh most important.

2.4.4 von Neumann graph kernel

Artificial dataset 1

von Neumann graph kernel parameter $\alpha = 0.85$ and p=2 norm has the best ranking performance.

Artificial dataset 2

von Neumann graph kernel parameter $\alpha = 0.85$ and p=2 norm has the best ranking performance.

Table 2.6: Artificial dataset 3: effect of α parameter and p norms on ranking nodes of dataset 3 when using exponential diffusion GKB-SVC

	$p=1$				$p=2$				$p=\infty$			
	α				α				α			
Node	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	5	5	5	5	5	4	4	4	4	4	4
5	4	5	5	5	6	5	4	4	4	4	4	4
6	4	5	5	5	6	5	4	4	4	4	4	4
7	7	4	4	4	4	4	7	7	7	7	7	7
8	13	13	13	8	8	8	8	8	9	8	8	8
9	15	15	15	15	9	9	9	9	10	9	9	9
10	14	14	14	14	9	9	9	9	8	10	10	10
11	8	8	8	9	9	9	9	9	11	11	11	11
12	8	8	8	9	9	9	9	9	11	11	11	11
13	8	8	8	9	9	9	9	9	11	11	11	11
14	8	8	8	9	9	9	9	9	11	11	11	11
15	8	8	8	9	9	9	9	9	11	11	11	11

Table 2.7: Artificial dataset 1: effect of α parameter and p norm on ranking nodes of Graph 1 when using von Neumann GKB-SVC

	$p=1$				$p=2$				$p=\infty$				Expected
	α				α				α				
Node	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	rank
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4	4	4	4
7	9	8	8	7	8	8	8	8	7	9	9	9	8
8	7	7	7	9	7	7	7	7	9	7	7	7	7
9	8	10	10	10	8	8	8	8	10	8	8	8	8
10	9	8	8	7	8	8	8	8	7	9	9	9	8

Table 2.8: Artificial dataset 2: effect of α parameter and p norm on ranking Node 1s of dataset 2 when using von Neumann GKB-SVC

	$p=1$				$p=2$				$p=\infty$				
	α				α				α				Expected
Node	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	rank
1	1	1	1	1	1	1	1	1	4	1	1	1	1
2	2	2	2	2	2	2	2	2	3	2	2	2	2
3	3	3	3	5	3	3	3	5	1	3	3	5	3
4	4	4	4	4	4	4	4	4	5	5	6	4	4
5	5	5	5	3	5	5	5	3	6	6	5	3	5
6	5	6	6	6	6	6	6	6	2	4	4	6	6

Artificial dataset 3

Depending on α parameter, Node 7 may be ranked fourth or seventh most important. Recall again the flexibility our method allows by way of choosing the α parameter value. A smaller α means that smaller degrees of indirect citation separation are given more weight. A larger α means that higher degrees of indirect citation separation are given more weight. This results follows from the fact that a power of α will be the coefficient for the powers of the adjacency matrix, $\mathbf{A}$.

We see that with the α values we used in the raking of the 15-node network, the node ranking results are not sensitive to the graph kernel type. That is, we get fairly consistent ranking results with the both the exponential diffusion and von Neumann graph kernels that we experimented with here.

We see that overall graph kernel parameter $\alpha = 0.85$ and p=2 norm has the best ranking performance, and that results are not very sensitive to graph kernel type used, among the two kernels used in this experiment.

2.4.5 Guideline for the selection of a graph kernel and its parameter

Given that graph kernels provide flexibility in weighing of indirect citations, the decision must be made as to which graph kernel to apply and how to select the best value of the graph kernel parameter. The tuning of graph kernel parameters will depend on the particular application, so it can vary from application to application [23]. As a part

Table 2.9: Artificial dataset 3: effect of α parameter and p norm on ranking node of dataset 3 when using von Neumann GKB-SVC

	$p=1$				$p=2$				$p=\infty$			
	α				α				α			
Node	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0	0.01	0.5	0.85	2.0
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	10	10	11	5	4	4	4	4	4	4	4
5	4	10	10	11	5	4	4	4	4	4	4	4
6	4	10	10	11	5	4	4	4	4	4	4	4
7	12	4	4	4	4	7	7	7	7	7	7	7
8	13	13	13	5	8	8	8	8	9	8	8	8
9	14	15	15	15	9	9	9	9	10	9	9	9
10	15	14	14	14	9	9	9	9	8	10	10	15
11	7	5	5	6	9	9	9	9	11	11	11	10
12	7	5	5	6	9	9	9	9	11	11	11	10
13	7	5	5	6	9	9	9	9	11	11	11	10
14	7	5	5	6	9	9	9	9	11	11	11	10
15	7	5	5	6	9	9	9	9	11	11	11	10

of this work, we experimented with different graph kernels, graph kernel parameters, and matrix norms. To emphasize the direct citations, and rank more highly the nodes that have more direct citations, we can use small α. To emphasize the indirect citations we can use a larger α. This results from greater α values giving greater weight to matrices of higher order in the expansion of the von Neumann graph kernel.

We experimented with different matrix norms and α parameter values using the two small artificial datasets described above, for which the expected relative influence ranking of nodes is well-understood, by design. For example, considering only direct citations, a patent with nine direct citations should be considered more influential than a patent with one direct citation. Further, considering both direct and indirect citations, a patent with one direct citation and two indirect citations should be considered more influential than a patent with one direct citation and one indirect citation. We experimented with different matrix norms and α parameters, comparing influence ranking results over the different matrix norms and α parameters to the expected ranking. We

then choose matrix norm and α parameter value that produce the same influence ranking results as the expected ranking in the case of the two artificial datasets. The idea is that a large patent citation network can be thought of as composed of many small citation networks. If the ranking results are desirable in small citation networks (artificial datasets), then ranking results will also be desirable when extended to the large citation network. Based on extensive experiments, for the application of ranking patent importance in patent citation networks, we recommend the following as GKB-SVC parameters: von Neumann graph kernel, with $\alpha = 0.85$, and Schatten norm with $p = 2$. For the experiments that follow, we fix the graph kernel, graph kernel parameter, and matrix norm as such. We use the von Neumann graph kernel because its interpretation in relation to the patent citation network dataset is reasonable, as the von Neumann graph kernel considers the powers of the adjacency matrix, which correspond to direct and indirect citations. Other kernels do not have a clear interpretation to the patent citation network.

2.4.6 Comparison of out-degree centrality, original SVC, and GKB-SVC

In this section we compare the ranking of all 10 nodes of Graph 1 seen in Figure 2.3 using the three methods described. We compare these results with the expected ranking. In particular, we leverage the exponential diffusion and von Neumann graph kernel, as graph kernels help to compute implicit similarities between patents in a high-dimensional feature space.

(1) Out-degree centrality

(2) Closeness centrality

(3) Betweenness centrality

(4) Original SVC

(5) Proposed GKB-SVC

Artificial dataset 1

Results for the case of ranking the 10 nodes in dataset 1 can be seen in Table 2.10. There we see our proposed GKB-SVC and out-degree centrality perform the best because they provide the ranking as is expected. Original SVC ranks Node 9 too highly since it considers citations and times cited as equally important. Refer to Table 2.2 for detail on expected ranking of nodes.

Table 2.10: Comparison of centrality measures for ranking 10 nodes from dataset 1

	Out-degree		Closeness		Betweenesss		SVC		GKB-SVC		Expected
Node	Patent	Score	Patent	Score	Patent	Score	Patent	Score	Patent	Score	rank
1	1	9	1	0.11	1	0.07	1	7.93	1	621.19	1
2	2	7	2	0.10	2	0.00	3	2.22	2	173.64	2
3	3	6	2	0.10	2	0.00	7	0.87	3	49.69	3
4	4	4	4	0.09	2	0.00	8	0.35	4	5.64	4
5	4	4	4	0.09	2	0.00	8	0.35	4	5.64	4
6	4	4	4	0.09	2	0.00	8	0.35	4	5.64	4
7	8	0	7	0.08	2	0.00	4	1.89	8	1.00	8
8	7	1	7	0.08	2	0.00	6	1.38	7	1.72	7
9	8	0	7	0.08	2	0.00	2	4.36	8	1.00	8
10	8	0	7	0.08	2	0.00	4	1.89	8	1.00	8

Artificial dataset 2

Results for the case of ranking the six Node 1s in dataset 2 can be seen in Table 2.11. Our proposed GKB-SVC performs the best because it provides the ranking as is expected. For this dataset, out-degree is unable to distinguish the difference in importance of citation networks 1, 2, and 3, since it only considers direct citations of Node 1, which are the same for those three graphs, while clearly Node 1 from network 1 has the greatest importance among the example networks provided. Original SVC ranks Node 1 from citation network 6 as fourth most important, ahead of Node 1 from network 4, which is not substantiated since Node 1 from network 4 has six direct citations and three indirect citations, while Node 1 in network 6 has only five direct citations and no indirect citations.

Table 2.11: Comparison of centrality measures for ranking Node 1s from dataset 2

	Out-degree		Closeness		Betweenesss		SVC		GKB-SVC		Expected
Node	Patent	Score	Patent	Score	Patent	Score	Patent	Score	Patent	Score	rank
1	1	9	1	0.1111	6	0.07	1	7.93	1	621.19	1
2	1	9	1	0.1111	5	0.19	2	5.51	2	197.93	2
3	1	9	1	0.1111	1	1.60	3	3.00	3	7.50	3
4	4	6	4	0.0833	2	1.42	5	1.23	4	6.90	4
5	5	5	5	0.0769	3	1.33	6	1.13	5	6.70	5
6	5	5	6	0.0000	4	0.44	4	2.24	6	4.61	6

2.5 Case study

In this section, the relative performance of the existing centrality measures and the proposed methods are compared using the Coefficient of Variation (CV), which is a method to evaluate the discrimination ability of the centrality measures [40]. The centrality measures considered for comparison in this case study are out-degree centrality, original SVC, and the graph kernel-based SVC proposed in this chapter.

The original pool of patents used for the computational experiments includes U.S. patents in the area of information and security issued from 1976 to 2007 [40]. The dataset actually used for the experiments are U.S. patents in the area of information and security issued between 1994 and 2007. For these experiments, we take the top 1% most frequently cited patents from 1994 to 2007 as the nodes in the patent citation network. In order to have a single connected tree structure to which to apply centrality measures, we select the patents that cite, either directly or indirectly, the most cited patent from the original dataset, which is patent US-5349655. Our patent citation network then consist of 4,241 nodes and 18,385 edges.

In this experiment for the graph kernel-based SVC, the singular value decomposition is performed on the similarity matrix with 4,241 columns and rows. In Table 2.14 we show results from ranking real-life patents. We see that patent US-5892900 is the highest ranked patent using out-degree centrality, and Patent while Patent US-5349655 is the the higher ranked patent when using GKB-SVC. Here again, we observe the nature of out-degree centrality measure, and the difference that considering indirect citations makes. Additionally, we see that original SVC ranks leaf nodes, which may

have many indirect paths leading to them, too highly, as original SVC rankings do not have many top 20 ranked patents in common with the other three approaches.

We use the top 1% of patents that directly or indirectly cite the most cited, Patent US-5349655, so we would expect that patent to be ranked among the most important. Observe the flexibility in our proposed approach since US-5349655 is ranked as the most important node when using $\alpha = 0.85$, while for lesser values, such as $\alpha = 0.5$, we see that US-5745604 is ranked first and US-5349655 is ranked second in importance (not shown in table).

Table 2.12: Comparison of ranking of U.S. patents using different centrality measures 'US-' prefix omitted)

Rank	Out-degree	SVC	GKB-SVC
1	5892900	7266217	5349655
2	5982891	7281133	5745604
3	5943422	7171020	5892900
4	5920861	7224819	5862260
5	5910987	7305104	5822436
6	5915019	7076445	6064764
7	5917912	7254249	5943422
8	6185683	7184572	5920861
9	5949876	7197160	5832119
10	6112181	7249257	6301590
11	5862260	7159240	5982891
12	6226618	7310823	6047296
13	6122403	7239734	6052486
14	5745604	7257707	5613002
15	6157721	7136502	5915019
16	6253193	7286685	5949876
17	6237786	7197461	6122403
18	6327652	7210034	5910987
19	6330670	7302574	6338070
20	6138119	7248717	6185683

In addition to comparing the ranking results of the 20 highest ranked nodes when applying different centrality measures, we also use CV, which is a normalized measure to quantify the spread of the data. CV is defined as follows, CV=$\frac{\sigma}{\mu}$, where σ is the standard deviation of the top 20 scores, and μ is the mean of the top 20 scores. Because CV is a measure of how well importance ranking is distinguished, the greater the CV,

the more desirable. Table 2.13 displays the CV for the different centrality measures applied to real-life data. In Table 2.13 we see that the proposed GKB-SVC performs better than existing centrality measures when considering how well importance scores distinguish patent rankings.

Table 2.13: Coefficient of variation for centrality scores of top 20 ranked nodes

Centrality measure	Average	Standard deviation	CV
Out-degree centrality	183.80	58.37	0.32
SVC	58808.74	60371.51	1.03
GKB-SVC	430961.62	1232572.6	**2.86**

We plot the citation network using the top 20 nodes based on our GKB-SVC centrality measure. If an edge exists between a top 20 node, and another top 20 node, then that edge is also drawn. The size of the node is proportional to its centrality score, with a larger node size indicating a larger centrality score. In Figure 2.6 we show a plot of a subset of the nodes and edges.

In addition to comparing the ranking results of the 20 highest ranked patents when applying different centrality measures, we also evaluate CV for the top 50 highest ranked patents quantify the spread of the data. Once again, because CV is a measure of how well importance ranking is distinguished, the greater the CV, the more desirable. Table 2.15 displays the CV for the different centrality measures applied to real-life data for the top 50 ranked patents. Since we are now considering more (additional) scores, the CV results will be different. In Table 2.15 we see that the proposed GKB-SVC still performs better than existing centrality measures when considering how well importance scores distinguish the top 50 ranked patents.

2.6 Conclusion

In this chapter, we present a graph kernel-based method for ranking patents in influence given a patent citation network. Specifically we propose the von Nuemann graph kernel to weigh both the direct and the indirect citations that a patent receives from later patents, in order to evaluate patent influence. The presented methods were

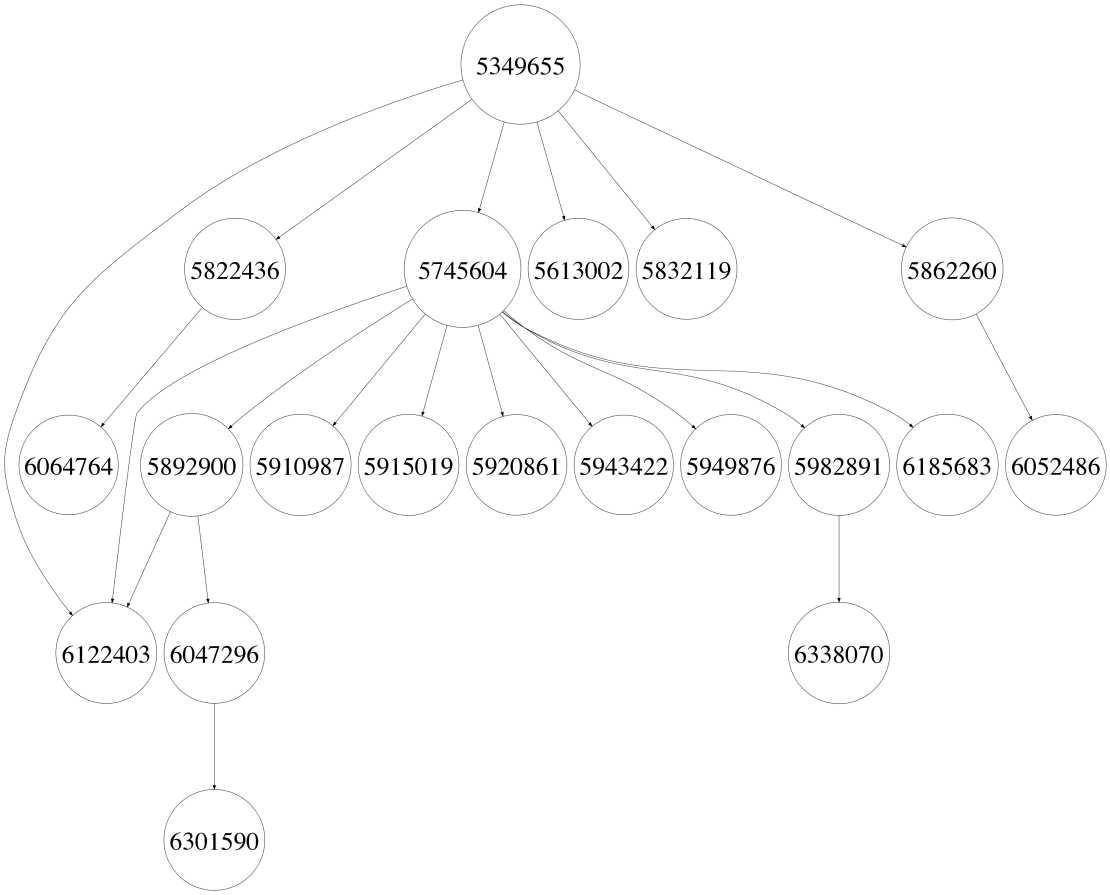

Figure 2.6: Patent citation network of the top 20 ranked important nodes using proposed von Neumann GKB-SVC with α=0.85 ('US-' prefix omitted)

Table 2.14: Comparison of ranking of U.S. patents using different centrality measures ('US-' prefix omitted)

	Out-degree		SVC (remove col.)		SVC (remove row)		GKB-SVC	
Rank	**Patent**	**Score**	**Patent**	**Score**	**Patent**	**Score**	**Patent**	**Score**
1	5892900	383	7266217	228288.94	5349655	1066981.05	5349655	4888907.01
2	5982891	249	7281133	191601.19	5745604	736123.08	5745604	2934315.58
3	5943422	240	7171020	114041.82	5892900	43737.29	5892900	156147.09
4	5920861	211	7224819	112578.89	5862260	36031.10	5862260	132886.45
5	5910987	201	7305104	99208.33	5822436	25995.72	5822436	105942.50
6	5915019	193	7076445	51187.34	6064764	19785.15	6064764	73534.48
7	5917912	193	7254249	43034.75	5943422	16728.27	5943422	53760.46
8	6185683	191	7184572	42847.58	5920861	13603.13	5920861	43824.74
9	5949876	183	7197160	36113.06	5832119	8937.72	5832119	32579.09
10	6112181	177	7249257	33727.89	6301590	8713.12	6301590	28903.35
11	5862260	175	7159240	32202.77	5982891	7253.75	5982891	23901.15
12	6226618	155	7310823	28926.90	5915019	6229.13	6047296	20890.77
13	6122403	154	7239734	27992.95	6052486	6025.27	6052486	20885.94
14	5745604	151	7257707	25856.69	5949876	5157.02	5613002	19465.97
15	6157721	145	7136502	20858.23	6122403	4954.15	5915019	18232.00
16	6253193	141	7286685	20627.20	5613002	4922.71	5949876	16781.38
17	6237786	137	7197461	19003.55	6047296	4824.71	6122403	14462.24
18	6327652	133	7210034	17301.14	5910987	4345.28	5910987	12004.94
19	6330670	133	7302574	15479.36	6185683	4073.57	6338070	11215.34
20	6138119	131	7248717	15296.15	6112181	3702.97	6185683	10591.89
21	6233684	124	7305553	13896.40	5917912	3678.54	6112181	10225.83
22	6311214	122	7278168	13022.05	6338070	3315.35	5917912	10092.34
23	6240185	121	7281274	11942.95	6157721	3123.07	6253193	8802.53
24	6389402	120	7266181	11678.15	6253193	2839.04	6157721	8055.41
25	6292569	119	7292692	11021.17	6138119	2512.59	6246777	7091.52
26	6363488	119	7289643	10714.20	6311214	2498.05	6311214	6772.44
27	5832119	107	7286667	10713.36	6246777	2465.96	6138119	6465.70
28	6345256	98	7287168	10713.36	6452915	1873.05	5826013	5652.22
29	6243480	92	7292691	10713.36	5930767	1840.87	6363209	5646.75
30	5822436	89	7302709	10433.82	5826013	1795.36	5930767	5572.54
31	6314409	85	7137004	10285.37	6243480	1646.52	6452915	4989.27
32	6427140	82	7139408	9879.76	6332031	1570.66	5765030	4696.82
33	6064764	80	7107463	9692.79	6363209	1513.48	6332031	4381.06
34	5349655	79	7302058	9202.21	6233684	1490.30	6243480	4228.81
35	6385596	78	7266704	8881.11	6237786	1473.54	6233684	3820.15
36	6345104	73	7263187	8412.81	6240185	1382.73	6275599	3809.62
37	6449367	73	7310422	8350.77	5765030	1353.12	6237786	3426.66
38	6614914	71	7305592	8149.52	6275599	1322.65	6499059	3278.72
39	6452915	63	7287159	7181.92	6292569	1244.37	6240185	3218.02
40	5930767	59	7299499	6826.34	6499059	1126.01	5822517	3212.33
41	6343138	58	7213757	6588.20	5822517	951.98	6292569	2838.73
42	6108644	57	7159210	6525.24	6026193	924.31	5854916	2706.00
43	6609199	57	7076655	6470.28	6345104	919.89	6026193	2686.74
44	6236365	56	7308576	6335.77	6285776	883.48	6345104	2421.40
45	6332031	55	7299358	6308.75	5854916	843.13	6285776	2399.89
46	6275599	52	7302689	6015.98	6324573	743.29	6324573	1883.64
47	6658568	52	7209573	5996.56	6343138	706.68	6567796	1700.14
48	6246777	50	7277695	5971.13	6567796	703.84	6286036	1654.21
49	6249252	50	6993154	5889.43	6327652	648.69	6343138	1609.11
50	6263313	47	7213269	5836.80	6507817	590.47	6424979	1480.26

Table 2.15: Coefficient of variation for centrality scores of top 50 ranked nodes

Centrality measure	Average	Standard deviation	CV
Out-degree centrality	121.28	66.16	0.55
SVC (rm col)	28796.49	45048.42	1.56
SVC (rm row)	41522.1	180667.35	4.35
GKB-SVC	175081.02	796012.32	**4.55**

specifically developed to be applied to patent citation networks, but may also be applied to literature citation networks, where there is a natural sense of later works being influenced by prior works. The proposed methods have some limitation in that older (*i.e.*, from an earlier date) patents tend to be considered more influential within the patent citation network because they tend to accumulate more direct and indirect citations with time.

The study utilizes the idea that the change in similarity matrix caused by removing a node from the network is valuable for determining the importance of that node in the network, since each node contributes to the similarity matrix. The proposed GKB-SVC approach works by leveraging graph kernels in order to generalize the SVC approach. Additionally, this approach works by removing the effect of only row values (*i.e.*, setting elements in row i of the similarity matrix to zero as an indication of its nonexistence in the citation network), and not ith column values of the similarity matrix when comparing the existence to the nonexistence of node i in a patent citation network. Using this method, the impact of node i as a *cited* patent, rather than *citing* patent, is emphasized. This fact means that leaf nodes, which may have many indirect paths leading to them, are not incorrectly ranked too highly.

The von Neumann diffusion graph kernel allows us to consider citation paths of greater than length one. In this way, we are able to account for patents that do not directly cite node i, but which also cite a patent that cited node i, extended to the general case of any number of intermediate citations. Our proposed method offers more generality and flexibility than that of out-degree centrality and SVC. As a guideline for the application of node centrality in patent citation networks, we find the von

Neumann graph kernel, with $\alpha = 0.85$, and Schatten norm with $p = 2$ to yield the best performance.

In the case of the real-life dataset, we saw that our method performs better than out-degree centrality and original SVC. In particular, our method allows for the generalization of these type of centrality measures, allowing for flexibility in the application of centrality measures. Finally, our proposed GKB-SVC outperforms existing centrality measures in discriminating ranking when CV is considered.

Potential future research directions include (1) considering assigning a weight of zero for indirect citations greater than some length c, so that older patents do not have the advantage of having many long indirect citation paths that contribute to their influence score, (2) considering time information of a patent to account for the rate of citations made (*i.e.*, consider how the citation network evolves with time), and (3) extending the application of the method to identify influential publications in literature citation networks.

In our next chapter, we will move from the analysis of patent citations for the purpose of influence ranking to the analysis of co-citation and bibliographic coupling links for the purpose of pairwise patent similarity ranking.

Chapter 3

Multi-stage Similarity Measure for Calculation of Pairwise Patent Similarity in a Patent Citation Network

3.1 Introduction

With the expected increase in the number and complexity of patents, quickly analyzing patents to find similar and outlying patents or groups of patents in a patent citation network has become a critical ability and provides business advantages [26, 34, 38, 41, 47, 80]. Most new patents are influenced by previous works in some way. This influence is captured by a patent's citation of a previous work, and can be thought of as an extension of the previous work(s). Taken all together, patents and the citation links between them can be represented in a patent citation network. It is important to analyze the patent citation network to gain an understanding of past, current, and possible future technological trends [27]. Most of the existing citation network research explains the similarity for a pair of patents as either *citing* or *cited* patent, strictly on an pairwise adjacency basis. That is, only direct citation links are considered [59]. Additionally, much patent citation network research calculates the patent similarity using keywords [80, 60]. Identifying patent relationships by analyzing direct and indirect **citation** links, as well as determining quality of the citing patents is given in [6]. In our work, we focus on patent **co-citations** for the purpose of developing a similarity measure, relying only on the patent citation network structure. Using only the patent citation network structure, we are able to extract important relational evidence that can be missed when using keyword analysis since word choice depends on author writing style, whereas citations directly capture patent relationship. A co-citation link occurs when two patents are cited together by another patent. For example, if patent x and patent y are both cited by patent i, then we say patent x and y are *co-cited* by patent i.

Whereas citations are important for considering *influence* of patents, co-citations give insight into *similarity* of patents. Figure 3.1 demonstrates the difference in considering citations versus co-citations.

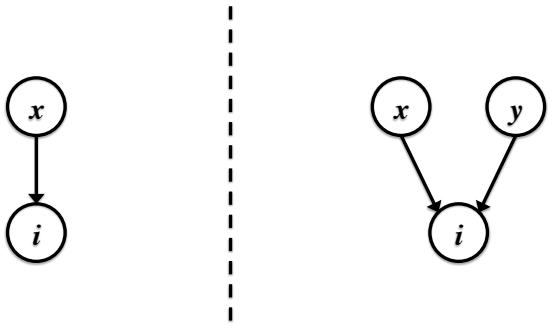

Figure 3.1: Example patent citation networks highlighting citation of x by i, indicating *influence* of patent x (left); and co-citation of x and y by i, indicating *relatedness* (or *similarity*) of patents x and y (right)

There are two main approaches used to explain the similarity (or relatedness) between nodes in a citation network when only considering the citation network structure – *co-citation* and *bibliographic coupling* [20, 16, 59]. These methods can be used for measuring the pairwise similarity between two patents of a patent citation network. Small [73] introduced co-citation to measure relatedness of scientific literature documents by their co-citation frequency. In this case, two patents are said to be co-cited if they are simultaneously cited by another patent. For example, in Figure 3.2, patents 14 and 15 are both cited by patents 17, 18, 19, and 20. This means that patents 14 and 15 are in co-citation, though they do not directly or indirectly cite each other. Patents are said to be bibliographically coupled if they have at least one same bibliographic reference in their own references [37]. In this example citation network, patents 14 and 15 both cite patents 9, 10, 11, and 12. This means that patents 14 and 15 are bibliographically coupled, though they do not have a direct or indirect citation between each other. Figure 3.2 shows examples of both co-citation and bibliographic coupling for patents 14 and 15 in an example patent citation network. In this work, we focus on co-citation as a similarity measure.

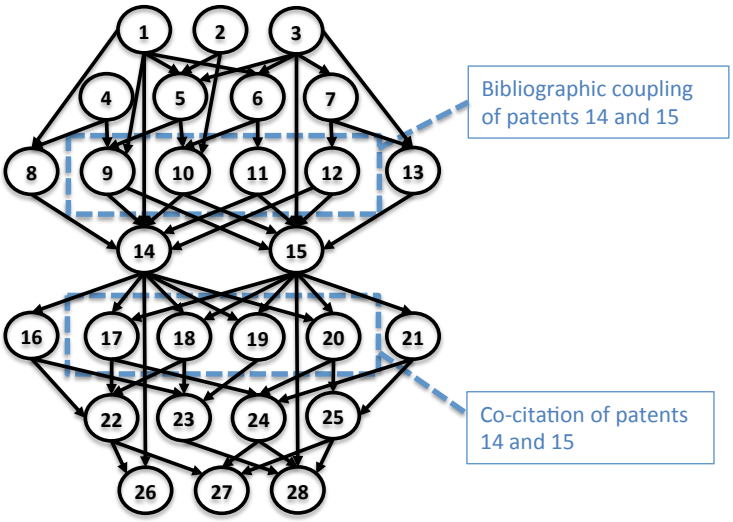

Figure 3.2: Example patent citation network highlighting bibliographic coupling and co-citation

To the best of our knowledge, no patent similarity research considers multi-stage co-citation for patent citations, and only leverages the citation network structure. That is, no patent similarity approach leverages co-citations of greater than length one in the patent citation network, while not leveraging any other patent data. By considering multi-stage co-citation, we are able to capture the importance of the citing patents by way of indirect co-citations, and we do not rely on writing style or word choice in keyword analysis.

An indirect citation means two patents are connected by one or more intermediate patents in the patent citation network. While direct citations reveal related recent prior arts, indirect citation links reveal tracks of technological change over time [81]. Considering both direct and indirect citations provides more information for assessing patent similarities. When evaluating the similarity of two patents, considering both direct and indirect co-citations leads to more complete similarity assessment, since it accounts for the immediate relationships of patents, as well as the patents' technology

track over time.

Some methods of patent citation analysis consider both direct and indirect citation links (usually these works *only consider a limited number of indirect stages* and do not leverage information about all the stages of citations). For example, multi-stage patent *citation* analysis was used by Wartburg *et al.* to measure *inventive progress* [80]. Our work is differentiated from the work in [80] since we use co-citation, rather than bibliographic coupling. In that work, similarity between a new patent and its directly cited prior patents depends on the number of patents which are cited by the later patent. For example, if a new patent cited 10 patents, then its similarity to each of those ten patents is equally 1/10. In general, if a new patent cites n patents, then its similarity to each patent is equally $1/n$. Extending the idea, similarity scores are multiplied in this way as the indirect citation path length increases. Our work is further distinguished from the work of Wartburg *et al.* since that work aims to gauge the technical value added of invention and cluster patents into technical subfields, whereas we aim to develop a similarity measure for calculating pairwise patent similarity. Wartburg *et al.* rely on expert judgement to validate the technical value added of patents. We use class codes of patents to validate the patent similarity results.

In this chapter, we propose a general method for assessing patent similarity, given a patent citation network, considering both direct and indirect co-citations. The main contribution of this chapter is developing patent similarity measures based on direct co-citation links and multi-stage (indirect) co-citation links, including a normalization technique to improve performance. It will also be shown that integration of direct and multi-stage indirect co-citation with normalization will improve the effectiveness of the similarity measure when compared to using the direct co-citation measures alone. To validate our approach, we use U.S. patent class codes as a distinct indicator of relatedness and the well-known Jaccard similarity coefficient [76].

The rest of the chapter is organized as follows. In Section 4.2 we provide background on the patent similarity measure problem. In Section 3.3 we define the new similarity measures based on direct and multi-stage co-citation. Section 3.4 provides experiment results using a real-life patent citation network. Finally, conclusions and future work

will be given in Section 3.5.

3.2 Background

The adjacency matrix of a given network, denoted **A**, is defined as follows. If patent i is cited by patent j, there is an arc between i and $j, (i, j)$. If there is an arc between patent i and patent j, then the (i, j)th element of adjacency matrix is 1, otherwise 0. For example, the citation edge represented by the arc 1→5 means that patent 1 is cited by patent 5. A simple example of node j cites node i is shown in Figure 3.3.

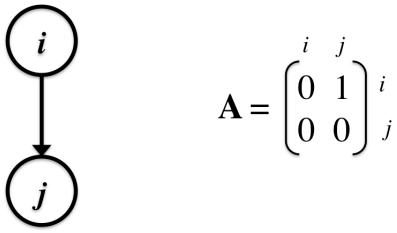

Figure 3.3: Node j cites node i, and corresponding adjacency matrix, **A**

3.2.1 Existing methods for patent similarity analysis

Patent citation analysis is based on the examination of citation links among different patents [19, 58, 20, 12]. In the area of similarity measures for patents in patent citation networks, leveraging only network structure, usually direct co-citations are considered. Other approaches rely on text or keyword analysis [83, 77, 81, 52], but in this work, we only consider network structure.

The most common approaches in previous graph-based similarity measures involve counting the number of neighbors two nodes have in common. Then, nodes are similar to the extent that they share common neighbors. In patent citation networks, the neighbor idea is adjusted to consider direct citations a patent receives. This most basic measure has the drawback that the nodes with large degree tend to be found more similar to other nodes than the lower degree nodes, because the higher degree nodes have the potential to have many neighbors in common with other nodes, even if a only

small fraction of their neighbors are in common. Salton proposed the Cosine similarity measure, which is widely used in citation networks [71]. This similarity measure regards the ith and jth rows of A as vectors and uses the cosine of the angle between them as their similarity score. In an undirected network, the number n_{ij} of common neighbors of nodes i and j is given by $\sum_k A_{ik}A_{jk}$, which is the (i,j)th element of $\mathbf{A}^2$. Suppose nodes i and j have degrees k_i and k_j respectively. The cosign similarity of i and j is the *number of common neighbors of the two nodes* divided by the *geometric mean of their degrees*, and is given by [59]:

$$\sigma_{ij} = \cos(\theta) = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\sum_k A_{ik}A_{jk}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{jk}^2}} = \frac{n_{ij}}{\sqrt{k_i k_j}},$$

where $0 \leq \sigma_{ij} \leq 1$, and $\sigma_{ij} = 1$ means that two nodes have exactly the same set of neighbors. $\sigma_{ij} = 0$ means that they have none of the same neighbors in common.

Another common neighbor-based similarity measure is the Pearson coefficient [59]. Pearson coefficients are used to identify when nodes are similar or dissimilar, compared with the expected number of common neighbors in the network, if neighbor connections were made at random. Suppose vertices i and j have degrees k_i and k_j respectively. We then consider the number of common neighbors we *expect* vertices i and j to have. In a network with N nodes, the probability of connecting to any other node is $\frac{1}{N}$, if chosen uniformly at random (neglecting the possibility of choosing the same node twice and choosing itself). Assume node j chooses k_j neighbors at random; node i then has $\frac{k_j}{N}$ probability of choosing a same neighbor that node j chose, and so on for each succeeding choice. Total expected number of common neighbors between the two vertices is $\frac{k_i k_j}{N}$. Non-normalized Pearson coefficients are given by $r^*_{ij} = \sum_k A_{ik}A_{jk} - \frac{k_i k_j}{N}$. Normalized Pearson coefficients are given by [59]:

$$r_{ij} \quad = \quad \frac{\mathrm{cov}(A_i, A_j)}{\sigma_i \sigma_j},$$

where $-1 \leq r_{ij} \leq 1$, and $\sigma_i\sigma_j$ is the maximum value of the covariance of any two sets of quantities.

The Jaccard index can also be used as a neighbor-based similarity measure between patents in a patent citation network. In particular, a relative co-citation of two patents i and j can be computed for a similarity score. The Jaccard index of the sets $C(i)$ and $C(j)$, where $C(i)$ denotes the set of all patents that cite i. The measure uses the cardinality of the intersection of nodes that directly cite both nodes i and j divided by cardinality of the union of nodes that cite i and j, and is given by:

$$sim_{Jaccard}(i,j) = \frac{|C(i) \cap C(j)|}{|C(i) \cup C(j)|}.$$

A graph lattice property is used to extend the Jaccard index in [20]. We note that the main difference in these measures is the normalization method used. In our proposed approach we normalize based on the total number of citations received by each node. We also propose using multiple stages of co-citations, not just direct neighbors for the similarity calculation.

In addition to the general graph-based similarity measures mentioned above, node similarity measures for specific applications have been developed. A similarity measure for the classification of texts, based on textual structure and semantics for natural language processing applications, is presented in [3]. The textual structure is evaluated using existing node similarity measures, such as Cosine similarity and Pearson coefficients. A similarity measure based on random walks on directed acyclic graphs is presented in [28]. The similarity measure is motivated by the potential need for literature recommendations for individuals who are searching for relevant literature in their topic of study. An application of similarity measures to resolve ambiguities of names of authors in scientific papers is presented in [4]. In this work, neighbor-based metrics are used to distinguish between authors represented by the same alias in collaborative networks. A similarity measure for the purpose of link prediction in both unweighted and weighted networks is proposed in [49]. The proposed similarity index combines a resource allocation index and a local path index, but the method neglects a key characteristics of citation networks – link direction. To the best of our knowledge, no research has been done on multi-stage indirect co-citation including normalization for the total

number of individual citation each patent has received in a PCN.

Table 3.1: A sample of current U.S. patent classes

Class	Description
2	Apparel
4	Baths, closets, sinks, and spittoons
5	Beds
7	Compound tools
8	Bleaching and dyeing: fluid treatment and chemical modification of textiles and fibers
12	Boot and shoe making
...	...
379	Telephonic communications
380	Cryptography
381	Electrical audio signal processing systems and devices
382	Image analysis
...	...
706	Data processing - artificial intelligence
707	Data processing: database, data mining, and file management or data structures
708	Electrical computers: arithmetic processing and calculating
709	Electrical computers and digital processing systems: multicomputer data transferring
...	...

3.2.2 Classification codes for U.S. patents

This section provides some background on the classification system for new U.S. patents. U.S. patents are manually classified by the United States Patent and Trademark Office (USPTO) into a scheme of about 400 classes and about 135,000 subclasses [46]. Table 3.1 provides a sample of patent classes and their descriptions. The classes and subclasses form a classification hierarchy, with possible subclasses of subclasses. The classification tree can go as deep as 15 levels, but varies greatly from patent to patent. Many domains have three or four levels of subclasses. In some domains, there is only one level of subclasses below a class. When applying our similarity measure developed in this work, we expect that the similarity between two patents containing the same classification codes to be higher than two patents that contain different classification codes. For example, let us consider three patents – patent x, patent y, and patent z. If patent x and patent y have 4 out of 5 classification codes in common, while patent x and patent z have 2 of 5 classification codes in common, then we expect that

using our co-citation method (which does not rely on classification codes), we would find patents x and y to be more similar than patents x and z. In this way, we use the class codes as an independent test of similarity. Using classification codes to compare patent relatedness and validate patent similarity measures are approaches that have been used in the past [11, 81].

As mentioned, in addition to classes, there are also subclasses for the classification of patents. For our validation, we use subclasses since subclasses capture with more detail the patents contents. Table 3.2 shows selected U.S. class codes for selected patents. When a non-zero value appears in the table for some patent and class code pair, that value represents the total number of subclasses within the class for that patent. For example, Patent US-5920861 has one subclass within class 375 and three subclasses within class 707. Table 3.5 contains detailed class and subclass code information for two patents.

Table 3.2: Number of subclasses within U.S. class codes that are associated with the selected patents

Patent ID	U.S. class codes										
	342	348	375	380	386	704	705	707	708	709	713
US-5920861	0	0	1	0	0	0	0	3	0	0	0
US-5917912	0	4	3	0	0	0	1	0	0	0	2
US-6138119	0	0	1	0	0	0	0	3	0	0	0
US-5930767	0	0	0	0	0	0	3	0	0	0	0
US-6363209	0	0	0	0	4	0	0	0	0	0	0
US-6237786	0	4	3	1	0	0	2	0	0	0	0
US-6240185	0	0	0	6	0	0	6	0	0	0	3
US-6499059	0	0	0	0	0	0	0	1	0	4	0
US-6292569	0	0	0	3	0	0	0	0	0	0	5
US-6658432	0	0	0	0	0	0	0	4	0	0	0
US-6226618	0	0	0	6	0	0	5	0	0	0	0
US-6389402	0	4	2	1	0	0	6	0	0	0	0
US-6016476	0	0	0	0	0	0	6	0	0	0	1
US-6427140	0	4	3	0	0	0	2	0	0	0	1
US-6249252	4	0	0	0	0	0	0	0	0	0	0
US-6208745	0	0	5	0	0	0	0	0	0	0	1
US-6449367	0	0	0	6	0	0	0	0	0	0	3
US-6606596	0	0	0	0	0	3	0	0	0	1	0
US-6507817	0	0	0	0	0	5	0	0	0	0	0
US-6578000	0	0	0	0	0	5	0	0	0	0	0

3.3 Proposed multi-stage similarity measures

3.3.1 Multi-stage co-citation similarity measure

In this section we define multi-stage co-citation similarity measures for the directed patent citation network. An example of a patent citation network is presented in Figure 3.4. Again, the adjacency matrix of the given network will be denoted **A**.

Let $G := (V, E)$ be a citation network and let N be the total number of nodes or patents in the citation network. $C_0(x, x)$ gives the number of nodes directly citing a patent x. $C_0(x, y)$ represents the number of nodes directly citing *both* nodes x and y. That is, citing both nodes x and y at stage 0 (total direct co-citations), and is given by the (x, y)th element of $\mathbf{A}\mathbf{A}^T$. That is, $C_0(x, y)$ represents the number of unique length-1 path pairs from both nodes x and y to a single node at level 0. $\{C_0(x, y)\}$ represents the set of nodes citing both x and y at stage 0. In our example citation network in Figure 3.2, $\{C_0(14, 15)\} = \{17, 18, 19, 20\}$. In order to define the multi-stage co-citation similarity measure, we introduce the concepts of the level-r citations for a node and the level-r co-citations for two nodes below.

Definition 1

Let $C_r(i, i)$ be the level-r citations for node i. That is, $C_r(i, i)$ is the number of citations that patent i receives by way of r intermediate patents. $C_r(i, i)$ is given by:

$$C_r(i, i) \quad = \quad \sum_{k=1}^{N} A_{ik}^{r+1}.$$

Definition 2

Let $C_r(i, j)$ be the level-r co-citations for patents i and j. That is, $C_r(i, j)$ is the number of co-citations that patents i and j receive by way of r intermediate nodes. The number of level-r co-citations is given by:

$$C_r(i, j) \quad = \quad \sum_{k=1}^{N} A_{ik}^{r+1} A_{jk}^{r+1}.$$

To illustrate the first definition, consider the following example. If one patent is cited directly by another patent, then there are no intermediate nodes, thus that is a

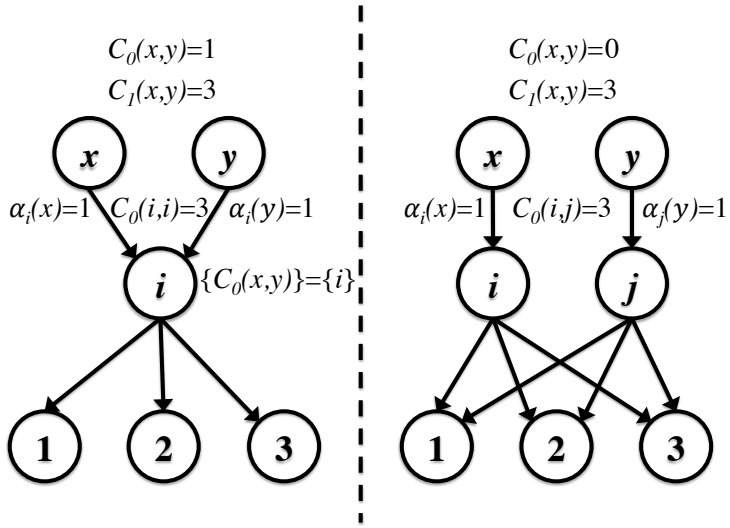

Figure 3.4: Level-1 co-citations in example patent citation networks to show $C_0(x, y)$ and $C_1(x, y)$ co-citation for node pair (x, y) in two cases: case 1: $i = j$ (left) and case 2: $i \neq j$ (right)

level-0 citation. To illustrate the second definition, consider the following example. If there is a directed path of length $r + 1$ from patent x to the patent v, and a directed path of length $r + 1$ from patent y to patent v, then the patents x and y are co-cited by patent v at level-r. If patent i is the same as patent j, then Definition 2 reduces to Definition 1.

Let $C_1(x, y)$ represent the number of indirect citations citing both patents x and y at level (or stage) 1. That is, $C_1(x, y)$ represents the number of unique length-2 path pairs from both x and y to individual nodes at level-1. $\{C_1(x, y)\}$ represents the set of indirect citations citing both x and y at stage 1, *i.e.,* represents the set of unique length-2 path pairs from both x and y to individual nodes at level-1.

Our formulation for $C_1(x, y)$, unique length-2 co-citations of nodes x and y, can be represented as follows:

$$C_1(x, y) \quad = \quad \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(x) \alpha_j(y) C_0(i, j),$$

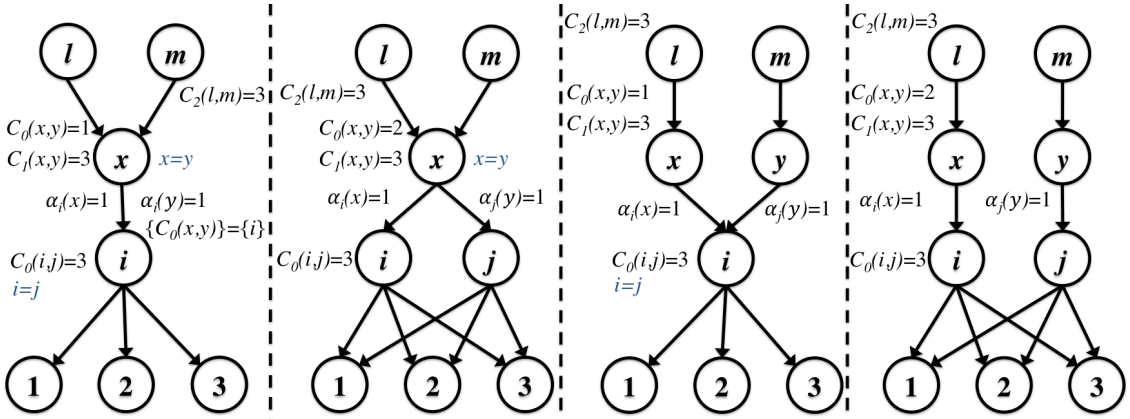

Figure 3.5: Four possible level-2 co-citations for node pair (l, m)

$$\begin{aligned} \text{where } \alpha_i(x) &= 1, \text{ if patent } i \text{ cites patent } x, \text{and } \alpha_i(x) = 0 \text{ otherwise,} \\ \text{and } \alpha_j(y) &= 1, \text{ if patent } j \text{ cites patent } y, \text{and } \alpha_j(y) = 0 \text{ otherwise.} \end{aligned}$$

$C_1(x, y)$ can be decomposed as follows:

$$C_1(x,y) = \begin{cases} \sum_{i \in S_1} \alpha_i(x)\alpha_i(y)C_0(i,i) + \sum_{i,j \in S_2} \alpha_i(x)\alpha_j(y)C_0(i,j), & \text{if } x \neq y \\ \sum_{i \in S_1} \alpha_i(x)\alpha_i(y)C_0(i,i) + \frac{1}{2}\sum_{i,j \in S_2} \alpha_i(x)\alpha_j(y)C_0(i,j), & \text{if } x = y, \end{cases} \tag{3.1}$$

where

$$\begin{aligned} S_1 &= \{i \in V | (x,i),\ (y,i) \in E\} \text{ is the set of all } i \text{ that cite both } x \text{ and } y, \\ S_2 &= \{i,j \in V | (x,i),\ (y,j) \in E,\ i \neq j\} \text{ is the set of all } i,j \text{ that cite both } x \text{ and } y. \end{aligned}$$

In Equation 3.1, $C_1(x, y)$ is the sum of the direct citations of the individual patents that **co-cite** x and y, plus the sum of the **direct co-citations** of the patents in which one node cites x and one node cites y. Figure 3.4 shows two possible level-1 co-citations for nodes x and y. For example, in our citation network in Figure 3.2, let nodes x and y be Nodes 14 and 15, respectively. Then $\{C_1(14, 15)\} = \{22, 23, 24, 25\}$.

Figure 3.4 shows two possible level-1 co-citations for patents x and y. Two patents may be very similar based on the co-citations received in the future, **but not directly co-cited**, as seen in the right had side of Figure 3.4 above. **If we use existing neighbor-based approaches, the lack of the direct co-citation will mean that the patents have a similarly score of zero, since they are not directly co-cited.** Using our proposed approach, those two patents can have a similarity score greater than zero, and indeed may be found to be very similar despite the lack of any direct co-citation. In Figure 3.4, $C_1(x, y) = 3$ for both cases since in our approach, x and y are co-cited by 3 nodes, when considering the one level of intermediate nodes, i and j. On the left hand side of Figure 3.4, x and y are co-cited at level 0, so $C_0(x, y)$ is greater for the left hand side network, than it is for the right hand side network. The ability to capture co-citations at different levels is the key contribution of this work.

Taking this idea further, we can increase the stage of indirect co-citation to gain more information. If we consider the level-2, then:

$$C_2(x,y) = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i(x)\alpha_j(y)C_1(i,j) \tag{3.2}$$

$$= \sum_{i\in S_1}\alpha_i(x)\alpha_i(y)C_1(i,i) + \sum_{i,j\in S_2}\alpha_i(x)\alpha_j(y)C_1(i,j), \tag{3.3}$$

where $C_2(x, y)$ is the number of indirect citations citing both x and y at level-2, citation path of length 3. In Equation 3.2, $C_2(x, y)$ is the sum of the level-2 indirect citations of the individual patents that **co-cite** x and y, plus the sum of the **indirect co-citations** of the patents in which one node cites x and one node cites y at level-1. Figure 3.5 shows four possible level-2 co-citations for two nodes l and m. Again, we demonstrate the ability to capture co-citations of various configurations at different levels using our approach. For example, nodes l and m may or may not be directly co-cited at level-0. Then, those node(s) that cite nodes l and m at level-0 may or may not be directly co-cited themselves, resulting in four combinations to consider at level-2. Our proposed approach introduces the level-r co-citation, which allows for node pairs to have a co-citation similarity score at each possible level of the citation network structure.

To gain the most information from the patent citation network, we need to take into account all of the direct and indirect citations of patents x and y. To take these

citations into account, we propose the following **multi-stage co-citation similarity measure**:

$$C_T(x,y) = \sum_{r=0}^{M} C_r(x,y),$$

where

$$\begin{aligned} C_r(x,y) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i(x)\alpha_j(y)C_{r-1}(i,j), \\ &= \sum_{i\in S_1} \alpha_i(x)\alpha_i(y)C_{r-1}(i,i) + \sum_{i,j\in S_2} \alpha_i(x)\alpha_j(y)C_{r-1}(i,j), \end{aligned}$$

for $r \geq 1$, and where M is such that $C_m(x,y) = 0$, for all $m > M$.

$C_T(x,y)$ is the sum of all the direct and indirect citations citing both patent x and y. That is, the sum of all the direct and indirect co-citations of a pair of patents. One of the drawbacks of $C_T(x,y)$ is that all levels of co-citations in the citation network have the same weight. To overcome this drawback we present **weighted multi-stage co-citation similarity measure** at level M as:

$$C^M(x,y) = \sum_{r=0}^{M} w_r C_r(x,y),$$

where $w_r = \alpha^{r+1}$, and $0 < \alpha \leq 1$. The result is that the closer a co-citation is to the patent pair in question, the greater weight it receives, with direct co-citations having the greatest weight.

3.3.2 Multi-stage bibliographic coupling similarity measure

Let $G := (V,E)$ be a patent citation network and let N be the total number of nodes or patents in the citation network. $B_0(x,y)$ represents the number of patents cited by both x and y at stage 0. That is, $B_0(x,y)$ represents the number of unique length-1 path pairs from a single node at level 0 to both x and y. $\{B_0(x,y)\}$ represents the set of patents cited by both x and y at stage 0. See Figure 3.6 for an example of nodes x and y bibliographically coupled. Additionally, the special case $B_0(x,x)$ gives the number of patents directly cited by a patent x.

$B_1(x,y)$ then represents the number of patents indirectly cited by both patents x

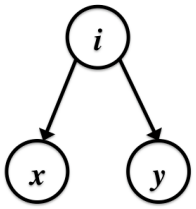

Figure 3.6: Example of simple bibliographic coupling of nodes x and y at level 0, indicating relatedness of patents x and y

and y at stage 1. That is, $B_1(x, y)$ represents the number of unique length-2 path pairs from a single node at level 1 to both nodes x and y. $\{B_1(x, y)\}$ represents the set of patents indirectly cited by patents x and y at stage 1. That is, $\{B_1(x, y)\}$ represents the set of unique length-2 path pairs from a single node at level 1 to both x and y. $B_1(x, x)$ gives the number of indirect citations cited by a patent x.

Our formulation for $B_1(x, y)$, indirect bibliographic coupling, can be represented as follows:

$$B_1(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_i(x)\delta_j(y)B_0(i, j),$$

where

$$\begin{aligned} \delta_i(x) &= 1, \text{ if patent } i \text{ is cited by patent } x,\ \delta_i(x) = 0 \text{ otherwise.} \\ \delta_j(y) &= 1, \text{ if patent } j \text{ is cited by patent } y,\ \delta_j(y) = 0 \text{ otherwise.} \end{aligned}$$

$B_1(x, y)$ can be decomposed as follows:

$$B_1(x, y) = \begin{cases} \sum\limits_{i \in S_1} \delta_i(x)\delta_i(y)B_0(i, i) + \sum\limits_{i,j \in S_2} \delta_i(x)\delta_j(y)B_0(i, j), \text{if } x \neq y \\ \sum\limits_{i \in S_1} \delta_i(x)\delta_i(y)B_0(i, i) + \frac{1}{2} \sum\limits_{i,j \in S_2} \delta_i(x)\delta_j(y)B_0(i, j), \text{if } x = y, \end{cases} \tag{3.4}$$

where

$$\begin{aligned} S_1 &= \{i \in V | (i, x),\ (i, y) \in E\} \text{ is set of all } i \text{ that are cited by both } x \text{ and } y \\ S_2 &= \{i, j \in V | (i, x),\ (j, y) \in E,\ i \neq j\} \text{ is set of all } i \text{ and } j \text{ that are cited by } x \text{ and } y. \end{aligned}$$

In Equation 3.4, $B_1(x, y)$ is the sum of patents that are directly cited by patents

in the set $\{B_0(x,y)\}$ plus the sum of the directly **bibliographic coupled** patents in which one patent is cited by x and one patent is cited by y.

Still, we can increase the stage of indirect bibliographic coupling to gain more patent similarity information. If we consider the stage 2 (citation length of 3 hops), then:

$$B_2(x,y) = \sum_{i=1}^{N}\sum_{j=1}^{N}\delta_i(x)\delta_j(y)B_1(i,j) \tag{3.5}$$

$$= \sum_{i\in S_1}\delta_i(x)\delta_i(y)C_1(i,i) + \sum_{i,j\in S_2}\delta_i(x)\delta_j(y)B_1(i,j), \tag{3.6}$$

where $B_2(x,y)$ is the number of patents indirectly cited by both patents x and y at level 2. In Equation 3.5, $B_2(x,y)$ is the sum of the indirect citations of the patents in the set of $B_0(x,y)$ at level 2 plus the sum of the **indirect bibliographic coupling** of the patents in which one of them is cited by x and one of them is cited by y at level 2.

To gain more information still from the patent citation network, we can take into account all of the direct and indirect citations in common by patent x and y. To take these citations into account, we propose the following **multi-stage bibliographic coupling**:

$$B_T(x,y) = \sum_{r=0}^{M} B_r(x,y),$$

where

$$B_r(x,y) = \sum_{i=1}^{N}\sum_{j=1}^{N}\delta_i(x)\delta_j(y)B_{r-1}(i,j)$$

$$= \sum_{i\in S_1}\delta_i(x)\delta_i(y)B_{r-1}(i,i) + \sum_{i,j\in S_2}\delta_i(x)\delta_j(y)B_{r-1}(i,j),$$

for $r \geq 1$, and where M is such that $B_m(x,y) = 0$, for all $m > M$.

$B_T(x,y)$ is the sum of all the common direct and indirect citations made by both patents x and y. One of the drawbacks of $B_T(x,y)$ is that it does not consider the position of the citation in the citation network. To overcome this drawback, we present **weighted multi-stage bibliographic coupling** at level M as:

$$B^M(x,y) = \sum_{j=0}^{M} w_r B_r(x,y),$$

where $w_r = \alpha^{r+1}$, and $0 < \alpha \leq 1$. Again, it is observed that nodes that are closer to the patents x and y in the citation network will have a greater weight than those that are farther. Patents that are directly cited by x and y will have the greatest weight.

3.3.3 Normalized multi-stage co-citation and bibliographic coupling similarity measures

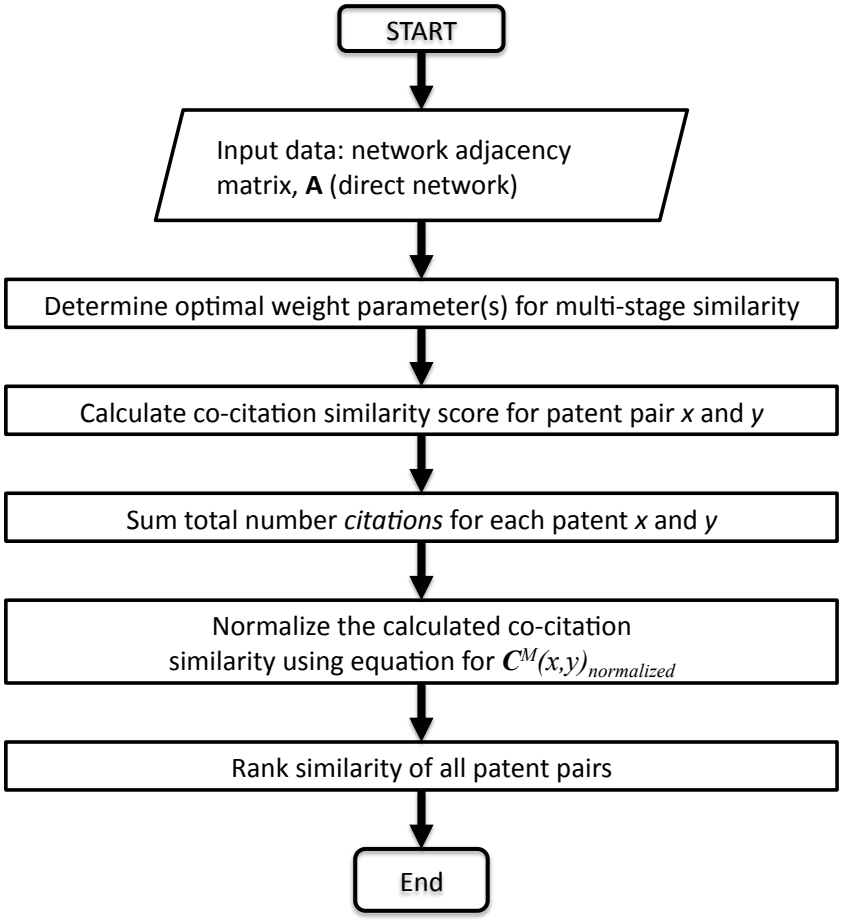

Figure 3.7: Flowchart showing co-citation similarity calculation with normalization

As we have seen in the previous sections, co-citation considers how patents are similar based on how *future* patents cite them. Investigating longer co-citation chains, and getting more information from the historical citation has its advantages. **The challenge is when some patents have a large number of citations, they**

may be considered similar to many other patents in the patent citation network, merely because they are highly cited. This is not always a detriment, but experimentation has shown that similarity performance can suffer because of the large number of citations, both direct and indirect, that a patent has. To overcome this drawback, we propose the new idea of leveraging the overall citation information for each patent in the patent citation network. See Figure 3.7 for a flowchart of this solution.

A stage-wise normalization would normalize the score contribution at each stage of the multi-stage co-citation. When computing the total citations over all levels, we weigh the number of citations at each level by the coefficient α^{r+1} weighting scheme, where r is the level. The normalized multi-stage co-citation similarity measure is given by:

$$C^M(x,y)_{normalized} = \frac{C^M(x,y)}{\mathrm{C}^\infty(x,x) + \mathrm{C}^\infty(y,y)},$$

where $C^\infty(x,x)$ and $C^\infty(y,y)$ are the weighted sums of all direct and indirect *citations of* patents x and y, respectively, over all M stages, and $C^\infty(i,i) = \sum_{r=0}^{M}\sum_{k=1}^{N}\alpha^{r+1}A_{ik}^{r+1}$, so that direct citations have the greatest weight and weight decreases as the indirect citation length increases. Normalized multi-stage bibliographic coupling similarity measure follows the same formulation, and is given by:

$$B^M(x,y)_{normalized} = \frac{B^M(x,y)}{\mathrm{B}^\infty(x,x) + \mathrm{B}^\infty(y,y)},$$

where $B^\infty(x,x)$ and $B^\infty(y,y)$ are the weighted sums of all *citations made by* patents x and y, respectively, over all M stages. Similar to the co-citation case, $B^\infty(i,i) = \sum_{r=0}^{M}\sum_{k=1}^{N}\alpha^{r+1}A_{ki}^{r+1}$.

When applying the co-citation similarity measure idea, patents that are cited together are considered similar. Our normalized similarity measures help to avoid skewing results such that highly cited patents are determined to be similar to each other merely because they both have many citations. A relatively small α values suggest that the direct and closer indirect citations are best for capturing patent similarity. Based on extensive experiments, we recommend that for multi-stage co-citation, without normalization, we use $\alpha = 0.01$ and for multi-stage co-citation, with normalization, we use

$\alpha = 0.1$. Based on extensive experiment, the following table provides guidelines for the case of multi-stage co-citation and bibliographic coupling.

Table 3.3: Guidelines for parameters of proposed co-citation and bibliographic coupling similarity measures

Method	α
Multi-stage co-citation, without normalization	0.01
Multi-stage co-citation, with normalization	0.1
Multi-stage bibliographic coupling, without normalization	10.0
Multi-stage bibliographic coupling, with normalization	0.7

3.4 Experimental results

In this section, we use the U.S. class codes as an independent test of similarity. Using classification codes to compare patent relatedness and to validate patent similarity measures are approaches that have been used in the past [11, 81]. In particular, the patent classification system is used for validation in [81]. The idea is that similarity between two patents belonging to the same patent category should be higher than two patents from different categories. We follow this validation idea in this work.

3.4.1 Data description

The dataset actually used for the experiments are U.S. patents in the area of information and security issued between 1994 and 2007 [78]. For these experiments, we take the top 1% most frequently cited patents from 1994 to 2007 as our nodes in the patent citation network. In order to have a single connected tree structure to which to apply similarity measures, we select the patents that cite, either directly or indirectly, the most cited patent from the original dataset, which is patent US-5349655. Our patent citation network then consist of 4,241 nodes and 18,385 edges.

3.4.2 Parameter optimization for multi-stage co-citation

Experimentation with co-citation similarity measure shows performance improves when we apply both of our proposed approaches: multi-stage co-citation and normalized co-citation (direct and multi-stage). See Figure 3.8. Consider direct co-citation, not normalized as the baseline. When we introduce normalization to the direct co-citation approach, by considering the total times the pair of patents is cited, we achieve an improvement over the baseline. Through experimentation, for the case of normalized multi-stage co-citation, we find that $\alpha = 0.1$ performs the best, achieving a Spearman rank correlation coefficient value of 0.4, thus we recommend this as the parameter value for normalized multi-stage co-citation. The better performance of $\alpha = 0.1$ over $\alpha < 0.1$ indicates that for normalized multi-stage co-citation, we should consider the indirect co-citations, and not merely consider the direct co-citations. The better performance of $\alpha = 0.1$ over $\alpha > 0.1$ indicates that for normalized multi-stage co-citation, much of the emphasis should be on direct and lower level co-citations.

3.4.3 Parameter optimization for multi-stage bibliographic coupling

Multi-stage bliographic coupling similarity measure performance improves when we increase the weight parameter to about $\alpha = 10$. After this α value, we observe very small improvement in the correlation coefficient for greater α values. An α value greater than 1 would give a greater weight (in the overall summation) to bibliographic coupling that occurs at indirect (higher up) levels, compared to the weight given at the direct level. See plotted red line in Figure 3.9. The interpretation of this result is that it is important for two nodes to be bibliographicly coupled at an higher indirect stage, in addition to just at a direct stage, for them to be considered similar, using our approach. It would require considering a longer citation history for two nodes to be similar, not merely the direct stage, or a very small weighted indirect stage. Rather, we must consider higher up indirect stages also, when calculating similarity of two patents.

Table 3.4: Pair-wise patent similarity scores using proposed normalized multi-stage co-citation similarity measure (CC score) and existing Jaccard similarity index for U.S. class codes of patents (US- prefix omitted in patent number)

Rank	Node pair	CC score	Jaccard index
1	5745604 , 5943422	1.0000	0.1818
2	5745604 , 5832119	0.7259	0.5556
3	6567796 , 6658432	0.7143	0.1429
4	6338070 , 6499059	0.6981	0.0769
5	6301590 , 6507817	0.6961	0.1429
6	6567796 , 6587547	0.6804	0.1250
7	6587547 , 6658093	0.6689	0.1250
8	6578000 , 6606596	0.6160	0.1000
9	5745604 , 6185683	0.6078	0.0714
10	6507817 , 6578000	0.6046	0.2222
11	5745604 , 5862260	0.5999	0.5714
12	5745604 , 6157721	0.5587	0.0667
13	6064764 , 6246777	0.5514	0.5000
14	5822436 , 5862260	0.4779	0.1000
15	6246777 , 6332031	0.4737	0.3333
16	5745604 , 6064764	0.4588	0.1429
17	5862260 , 6122403	0.4545	0.5000
18	5862260 , 6052486	0.4485	0.1667
19	6064764 , 6332031	0.4466	0.2500
20	6064764 , 6275599	0.4098	0.5000
21	5943422 , 6185683	0.4020	0.0667
22	5745604 , 5822436	0.3930	0.2000
23	5915019 , 5920861	0.3886	0.0667
24	5943422 , 6157721	0.3856	0.2143
25	5765030 , 5826013	0.3851	0.2857
26	5910987 , 5920861	0.3803	0.0769
27	6246777 , 6275599	0.3780	1.0000
28	5910987 , 5915019	0.3773	0.5385
29	5915019 , 5943422	0.3765	0.0588
30	6157721 , 6185683	0.3731	0.0556
31	5832119 , 5862260	0.3724	0.6250
32	5920861 , 6185683	0.3720	0.0769
33	5915019 , 5917912	0.3698	0.4667
34	5915019 , 5949876	0.3641	0.5714
35	6138119 , 6185683	0.3632	0.0769
36	5745604 , 6240185	0.3603	0.0870
37	5910987 , 5917912	0.3591	0.5385
38	5910987 , 5949876	0.3531	0.5385
39	6122403 , 6311214	0.3523	0.4286
40	5920861 , 6138119	0.3511	1.0000
41	5917912 5920861	0.3511	0.0667
42	6064764 6243480	0.3503	0.5000
43	5915019 6138119	0.3489	0.0667
44	5915019 6185683	0.3459	0.4286
45	5745604 6292569	0.3411	0.0667

Table 3.5: Detailed information on the US-6240185, US-6389402 patent pair

	US-6240185	US-6389402
Title	Steganographic techniques for securely delivering electronic digital rights management control information over insecure communication channels	Systems and methods for secure transaction management and electronic rights protection
Issue date	May 29, 2001	May 14, 2002
Class codes	380/232 ; 380/205; 380/210; 380/221; 380/227; 380/231; 705/51; 705/52; 705/54; 705/55; 705/59; 705/76; 713/176; 713/189; 713/193; 726/21; G9B/20.002; G9B/27.01; G9B/27.05	705/51; 348/E5.006;348/E5.008; 348/E7.06; 348/E7.07; 375/E7.009; 375/E7.024; 380/201; 705/1.1; 705/37; 705/53; 705/57; 705/80

3.4.4 Validation of similarity scores

To validate results obtained by applying our proposed similarity measure that is based on the patent citation network, we compute the well-known Jaccard similarity coefficient for the set of the top 100 ranked patents, and compare them to our developed approach. The top 100 patents are determined based on the centrality (importance) measure developed earlier in work [68, 67]. Table 3.4 shows the similarity score for the pairs of patents (separated by a comma) using two different methods. The scores are ordered, or ranked, such that the most similar pairs of patents are at the top of the table for the proposed co-citation approach. The Jaccard similarity coefficient is given in the fourth column for comparison. The Jaccard similarity coefficient is defined as the size of the intersection divided by the size of the union of the sample sets [76]:

$$J(A,B) \quad = \quad \frac{|A \cap B|}{|A \cup B|}, \tag{3.7}$$

where $|A \cap B|$ is the cardinality of the intersection of subclass codes for Patents A and B, and $|A \cup B|$ is the cardinality of the union of subclass codes for Patents A and B. For example, if patent A has subclass codes 1, 2, and 3, and patent B has subclass codes 1, 3, 4 and 7, then $|A \cap B| = 2$ and $|A \cup B| = 5$, and we have $J(A,B) = 2/5$.

Table 3.4 shows pair-wise node similarity scores using the Jaccard index method and the proposed co-citation similarity measure.

To validate using Jaccard similarity coefficient, we use the set of "Current U.S. Class" codes for each patent. When a patent is created, it is associated with class and subclass codes describing the nature of the work. Codes for all U.S. patents can be found on the UP Patent and Trademake Office website [78]. In addition to U.S. Class codes, there are other codes such as International codes that may be used. In this study, we use the classification codes titled "Current U.S. Class" and consider the subclass for the intersection and union counts for the Jaccard similarity index. Table 3.5 shows detailed class code information for two patents: US-6240185 and US-6389402. Notice the class/subclass hierarchy where the class is the number preceding the forward slash, and the subclass is the code following the forward slash.

Since our proposed measure considers the patent citation network structure, rather than the U.S. Class codes, our approach shows a different similarity, which focuses more on the network structure characteristics, rather than the category and subcategory of patents. In addition, our approach is not sensitive to the writing style of the author of the patent.

We compare the Spearman rank correlation coefficient, r, for co-citation methods:

1) Single stage co-citation, without normalization (baseline)

2) Multi-stage co-citation, without normalization

3) Single stage co-citation, with normalization

4) Multi-stage co-citation, with normalization after CC calculation

Similarly, we compare the Spearman rank correlation coefficient, r, for bibliography methods:

1) Single stage bibliographic coupling, with normalization

2) Single stage bibliographic coupling, without normalization

3) Multi-stage bibliographic coupling, with normalization after CC calculation

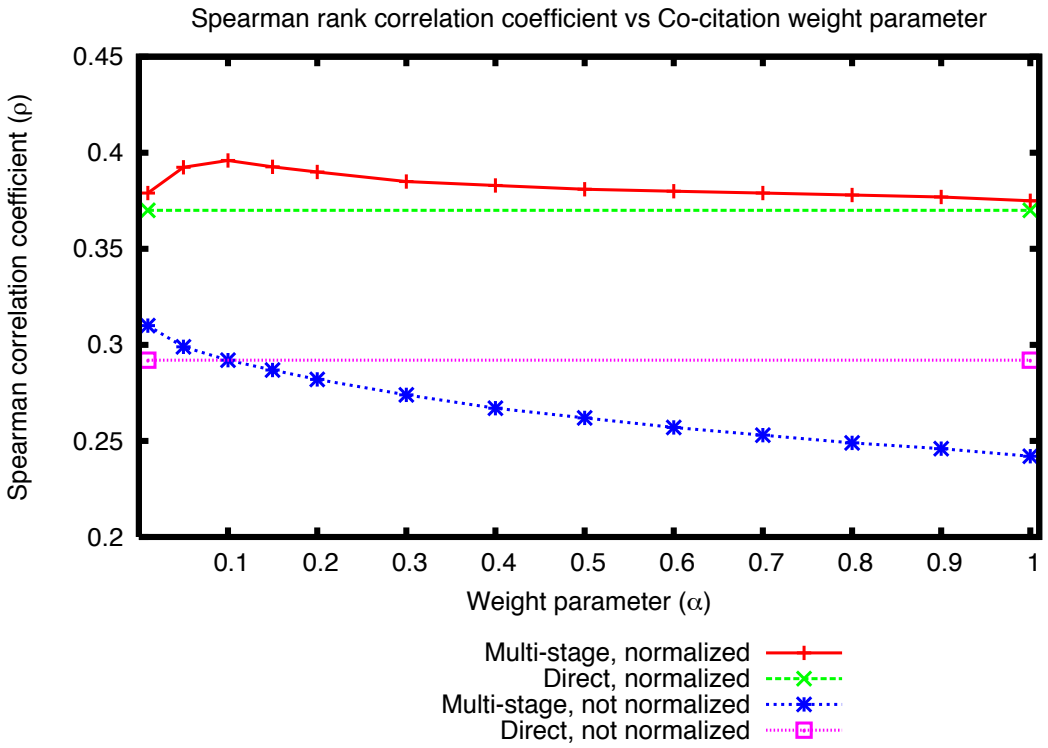

Figure 3.8: Parameter evaluation for multi-stage co-citation similarity, normalized and not normalized, using weighting coefficient α^{r+1}, where r is the stage. Spearman rank correlation performance when compared to Jaccard similarity between patent pairs using U.S. Class codes for 100 patents shows best performance when applying our approaches: normalized multi-stage co-citation.

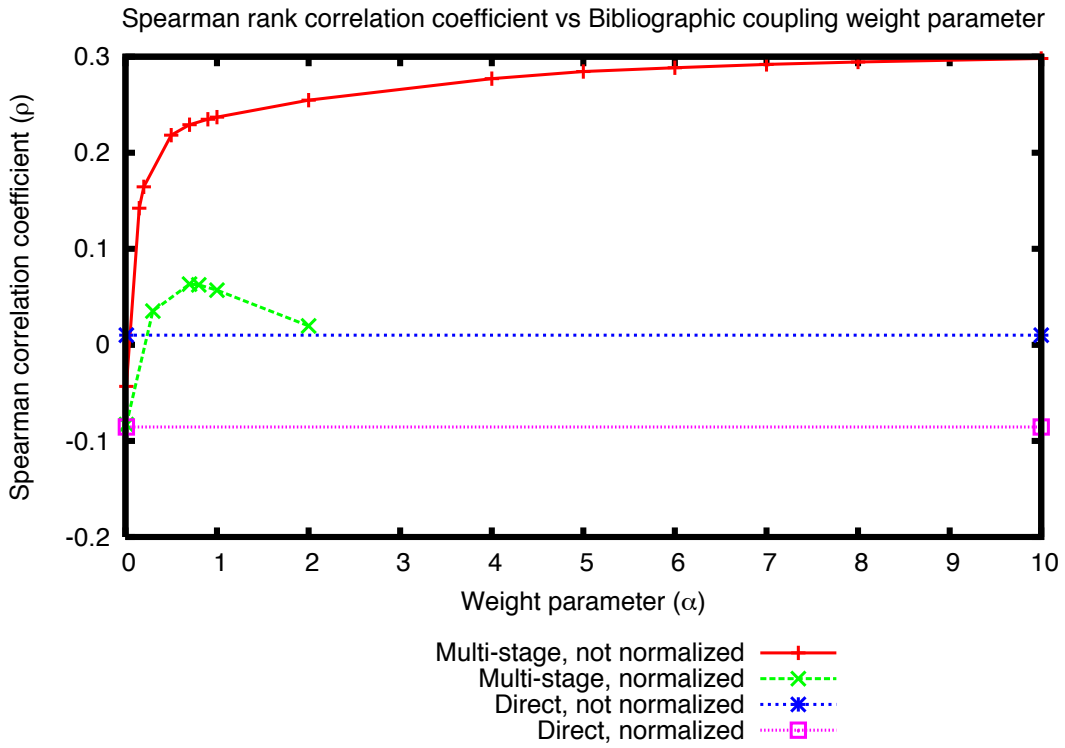

Figure 3.9: Parameter evaluation for multi-stage bibliographic coupling similarity, normalized and not normalized, using weighting coefficient α^{r+1}, where r is the stage. Spearman rank correlation performance when compared to Jaccard similarity between patent pairs using U.S. Class codes for 100 patents shows best performance when greater weight is given to the higher levels of bibliographic coupling.

4) Multi-stage bibliographic coupling, without normalization

The results for co-citation similarity and bibliographic coupling are plotted in Figures 3.8 and 3.9, respectively. Tables 3.7 and 3.8 show the corresponding best Spearman rank correlation coefficient performance over the parameter evaluation. For the co-citation similarity measures, we achieve the best results with normalized multi-stage. For bibliographic coupling similarity measures, we achieve the best results when we use multi-stage bibliographic coupling. In both cases, multi-stage approaches outperform direct approaches, validating that consideration of indirect co-citations and bibliographic coupling do assist in determining patent pair similarity. Normalization helps to improve results in the case of co-citation because the variance of the citations that a patent receives is greater than the variance of the citations makes. For example, consider the 100 most cited patents and the 100 patents that make the most citations from our dataset. The 100 most cited patents have a range of 712 citations and a variance of 16036.82, while the 100 patents that make the most citations have a range of 52 citations and a variance of 101.97. These statistics support our results wherein multi-stage co-citation benefits from normalization, while multi-stage bibliographic coupling does not.

Table 3.6: Spearman correlation performance of proposed co-citation similarity methods when compared to Jaccard similarity using U.S. Class codes for 100 U.S. patents

Similarity measure	r
Single stage co-citation, without normalization	0.29
Multi-stage co-citation, without normalization	0.31
Single stage co-citation, with normalization	0.37
Multi-stage co-citation, with normalization	**0.40**

3.5 Conclusion

The objective of this work was to develop a similarity measures for patents in complex patent citation networks. To this end, we introduce new similarity measures that uses direct and multi-stage co-citation, as well as normalization of the co-citation

Table 3.7: Improvement factor of Spearman correlation over baseline: performance of proposed co-citation similarity methods when compared to Jaccard similarity using U.S. class codes for 100 U.S. patents

Similarity measure	Improvement (%)
Single stage co-citation, without normalization	Baseline
Multi-stage co-citation, without normalization	6.8
Single stage co-citation, with normalization	27.5
Multi-stage co-citation, with normalization	**37.9**

Table 3.8: Spearman correlation performance of proposed bibliographic coupling similarity methods when compared to Jaccard similarity using U.S. Class codes for 100 U.S. patents

Similarity measure	r
Single stage bibliographic coupling, with normalization	-0.08
Single stage bibliographic coupling, without normalization	0.01
Multi-stage bibliographic coupling, with normalization	0.06
Multi-stage bibliographic coupling, without normalization	**0.30**

similarity score. The multi-stage co-citation provides more complete information from given patent citation network because it considers direct as well as indirect co-citations. We compared our similarity measure to one based on U.S. class codes using the Jaccard index. We achieved the best performance when we considered multi-stage co-citation and normalized with parameter $\alpha = 0.1$. The proposed similarity measure helps analysts determine patent similarity, which can be extended for the clustering of patents, the detection of outlier patents, and so on. Additionally, these methods may be applied to literature citation networks which have a structure similar to patent citation networks. For bibliographic coupling, we achieved the best results when looking further up the citation chain. The proposed similarity measure helps analysts determine patent similarity, which can be used for the clustering of patents and the detection of outlier patents, as well.

For future work, we plan to explore the idea of distinguishing the weights for the two co-citation cases shown in Figure 3.4. That is, we explore the effect of weighting

length two co-citation differently in the case that: (1) a single patent is the intermediate patent, or (2) two different patents are intermediate patents. Additional future work is to leverage the proposed similarity measures in order to identify outlier or anomaly patents. In calculating the similarity, we are able to calculate dissimilarity between patents. Finally, while integrating co-citation and bibliographic coupling similarity measures seems like a natural extension of this work, there are challenges to doing so. For example, a patent author can decide which prior patents to cite, but a patent author cannot decide what future patents will cite his patent. As a future work, we can study the development of a bibliographic coupling similarity measure and the integration of co-citation and bibliographic coupling approaches.

In the next chapter, we leverage the co-citation similarity result for the purpose of patent outlier ranking in patent citation networks.

Chapter 4

Patent Clustering and Outlier Ranking Methodologies for Attributed Patent Citation Networks

4.1 Introduction

In recent years, there has been an emphasis on analyzing data using graph theoretical methods [17, 79]. Graph-based data mining approaches attempt to analyze data that can be represented in a graph, consisting of nodes and edges. While there has been much work on graph-based data mining [16, 82, 31, 36], there is still much room for contribution in the area of graph-based outlier ranking and detection. Figure 4.1 is a representation of a real-life patent citation network, and an example of such graph data, where nodes represent individual patents and edges represent citations made by one patent to another.

Outlier detection, or anomaly detection, has to do with identifying entities that are unusual or that deviate from the rest of the dataset [7, 30]. This is an important research topic that has been researched within diverse areas and application domains [30, 48, 5, 66]. Many anomaly detection techniques have been specifically developed for certain application domains, while others are more generic [14]. Researchers have adopted concepts from diverse disciplines such as statistics, machine learning, data mining, information theory, spectral theory, and have applied them to specific problem formulations. The goal of graph outlier ranking is to score and rank objects to the degree that they differ from majority of dataset in the graph data. That is, node relationship data is analyzed to identify interesting or exceptional objects. From an abstract level, an anomaly or outlier is defined as a pattern or object that does not conform to expected normal behavior [14]. A straightforward anomaly detection approach, then, is to define a region or characteristic representing normal behavior and identify

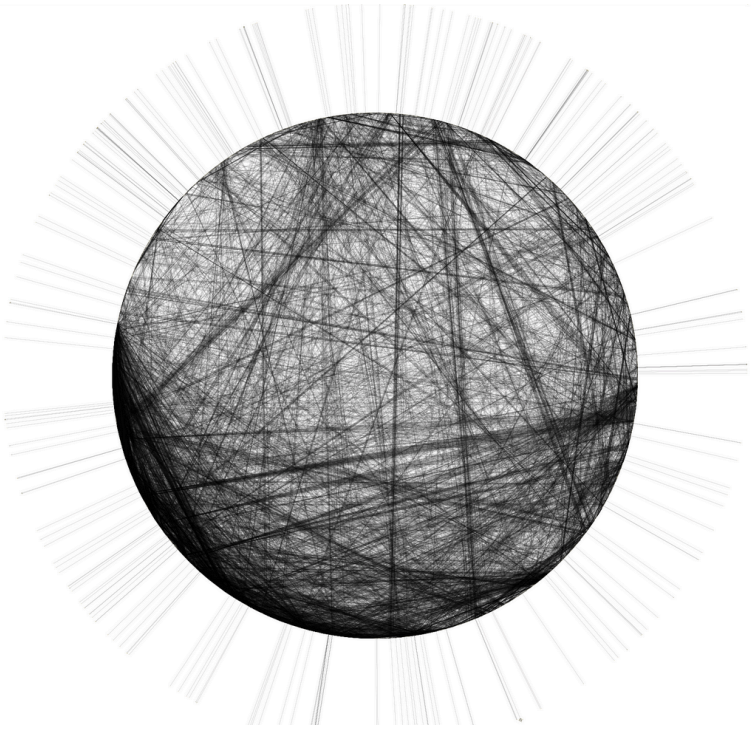

Figure 4.1: Representation of 4,241 patents in a real-life patent citation network

any observation in the data that does not belong to this normal region as an anomaly. Several factors make these seemingly simple tasks of identifying outliers and validating results very challenging. Firstly, often the input data contain noise that tends to be similar to the actual anomalies and is therefore difficult to distinguish and remove. For identifying outliers, defining a region that encompasses all possible normal behavior can be very difficult. In addition, the boundary between normal and anomalous behavior is often not well-defined. For this reason, an anomalous observation that are close to the boundary can actually be normal, and vice versa. For validation, the availability of labeled data for training/validation of models used by anomaly detection techniques is often a major challenge. In many domains normal behavior evolves and a current notion of normal behavior might not be sufficiently representative in the future. The exact notion of an anomaly is different for different application domains, and applying a technique developed in one domain to another is not always straightforward.

Applications of graph outlier ranking include credit card fraud detection, computer network security and intrusion detection, identifying exceptionally cross-disciplinary

authors in author-paper networks, virus detection in a computer network, detecting fraudulent financial transaction, detecting abusive users in a communication or online social network, and so on [15, 72]. For example, in a computer network, it is critical to track the spread of diseases in the form of viruses and worms spreading from host to host in a computer network [5]. Affected machines may exhibit slightly anomalous behavior, such as a loss of performance or violations of specific policies, which may be hard to detect on the basis of an individual machine, but may be apparent when the communication graph structure is analyzed.

With the expected increase in the number and complexity of patents, quickly analyzing patents to find outlier patents or groups of patents in a patent citation network (PCN) has become an advantage and provides business advantages [80, 47]. Most new patents are related to previous works in some way. This relatedness between patents is captured by a patent's citation of a previous work, and can be thought of as an extension of the previous work(s). Taken all together, patents and the citation links between them can be represented in a patent citation network. Individual patents also contain rich information that describe its characteristics. It is important to analyze the patent citation network to gain an understanding of past, current, and possible future technological trends [27]. Outlier detection has been used to identify new technological opportunities using semantic patent analysis [84]. In this proposed work, we develop outlier ranking methods that can be used to identify patents that differ from the majority of patents in the patent citation network. The identified outlier patents from this work serve as a starting point to evaluate possible new areas of technological opportunity.

In a patent citation network, an outlier node may correspond to a patent in the citation network that is different from the other patents in some way. For example, a patent itself may deal with an innovation that bridges multiple technology areas, such as automotive and metal alloys. Information between patents is represented by directed edges (*i.e.* arc) indicating a citation. An outlier citation corresponds to a patent citation that links two patents that we would not expect to be linked, given the rest of the patent citation network. For example, an outlier edge may be a medical patent that

Table 4.1: Sample attributes for U.S. patent data

Patent ID	U.S. patent attributes			
	Class codes	Year granted	Country	Times cited
5920861	375/E7.009; 707/999.004; 707/999.009; 707/999.102	1999	USA	550
6138119	375/E7.009; 707/999.004; 707/999.009; 707/999.102	2000	USA	411
6363209	386/252; 386/259; 386/329; 386/E5.004; G9B/20.002; G9B/27.033; G9B/27.05	2002	Japan	13
5930767	705/26.41; 705/27.2; 705/77	1999	USA	268
6606596	704/270 ; 704/231; 704/246; 704/251; 709/206; 715/752	2003	USA	93

cites a patent in the area of computer networking. Clusters of outlier nodes represent a subgraph of nodes that are different from the majority of the graph [82]. A cluster outlier corresponds to a subgraph that is different from the majority of the PCN, which indicates an unusual patent group relative to the surrounding patents in the PCN.

In general, outlier ranking in patent citation networks corresponds to identifying exceptional: (1) nodes, (2) edges, or (3) clusters (or subgraphs). Detection of an anomalous cluster of nodes in a network is a different and more recent research topic [61, 18, 5], and may be a sign of a technology shift. In this work we focus on ranking nodes in outlierness.

Patent datasets are rich with information that is carefully assembled. Additionally, patent citation network data contain both *graph structure data* and object *attribute data*. Graph structure data in the form of citations made and received are carefully considered by patent writers. Patent attributes such as classification codes are specifically assigned based on the nature of the technology. See Table 4.2 for a sample of patent attribute data. Traditional outlier ranking techniques typically focus on either homogeneous vector data or on graph structures. However, many recent complex applications contain both types of data: multi-dimensional numeric and/or categorical information and relations between objects in *attributed graphs*. An open challenge is how outlier ranking should handle these different data types in a unified or integrated fashion. **There is currently no work on patent citation network outlier detection that considers attributes in patent citation network, in addition to the citation network structure.** This work proposes new methods for the ranking of outlier patents within patent citation networks represented by attributed graphs.

In this work, we propose the following characteristics for outlier nodes in graphs. First, an outlier node is not highly clustered with other objects (based on node attributes and graph structure). That is, a node should be considered more regular, the more it can be clustered with other nodes in the network. For this reason, this work will also address the patent clustering problem. Second, an outlier node has low node centrality within the network. Since centrality is a measure of importance or cohesiveness of a node to the network which it belongs, a highly central node should not be considered an outlier node. Finally, low similarity to other nodes in the network, based on graph structure, is a characteristic of an outlier node. If a node is similar to other nodes in the network, then it should not be considered an outlier. We will use these three characteristics in our proposed outlier score function.

The remainder of the work is organized as follows. First, Section 4.2 provides background on outlier node detection in graph data. Next, Section 4.3 presents a new subspace clustering algorithm for patents in an attributed patent citation network. Section 4.4 presents our proposed node outlier scoring functions. Section 4.5 presents a data description for the artificial and real-life data used in experiments, along with experimental results for outlier ranking in patent citation networks data. Finally, Section 4.6 concludes this work and presents future research plans.

4.2 Background

In this section we provide a literature review of some existing outlier node ranking methods. We review both graph-based approaches, as well as integrated attribute and graph-based outlier ranking algorithms.

Anomaly detection methods can be classified based on the underlying design principles, such as those based on graph communities, graph compression, graph decomposition, distance metrics, and probabilistic modeling of graph features [66]. For an extensive survey of anomaly detection, see [14]. For an extensive survey on anomaly detection in time-evolving networks, covering anomalous nodes, edges, subgraphs, and events, see [66]. Anomaly detection techniques can then be further categorized based on the types of anomalies they detect and within the type of data they detect anomalies.

For example, identifying anomalies in weighted graphs was presented in [1]. Anomalies have also been researched in large graph datasets [48, 36], which is a data type that is becoming more common due to growing data collection and richness of data, and in bipartite graphs [75]. In the following sections, we present some of the most relevant existing methods for outlier node ranking.

4.2.1 Graph structure-based node outlier ranking methods

In 2007, Xu *et al.* present a method of network clustering, or graph partitioning, called Structural Clustering Algorithm for Networks (SCAN) in order to discover structures in networks [82]. In particular, this work attempts to distinguish roles of nodes (also called vertices) . In addition to partitioning the network, the method also identifies two types of nodes that have a unique role within the network: hubs, which bridge more than one cluster, and outliers, which are marginally connected to a single cluster. This approach uses only the network structure in order to detect the clusters, hubs, and outliers. Vertices are clustered base on the structural similarity of a neighborhood.

Network clustering or graph partitioning is the division of a graph into a set of subgraphs called clusters. Given a graph $G := (N, E)$, where N is a set of nodes and E is a set of edges between nodes, the goal of graph partitioning is to divide G into k disjoint subgraphs $G_i := (N_i, E_i)$, in which $N_i \cap N_j = \emptyset$ for any $i \neq j$, and $N = \sum_{i=1}^{k} N_i$.

SCAN is applied to simple undirected and unweighted graphs with the goal of clustering networks optimally and identifying hubs and outliers. In SCAN, the neighborhood around two connected vertices is considered. The neighborhood of a vertex are itself plus all of the vertices connected to it by an edge. When two nodes are considered together, their combined neighborhood reveals neighbors common to the two vertices. SCAN is based on common neighbors, and nodes are assigned to a cluster based on how they share neighbors. A social network is a good illustration for how and why SCAN works. Persons who share many friends create a community. The more friends they have in common, the tighter the community. In social networks also, there are persons who play roles. Some are friendly with numerous communities and bridge tie communities together: hubs. Some are not well connected to a group: outliers. A similar graph

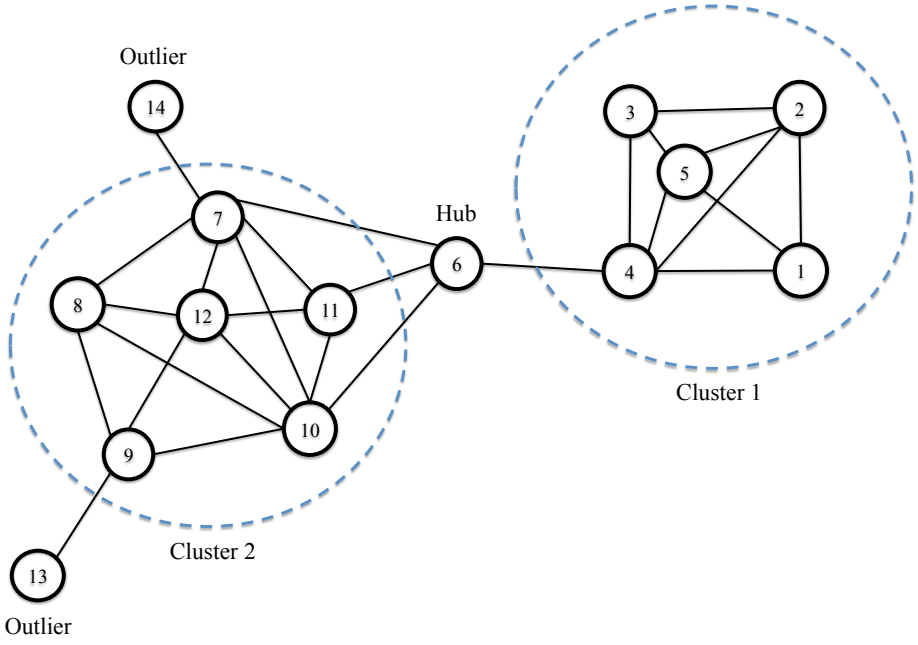

Figure 4.2: Sample SCAN result showing two clusters, one hub, and two outliers

partitioning methodology, for the purpose of outlier detection is presented, in [13].

Random walk methods have been used for for numerous information retrieval tasks, including keyword extraction, text summarization, and web searches [55]. These approaches represent data in a stochastic graph and perform a random walk along edges of the graph to determine the importance or *centrality* of individual nodes.

In 2006, Moonesinghe et al. present a stochastic graph-based algorithm, called OutRank for detecting outlying objects [55]. The main idea of this approach is to represent the underlying dataset as a weighted undirected graph, where each node represents an object and each weighted edge represents the similarity between objects. By transforming the edge weights into transition probabilities, the approach models the system as a Markov chain, and finds the dominant eigenvector of the transition probability matrix. The values in the eigenvector are then used to determine the *outlierness* of each object. A key challenge of using the random walk approach is defining the appropriate similarity metric for constructing the neighborhood graph. The paper proposes

two approaches for defining similarity: (1) cosine similarity between objects and (2) shared-nearest neighbor density.

The Markov chain formulation, is as follows:

$$c \;=\; \mathbf{S}^T c,$$

where $\mathbf{S}$ is the transition matrix and c is the stationary distribution representing connectivity value for each object in the dataset. For a general transition matrix, neither the existence nor the uniqueness of a stationary distribution is guaranteed, unless the transition matrix is irreducible and aperiodic, as stated in the Perron-Fobenius theorem. The transition matrix, $\mathbf{S}$, of the Markov chain model is obtained by normalizing the originally defined similarity matrix, $\mathbf{S}_O$:

$$\mathbf{S}(i,j) \;=\; \frac{\mathbf{S}_O(i,j)}{\sum^n \mathbf{S}_O(i,j)}.$$

The normalization ensures that the row sum for each row of the transition matrix equals 1, which is a required property of the Markov chain. Another important property is that the probabilities in $\mathbf{S}$ do not change with time. After computing the transition matrix $\mathbf{S}$, we must be sure that it is both irreducible and aperiodic so that a unique stationary distribution is achieved [33]. An approach provided in [64].

The OutRank-α algorithm uses cosine similarity to form the transition probability matrix for the Markov chain model. In this model, the initial connectivity vector is set to $1/n$, where n is the number of nodes, and the damping factor is set to 0.1. For the OutRank-β algorithm, a similarity measure is used that considers the number of common neighbors for each node pair. In particular, this algorithm uses the cardinality of the set of common neighbors in the similarity measure.

In 2010, Akoglu *et al.* presented a method for the identification of outliers called OddBall. Outliers are identified using their neighborhoods, that is, a sphere, or a ball that surrounds the node (hence the name OddBall). The goal is to spot strange nodes in a graph with weighted edges. The proposed methods uses *egonets*, which are the induced subgraph of the node of interest and its neighbors; and gives a set of numerical features for egonets. To identify outliers, the methods uses patterns that egonets follow, such as patterns in density, weights, principal eigenvalues, and ranks.

4.2.2 Characteristics of patent citation networks

Patent citation networks have specific characteristics. For example, graph structure contains important citation relationship information among patents. Additionally, patents (nodes) can be seen as individual objects described by carefully assigned multivariate attribute data. The rich combination of these data types is called *attributed graphs* or *information networks*. For this specialized data, there are very few methods for graph outlier detection, in general. **There is currently no existing approach for node outlier ranking that considers both graph structure and attribute data for patents in PCN.**

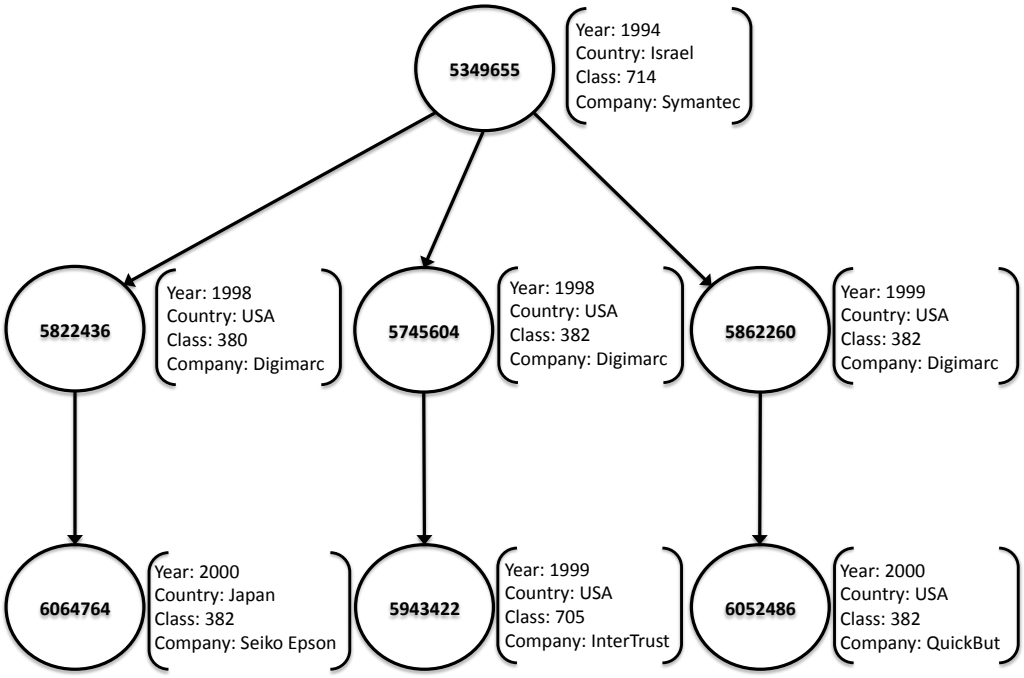

Figure 4.3: Attributed graph representation of patent citation network – combined graph structure and node attribute data

Attributed graphs, also called *information networks*, contain additional data to the usual $G := (N, E)$ graph data. In the attributed graph data, nodes are individual

objects. In our case, a node represents an individual patent in a patent citation network. The difference from traditional graph data is that nodes are described by multivariate attribute data. Edges between nodes still signify some relationship between the nodes. For example, in a patent citation network, edges may represent citations among patents. An example attributed graph for patent data is shown in Figure 4.3. In this figure, the citation information between patents is represented by directed edges, and node attributes are shown in an attribute vector.

4.2.3 Graph structure and node attribute-based node outlier ranking methods

Traditional outlier ranking techniques focus on either multivariate (vector) data or on data in the form of graphs. As we have described, there are however datasets that contain both, multidimensional information and relationships between objects in the form of attributed graphs. There are currently two existing approaches for identifying outliers in graphs, based on both node attributes and graph structure. The GOutRank approach from 2013 [56] handles such datasets by combining the outlier scoring. The approach first uses a subspace cluster result of the attributed graph data, which in this case is allowed to have cluster overlap, in that a node may belong to more than one cluster. A *subspace clustering* result in an attributed graph is a set of subspace clusters $Res = \{(C_1, S_1) \dots (C_n, S_n)\}$, where $C_i \subset V$ is a densely connected subgraph with high attribute similarity in the subspace $S_i \subset A$, where A is the set of all attributes.

For the second approach, in 2010, Goa *et al.* present a method for the identification of *community outliers* [25]. This approach addresses a different problem from the problem we are addressing, since their goal is to find the outlier *within* a given community. In the next paragraphs we will focus on the GOutRank method presented in [56], and show how our proposed subspace clustering approach outperforms this existing method.

In 2013, Muller *et al.* present an approach for outlier ranking using subspaces of attributed graphs [56]. Subspaces are subset of the attribute space, as opposed to the full feature space. The method, GOutRank, introduces a methodology for scoring and ranking nodes of a graph by deviation in both graph and attribute properties.

First, GOutRank score function for node n, based only on the subspace clustering result, is defined as:

$$\text{GOutRank}(n) \quad = \quad \frac{1}{2} \times \sum_{\{(C,S) \in Res | o \in C\}} \frac{|C|}{C_{max}} + \frac{|S|}{S_{max}}, \tag{4.1}$$

where $|C|$ is the number of objects in cluster C, and $|S|$ is the number of attributes used to define the subspace; C_{max} is the maximum cluster size in Res and S_{max} the maximal dimensionality of all subspace clusters in Res. Note that this fist formulation for outlier score uses **only the subspace clustering result** for outlier score. Note also that all nodes in the same cluster will have the same outlier score using this formulation since there is no way to distinguish the outlierness of nodes in the same cluster. Our proposed outlier scoring approach will directly address this drawback, and provided a way to distinguish the outlierness of nodes in the same cluster.

GOutRank was modified to include a graph structure term in the outlier score function. The score function for node n with node degree scoring is defined as:

$$\text{GOutRank}(n) \quad = \quad \frac{1}{3} \times \sum_{\{(C,S) \in Res | o \in C\}} \frac{|C|}{C_{max}} + \frac{|S|}{S_{max}} + \frac{deg(n)}{deg_{max}}, \tag{4.2}$$

where $|C|$ is the number of objects in cluster C, and $|S|$ is the number of attributes used to define the subspace; C_{max} is the maximum cluster size in Res, and S_{max} the maximal dimensionality of all subspace clusters in Res. deg_{max} is the maximum degree for all nodes in the overall network. The third term in the formulation considers the graph properties of each node. Specifically, the term considers degree, or number of connected edges, for node n, $deg(n)$. We note that this value only considers the direct neighbors of a node and does not consider any indirect link information. Traditionally, degree is used as a basic centrality measure. A contribution of our proposed outlier scoring approach will address this drawback by considering indirect links and co-citation links in the graph structure.

4.2.4 Subspace clustering for outlier detection

In data mining, clustering is a process that aims to group similar objects while separating dissimilar ones. For traditional clustering of multivariate data, the similarity

of objects corresponds to the similarity of attribute values. For graph data, clustering methods determine groups of nodes that correspond to densely connected subgraphs within a graph. We begin this section with a review of *subspace clustering*. As with the datasets used in this work, much of the complex data today has two aspects: **attribute data** to characterize single objects and **graph data** to represent some relationship between objects. Researchers have found that analyzing both data sources simultaneously can increase the quality of mining methods [29]. Combined object attribute and graph structure clustering approaches have been introduced [29]. Approaches detect *densely connected node sets* within a large graph that also show high similarity according to their attribute values. Depending on the attribute set, it may be that full-space clustering leads to uninformative clustering results. For this reason, subspace clustering was introduced to identify *locally relevant subsets of attributes* for each cluster. We will review in detail a method to determine sets of nodes that show high similarity in subsets of their dimensions (*i.e.* attributes), and that are also densely connected within their given graph. Resulting clusters are optimized according to their graph density, size (in terms of number of nodes), and number of relevant attributes or dimensions. The goal of the existing model, called GAMER, is to confine the clustering to a manageable size of only the most interesting clusters, and was not originally developed for node outlier ranking.

Subspace clustering is an important process because it avoids the curse of dimensionality. Additionally, it avoids identifying an outlier node based on single or few attributes (which may have little significance) in high-dimensional data. Subspace clustering also allows for identification of nodes that may be outliers in various ways, considering some combinations of the attributes. In other words, there may exist nodes in the attributed graph that are similar if some subset of the attributes are considered while others are ignored, which helps to identify nodes that are outliers in particular ways. In this way, subspace clustering methods detect relevant subspace projections individually for each cluster [44]. Consider the following example for demonstration purposes. In this example nodes are persons, edges are friendships, and the attribute

tuple considered is a person's *(Interest, Age, Employer)*. First, we consider the *(Interest, Employer)* pair:

1. Nodes in subspace cluster 1: **Interest:** Programming, **Employer:** Tech company
2. Nodes in subspace cluster 2: **Interest:** Art, **Employer:** Museum
3. Node in no subspace cluster: **Interest:** Art, **Employer:** Tech company

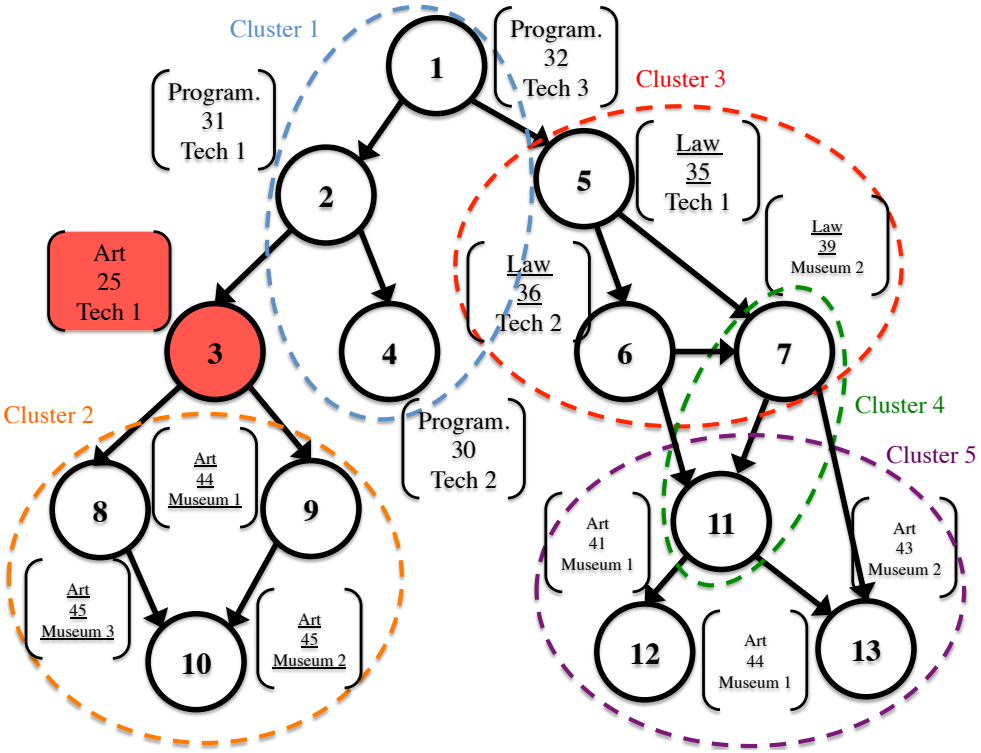

Figure 4.4: Example of the ability of subspace clustering to identify node clusters based on various attribute combinations and graph connectivity over the network

In this example, subspace cluster 1 is shown at the top of the graph, while subspace cluster 2 is shown at the bottom left. Node(s) that do not belong to any cluster, such as *(Interest, Employer)* subspace clustering, including Node 3, are more likely potential outliers, based only on subspace clustering, since they are not highly clustered with other nodes in the network. Notice that cluster 2 and cluster 5 are both **Interest:** Art, **Age:** early to mid-40s, and **Employer:** Museum. The reason these are two

different subspace clusters is because the two clusters are **separated in terms of graph structure**. In this way, subspace clustering in attributed graphs takes on a different purpose than traditional multivariate data clustering – **both attribute values and graph structure must be considered simultaneously in order to cluster nodes**. Another key difference is that clusters may be defined by 2 of 3 attributes (*i.e.*, some subspace of the attributes) or 3 of 3 attributes (*i.e.*, full space, depending on algorithm parameter value for number of attributes), allowing for more flexibility in the identifying of clusters. For example, cluster 4 uses the two attributes *(Age, Employer)*. Finally, we allow for overlap in clusters so that we can identify the multiple ways in which nodes are similar to other nodes. See Nodes 7 and 11, for example, which belong to two different clusters each, indicating that they are similar to other nodes, and thus have less outlierness.

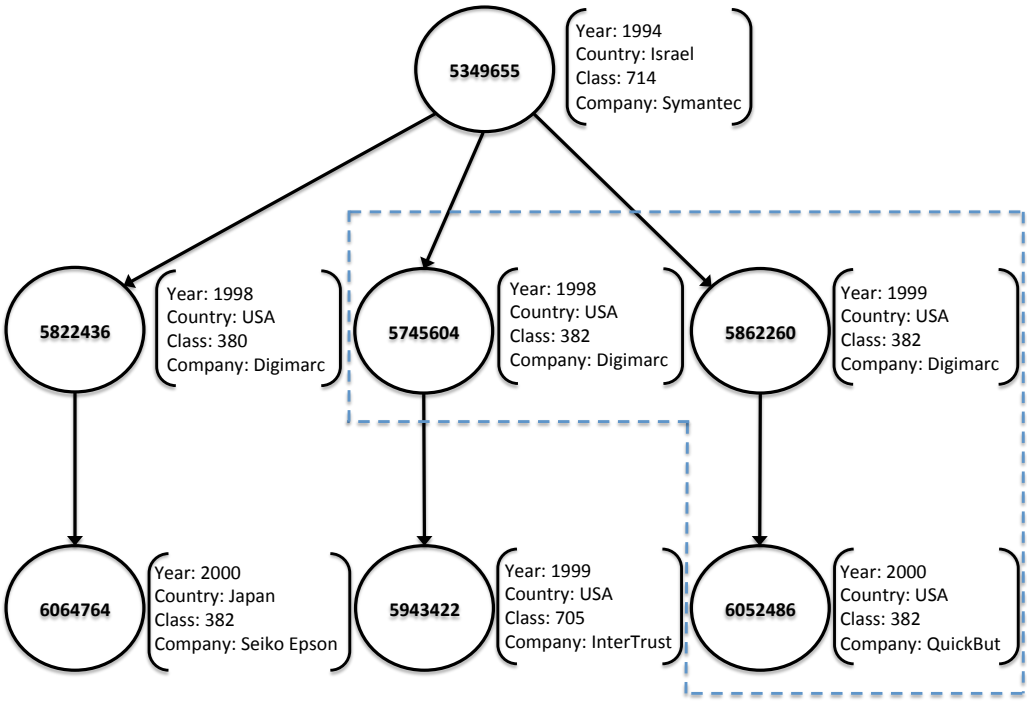

Figure 4.5: Patent citation network with combination of attribute and graph data, demonstrating one potential attribute subspace cluster, considering the attribute subspace *(Country, Class)*

For our problem, we have a dataset that includes node attributes and graph structure. Our subspace clustering algorithm considers both types of data to find subspace clusters. Because our input data includes graph data, our subspace clustering takes on important role, as it also identifies clusters considering graph data, and not merely attribute values. That is, we do not merely use a multivariate clustering result, nor do we merely use a graph structure clustering result, but both multivariate data and graph data are considered simultaneously for subspace clustering. Nodes in clusters may have the same or similar attribute values but be separated within the the graph structure, so they are in different clusters. Note that the result of our subspace clustering algorithm will be used in our **outlier score function**, shown in Equation 4.7, so we leverage the ability of nodes belonging to more than one cluster as we calculate the outlier score for each node.

In this section, we model attribute data together with graph data by using a vertex-labeled graph $G := (V, E, l)$ with vertices V, edges $E \in V \times V$ and a labeling function $l : V \rightarrow \mathbb{R}^d$, where $Dim = 1, ..., d$ is the set of dimensions, or attributes, that describe objects.

Next we describe a state-of-the-art subspace clustering algorithm for attributed graphs, called GAMER, as presented in [29]. At a high level, the GAMER algorithm is as follows:

% Step 1
for all 2^V -1 many possible vertex combinations **do**
  for for all 2^d -1 many possible attribute dimension combinations **do**
    if subset satisfies Definition 1 for the attribute similarity condition **then**
      add to set STEP_1_CLUSTERS
    end if
  end for
end for
% Step 2
if a cluster in STEP_1_CLUSTERS meet the graph connectivity condition **then**
  add to set STEP_2_CLUSTERS

end if

% Step 3:

if a cluster in STEP_2_CLUSTERS is found to be redundant **then**

remove redundant subspace cluster from STEP_2_CLUSTERS

end if

Now we go into the details of the GAMER algorithm, and break up the steps according to the steps provided in the original literature.

GAMER Step 1: Find subspace clusters considering object attribute values

Consider all possible combinations of attribute subspaces and all vector combinations. For vertex combinations, there will be a total of $2^V - 1$, where V is the number of vertices in the graph. For attribute subspaces, there will be a total of $2^d - 1$, where d is the number of attributes for the overall dataset. A subspace cluster is a set of objects with a set of relevant dimensions, where within the relevant dimensions the variation of the object's attribute values is restricted to a maximal width w. Of all possible subspaces, choose the subspaces satisfying the following definition.

Definition 1: Subspace cluster property

Given a set of vectors $X \subseteq \mathbb{R}^d$ and **given** a set of dimensions $S \subseteq Dim$, the pair (X, S) is a subspace cluster, if it meets the following two conditions:

(1.1) $\forall i \in S : \forall x_1, x_2 \in X : |x_1[i] - x_2[i] \leq w$

(1.2) $\forall i \in Dim \setminus S : \exists x_1, x_2 \in X : |x_1[i] - x_2[i] > w$

GAMER Step 2: Calculate density of resulting quasi-clique

The density of a quasi-clique is given by:

$$\gamma(O) \quad = \quad \frac{min_{v \in O}\{deg^O(v)\}}{(|O| - 1)}.$$

where $deg^O(v)$ is the degree of vertex v within the set O. That is,

$deg^O(v) = |\{o \in O | (v, o) \in E\}|$.

GAMER Step 3: Find twofold clusters

Twofold clusters satisfy the following requirements:

(3.1) Dimension requirement: (X, S) is a subspace cluster with $|S| \geq S_{min}$

(3.2) Density requirement: O fulfills the quasi-clique property with $\gamma(O) \geq \gamma_{min}$

(3.3) Cluster size requirement: Induced subgraph of O is connected and $|O| \geq n_{min}$

GAMer Step 4: Calculate quality of a twofold cluster

Given a twofold cluster $C = (O, S)$, the quality of C is given by:

$$Q(C) = \gamma(O)^a \times |O|^b \times |S|^c$$

GAMer Step 5: Check redundancy of subspace clusters

The redundancy model identifies clusters that provide little or no additional information when compared to another cluster. There are two redundancy parameters: r_{obj} for the objects in a cluster, and r_{dim} for the attributes (dimensions) that describe a cluster. Given the redundancy parameters $r_{obj}, r_{dim} \in [0, 1]$, the binary redundancy relation denoted $\prec_{red}$ is defined by:

For all twofold cluster pairs $C = (O, S)$, and $\overline{C} = (\overline{O}, \overline{S})$:

$$C \prec_{red} \overline{C} \Leftrightarrow Q(C) < Q(\overline{C}) \wedge \frac{|O \cap \overline{O}|}{|O|} \geq r_{obj} \wedge \frac{|S \cap \overline{S}|}{|S|} \geq r_{dim}.$$

GAMer Step 6: Output optimal overall clustering

After defining a relation for pairwise redundancy of clusters, the algorithm now defines the overall clustering, *i.e.*, given all twofold clusters $Clusters$ they want to get a meaningful subset $Result \subseteq Clusters$.

The final clustering must be redundancy-free and maximal. Assume we are given the set of all twofold clusters in the set $Clusters$, the **optimal** twofold clustering set $Result \subseteq Clusters$ satisfies the following two conditions:

6.1 $\neg\exists C_i, C_j \in Result : C_i \prec_{red} C_j$ (redundancy-free requirement)

6.2 $\forall C_i \in Clusters \setminus Result : \exists C_j \in Result : C_i \prec_{red} C_j$ (maximality)

Given the set of all twofold clusters in the set $Clusters$ as $\{C_a, C_b, C_c\}$ with binary redundancy relationships:

$$
\begin{aligned}
C_a &\prec_{red} C_b, \\
C_b &\prec_{red} C_c, \\
\neg(C_a &\prec_{red} C_c),
\end{aligned}
$$

the resulting subspace clusters $Result = \{C_a, C_c\}$.

The GAMER subspace clustering algorithm, summarized above, is used in the GOutRank outlier score function shown in Equations 4.1 and 4.2. For applications to PCN, it has a couple of significant drawbacks. (1) For sparse data, a subspace cluster is identified in the case where one patent has attribute value 0 and another patent has less than or equal to specified threshold, w. The drawback results from the attribute similarity measure simply checking the absolute value of the difference of the node attributes. This means patents with no subclasses and patents with a small number of subclasses can be clustered together (see later in this section for a detailed description of patent attribute data used). (2) The GAMER algorithm only considers direct links between objects for its graph connectivity measure, since the minimum node degree among nodes in a cluster is used. This means that the entire cluster graph structure is not considered, but only a characteristic of a single node is considered.

As seen in Section 4.2, many node outlier ranking methods focus strictly on graph structure. Graph structure-based models often only consider a nodes direct connections (or neighbors). Most node outlier ranking approaches, including SCAN, GOutRank-α, GOutRank-β, and OddBall, do not simultaneously consider both patent attribute data and graph structure data. Graph structure-based models have the drawback of not considering the important characteristics of patent citation network data such as **co-citation relationships** and the node similarity that can be mined from those relationships. Another major issue is that to date no approach has been developed specifically for patent citation networks.

Due to these drawbacks, the outlier detection problem in PCN data has much opportunity for contribution. In this work we propose the use of both graph structure data and node attribute data to improve the results of outlier ranking of patents in a

patent citation network. In particular, the first contribution is in the area of subspace clustering in PCNs, in which we define new attribute similarity and graph connectivity criteria. The second contribution allows for nodes in the same cluster to be distinguished in outlierness. We are better able to quantify a nodes regularity by considering the attributes that describe nodes. In particular, we consider the distance of an object from the center of the cluster to which it belongs. Existing graph structure-based approaches only use direct neighbor information, and do not consider indirect node relationships. We propose using a similarity matrix, which is constructed from the adjacency matrix, but considers multi-stage indirect co-citations between pairs of patents. The similarity measure results provides more information on the relationship for a node to the other nodes in the graph to which it belongs than using only adjacency information. To summarize, the key contributions of this chapter are:

1. A new patent subspace clustering algorithm for outlier ranking, based on attribute and graph data

    1.1. A new attribute similarity measure for sparse data

    1.2. A new graph connectivity measure based on direct and indirect links within a cluster

2. A weighted subspace clustering measure for node regularity within a cluster

3. A new graph-based node outlier measure for patents based on node centrality and the co-citation similarity measure

In this work we use the USPTO U.S. class codes for attributes to describe the patents (nodes) in the network. Values for the class code attributes then will be the total number of *subclass* codes for that class. This will be the entirety of the vector data used in our work going forward. See Table 4.2 for a sample of patent classes and subclasses. We will demonstrate the patent attribute data extraction from this type of raw class/subclass data, to the final multivariate attribute data. Based on this subclass count attribute vector data, outlier patent may be one of the following:

(1) Belong to different technology area than rest of dataset

(2) Focused in narrow technology area (high count of subclasses within one/few classes)

(3) Spread over many technology areas (lower count of subclasses within many classes).

Table 4.2: Sample U.S. Class codes for U.S. patent data

Patent ID	U.S. patent class codes class/subclass
6658432	707/999.001; 707/999.104; 707/999.107; 707/E17.117
6226618	705/51; 380/279; 380/281; 380/282; 380/285; 380/30; 380/44; 705/53; 705/57; 705/59; 705/71
6389402	705/51; 348/E5.006; 348/E5.008; 348/E7.06; 348/E7.07; 375/E7.009; 375/E7.024; 380/201; 705/1.1; 705/37; 705/53; 705/57; 705/80
6016476	705/18; 705/26.1; 705/42; 705/44; 705/65; 705/76; 713/186
6427140	705/80; 348/E5.006; 348/E5.008; 348/E7.06; 348/E7.07; 375/E7.009; 375/E7.024; 375/E7.025; 705/53; 713/193

Based on the five U.S. patents and the U.S. classes/subclasses in Table 4.2, we get the classes in Table 4.3, with six unique classes: 348, 375, 380, 705, 707, and 713.

The result is the attribute vector for each of the five patents (patent row data from Table 4.2 is shown in column vector form):

$$v_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 0 \end{bmatrix} \quad v_2 = \begin{bmatrix} 0 \\ 0 \\ 6 \\ 5 \\ 0 \\ 0 \end{bmatrix} \quad v_3 = \begin{bmatrix} 4 \\ 2 \\ 1 \\ 6 \\ 0 \\ 0 \end{bmatrix} \quad v_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 0 \\ 1 \end{bmatrix} \quad v_5 = \begin{bmatrix} 4 \\ 3 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

Table 4.3: Example patent attributes: number of subclasses within U.S. class codes for the five U.S. patents from Table 4.2

Patent ID	U.S. class codes subclass counts					
	348	375	380	705	707	713
6658432	0	0	0	0	4	0
6226618	0	0	6	5	0	0
6389402	4	2	1	6	0	0
6016476	0	0	0	6	0	1
6427140	4	3	0	2	0	1

4.3 New subspace clustering algorithm for patents in a patent citation network

Subspace clustering will be based on patent attributes and graph connectivity. See the attributes shown in Table 4.4 for example. An example of the key subspace clustering idea is as follows. Class A and Class D values for Patent 1 and Patent 3 are similar, indicating they have similar combination of technology, while Patent 2 has value 0 for both of those classes, thus we are able to distinguish Patent 2 as not belonging to the same subspace cluster, giving Patent 2 a higher outlier score. Consider the class examples:

Class A: Encryption

Class B: Databases

Class C: Programming languages

Class D: Hardware

We can cluster Patent 1 and Patent 3 as patents related to hardware encryption, while Patent 2 is related to databases, and is not in that same cluster. If Patent 2 is not highly clustered with other nodes in the network, it is more likely to be an outlier, using our proposed score function.

Table 4.4: Node attributes example

	Class A	Class B	Class C	Class D
Patent 1	**7**	0	0	**2**
Patent 2	0	2	0	0
Patent 3	**5**	0	0	**3**

The high level steps for our proposed Patent Clustering for Outlier Ranking, PCOR, are as follows:

Step 1: Find subspace clusters based only on object attributes, add to set STEP_1_CLUSTERS.

Step 2: Refine those subspace clusters in STEP_1_CLUSTERS based only on graph connectivity of cluster, add to set STEP_2_CLUSTERS.

The proposed algorithm works by first clustering nodes based on attribute values

and then refining those clusters based on graph connectivity. In Step 1, we consider that the PCN data is sparse data, and avoid clustering patents based on zero-value attributes. This reduces the number of uninformative clusters that result using existing approaches. In Step 2, we consider how well clusters are connected by considering the direct and indirect links among nodes within a subspace cluster. This approach is an improvement over the existing approach since it is a cluster-based measure, rather than a node-based measure. That is, the links of the entire cluster are considered, not just the direct links of the least well connected node.

4.3.1 Attribute similarity criterion

PCOR Step 1: Find subspace clusters based only on object attribute values

We again consider all possible combinations of attribute subspaces and all vector combinations. A subspace cluster is a set of objects with a set of relevant dimensions, where within the relevant dimensions, the variation of the objects' attribute values is restricted to a ratio greater than or equal to q_{min}. Of all possible subspaces, we choose the subspaces satisfying the following definition.

Definition 1: Subspace cluster property

Given a set of vectors $X \subseteq \mathbb{N}^d$ and **given** a set of dimensions $S \subseteq Dim$, the pair (X, S) is a subspace cluster, if it meets the following three conditions:

(1.1) $\forall i \in S : \forall x_1, x_2 \in X : x_1[i] \neq 0 \wedge x_2[i] \neq 0$

(1.2) $\forall i \in S : \forall x_1, x_2 \in X : min\left\{\frac{x_1[i]}{x_2[i]}, \frac{x_2[i]}{x_1[i]}\right\} \geq q_{min}$

(1.3) $\forall i \in Dim \setminus S : \exists x_1, x_2 \in X : min\left\{\frac{x_1[i]}{x_2[i]}, \frac{x_2[i]}{x_1[i]}\right\} < q_{min}$

4.3.2 Graph connectivity criterion

PCOR Step 2: Calculate graph connectivity of resulting subspace clusters from Step 1

After obtaining the set of subspace clusters based only on attribute data, STEP_1_CLUSTERS, check the indirect links among objects within the cluster.

Definition 2: Graph connectivity property

$I^M(o_{i,j})$ is the **level-M node-cluster indirect link measure** for node $o_{i,j}$, which is node i in cluster j, and is given by:

$$I^M(o_{i,j}) = \sum_{r=1}^{M} \sum_{k=1}^{N} A^r_{C_i\, j,k},$$

where $\mathbf{A}^{\ r}_{C_i}$ is the adjacency matrix for objects within cluster C_i, M is the desired length of indirect path between two nodes within the subspace cluster to be considered, and N is the number of nodes in PCN. If node p is not in this cluster, then the pth row and pth column of the original adjacency matrix are 0 vectors in $\mathbf{A}_{\ C_i}$. In order for a cluster (C_i, S_i) to be in STEP_2_CLUSTERS, it must meet the following constraint:

$$c(O_i) \geq c_{min},$$

where $c(O_i) = min_j\{I^M(o_{i,j})\}/|O_i|$ is the *node-cluster indirect link* value for object o, and I^M_{max} is the maximum *node-cluster indirect link* value over all objects in cluster C_i. The algorithm is summarized in Algorithm 1. A numerical example is presented to show the advantages of our proposed approach over GAMER, using the attributed patent citation network shown in Figure 4.6.

4.3.3 Subspace clustering numerical example

In this section we present a numerical example for our proposed subspace clustering algorithm, PCOR, and compare it to the state-of-the-art GAMER algorithm. The attributed graph used as input for this numerical example is seen in Figure 4.6. Parameters for both the proposed subspace clustering algorithm PCOR and the existing GAMER algorithm are given in Table 4.5. Note that PCOR has some similar parameters to GAMER, but fewer overall parameters. In particular, PCOR does not have parameters for reducing the redundancy in the resulting subspace clusters. This is because PCOR was developed specifically for outlier detection, while GAMER was

Algorithm 1 Patent subspace clustering algorithm

procedure PCOR(attributed graph, $\mathrm{n}_{min}, a_{min}, q_{min}, c_{min}$)
  % Step 1: Find valid attribute subspace clusters based only on
  % object attribute values
  for ($\forall o \subseteq$ O, where $|o| \geq \mathrm{n}_{min}$) **do** ▷ x are attribute vectors for objects o
    for ($\forall s \subseteq$ S, where $|s| \geq a_{min}$) **do**
      for $(\forall x_1, x_2 \in X, \forall e \in s)$ **do**
        if $(x_1[e] \neq 0$ && $x_2[e] \neq 0)$ &&
          $min\{x_1[e]/x_2[e], x_2[e]/x_1[e]\} \geq q_{min}$ &&
          $\forall i \in \backslash s : \exists x_1, x_2 \in X : min\left\{\frac{x_1[e]}{x_2[e]}, \frac{x_2[e]}{x_1[e]}\right\} < q_{min}$ **then**
          % (X_i, S_i) meets the attribute similarity conditions,
          % (X_i, S_i) is specific *(attribute set, node set)* pair
          <add (X_i, S_i) to set STEP_1_CLUSTERS>
        end if
      end for
    end for
  end for

  % Step 2: Select portion of STEP_1_CLUSTERS based on
  % graph connectivity
  for ($\forall i \in$ attribute-based clusters set STEP_1_CLUSTERS) **do**
    <calculate cluster graph connectivity using indirect links>
    for ($\forall j \in$ nodes in this ith subspace cluster) **do**
      <calculate connectivity for this node $o_{i,j}$ in this cluster>
      %$A^r_{C_i}$ is rth power of adjacency matrix of cluster i
      %M = number of indirect levels, N= number of nodes in PCN
      $I^M(o_{i,j}) = \sum_{r=1}^{M} \sum_{k=1}^{N} A^r_{C_i j,k}$ ▷ $o_{i,j}$ is node j of cluster i
    end for
    $c(O_i) = min_j\{I^M(o_{i,j})\}/|O_i|$
    if $c(O_i) \geq c_{min}$ **then**
      % (X_i, S_i) meets the graph connectivity condition
      <add (X_i, S_i) to set STEP_2_CLUSTERS>
    end if
  end for
  FinalClustering ← STEP_2_CLUSTER
  return FinalClustering ▷ Final patent subspace cluster result
end procedure

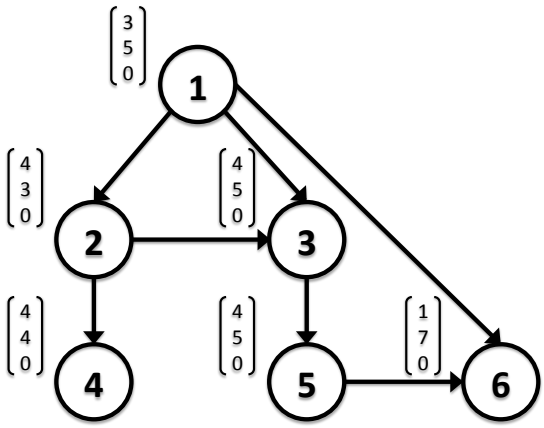

Figure 4.6: Example attributed PCN to demonstrate subspace clustering algorithm

developed as a general purpose subspace clustering algorithm. In our algorithm, we want to identify all the ways in which patents are similar to other patents in the PCN, so we do not perform a redundancy reduction step as the final step of our algorithm. We break down the steps of the algorithms into two common steps for both algorithms, so that the working of the algorithms can be compared side-by-side. Step 1: find subspace clusters based on attribute values only. Step 2: based on the result of Step 1, select the portion of subspace clusters that satisfy the graph connectivity requirement. We again note that GAMER would have, Step 3: reduce the redundancy in the set of subspace clusters from Step 2, while PCOR has no such step.

Step 1: Find subspace clusters based on attribute values only. Let the set S be the set of attributes used in this first possible subspace clustering, where each value in the set S refers to the attribute number. Let O_i be the ith subspace cluster where the values in the set O_i are the node IDs belonging to that subspace cluster.

GAMER:

$\underline{S = \{1, 2, 3\}}$

$O_1 = \{1, 2, 3, 4, 5\}$

Table 4.5: Subspace clustering algorithm parameters

GAMer	PCOR	Parameter meaning
$w = 2.0$	$q_{min} = 0.6$	Attribute similarity threshold
$\gamma_{min} = 0.5$	$c_{min} = 0.5$	Graph connectivity threshold
$n_{min} = 4$	$n_{min} = 4$	Minimum number nodes per cluster
$a_{min} = 2$	$a_{min} = 2$	Minimum number attributes per cluster
$r_{obj} = 0.2$	[none]	Redundancy in nodes (objects) threshold
$r_{dim} = 0.2$	[none]	Redundancy in attributes (dimensions) threshold
$a = 1$	[none]	Weight for graph structure cluster quality term
$b = 1$	[none]	Weight for number of nodes cluster quality term
$c = 1$	[none]	Weight for number of attributes cluster quality term

$O_2 = \{1, 2, 3, 4\}$

$\ldots$

$O_6 = \{2, 3, 4, 5\}$

$\underline{S = \{1, 2\}}$

$O_7 = \{1, 2, 3, 4, 5\}$

$\ldots$

$O_{12} = \{2, 3, 4, 5\}$

$\underline{S = \{1, 3\}}$

$O_{13} = \{1, 2, 3, 4, 5\}$

$\ldots$

$O_{18} = \{2, 3, 4, 5\}$

$\underline{S = \{2, 3\}}$

$O_{19} = \{1, 2, 3, 4, 5\}$

$\ldots$

$O_{24} = \{1, 3, 5, 6\}$

$O_{25} = \{2, 3, 4, 5\}$

The major drawback of this step of the GAMER algorithm is that GAMER does not check for zero values among attributes. As a result, the algorithm identifies many subspace clusters because of the attributes with zero values, and potentially clusters those patents with other patents that have value less than or equal to the threshold, w.

PCOR:

$\underline{S = \{1, 2\}}$

$O_1 = \{1, 2, 3, 4, 5\}$

$O_2 = \{1, 2, 3, 4\}$

$O_3 = \{1, 2, 3, 5\}$

$O_4 = \{1, 2, 4, 5\}$

$O_5 = \{1, 3, 4, 5\}$

$O_6 = \{2, 3, 4, 5\}$

Our proposed algorithm checks for zero values in the comparison of attribute values, and uses a ratio comparison rather than simple absolute value of difference (width), in order to better handle sparse data. The result is fewer and higher quality subspace clusters in Step 1.

Step 2: Select portion of clusters from Step 1 based on graph connectivity. We use the graph connectivity measures provided earlier for both GAMER and PCOR to compute these scores.

GAMER:

$\gamma(O_1) = 1/4$

$\gamma(O_2) = 1/3$

...

$\gamma(O_{24}) = 2/3$

$\gamma(O_{25}) = 1/3$

Using min degree only considers the direct links in the cluster for the least well-connected node. This means that the more nodes in the cluster, the less connectivity. With $\gamma_{min} = 0.5$, only cluster O_{24} remains after the graph connectivity check. Low degree of a single node in the other subspace clusters cause the clusters to be lost.

PCOR:

$c(O_1) = 3/4$

$c(O_2) = 1$

$c(O_3) = 1$

$c(O_4) = 0$

$c(O_5) = 0$

$c(O_6) = 2/3$

Our algorithm considers indirect links among nodes within a cluster. The result is that we do not penalize lesser degree nodes, as long as indirect (length greater than one) paths exist among nodes in a cluster.

Step 3: Reduce redundancy among subspace clusters from Step 2.

GAMER:

GAMER uses a quality to reduce redundancy. This is likely because GAMER was not originally developed for outlier detection, but rather for clustering. In this numerical example, there is only 1 subspace cluster, so we do not actually have any redundant clusters to remove.

PCOR:

Our algorithm does not have a redundancy reduction step since our subspace clustering algorithm was developed specifically for outlier detection in PCN. For this reason, we want to know all of the different ways that patents can be clustered with other patents.

Table 4.6: Resulting subspace clusters using existing and proposed algorithms, with attribute set, S, specified

GAMER clustering result	PCOR clustering result
$S = \{2, 3\}$:	$S = \{1, 2\}$:
$O_1 = \{1, 3, 5, 6\}$	$O_1 = \{1, 2, 3, 4, 5\}$ $O_2 = \{1, 2, 3, 4\}$ $O_3 = \{1, 2, 3, 5\}$ $O_4 = \{2, 3, 4, 5\}$

4.4 New node outlier ranking methods for attributed graphs

Next we will apply the PCOR subspace clustering algorithm developed in Section 4.3 of this work for node outlier scoring and ranking in this section. Patent datasets are rich with information. Patent attributes such as classification codes are specifically assigned based on the nature of the technology. Similarly, graph structure data in the form of citations made and received are carefully considered by patent writers. Traditional outlier ranking techniques focus on either homogeneous vector data or on graph structures. Our hypothesis is that outliers are best detected by a combination of all available information. In this section we present score functions for outlier ranking in patent citation networks, which contain both types of data: multi-dimensional numeric and relations between objects in *attributed graphs*. There is currently no work on patent citation network outlier detection that considers attributes in patent citation network, in addition to the citation network structure. We will use the developed PCOR subspace clustering algorithm from the previous section for this outlier ranking.

4.4.1 Integrated graph structure-based and node attribute model

In this section we present the score function for the combined outlier score, starting with the weighted subspace clustering and moving to the graph-based contribution. One of our contributions is the formulation of an outlier scoring function that considers the outlier score based on subspace clustering and on graph structure. To this end, we propose the general integrated outlier score for patent o as:

$$OS_I(o) \quad = \quad (w_C \times OS_C(o)) + (w_G \times OS_G(o)) \,,$$

where w_C is the weight given to the cluster-based outlier score, $OS_C(o)$ is the cluster-based outlier score for object o, w_G is the weight given to the graph-based outlier score, and $OS_G(o)$ is the graph-based outlier score for object o. The cluster-based outlier score will consider both attribute and graph structure information as was presented

in the proposed subspace clustering algorithm above. In the following paragraphs, we propose specific formulations.

Our proposed integrated outlier rank score for patent o, $OS_I(o)$, will be given by a weighted combination of *clustering* and *graph structure* outlier scores. For the clustering outlier score, we use our proposed subspace clustering algorithm that was presented above. In addition, we consider the distance of an object from the center of the cluster to which it belongs so that outlierness of nodes belonging to the same cluster can be distinguished. We call this contribution **weighted subspace clustering**. For the graph structure-based method, we use a node's centrality and a node's similarity to the other nodes in the network to quantify outlierness.

4.4.2 Weighted subspace clustering

The objective of the proposed research is to develop advanced methods in patent outlier ranking leveraging both node attribute data and patent citation data. To this end, we propose new score functions which outperform the GOutRank approach [56]. For the calculation that uses cluster information, we propose the following score function for the outlier score of node n that considers the all clusters to which the node belongs:

$$score_{C_{IF}}(o) = \frac{\alpha}{C_{max}} \sum_i |C_i| \times \mathrm{I}(o \in C_i) + \frac{(1-\alpha)}{S_{max}} + \sum_i |S_i| \times \mathrm{I}(o \in C_i), \quad (4.3)$$

where $\mathrm{I}(o \in C_i)$ is an indicator function such that $\mathrm{I}(o \in C_i) = 1$ if object o is in Cluster C_i, and $\mathrm{I}(o \in C_i) = 0$ otherwise. We extend the idea of object belonging to a cluster to consider the distance of an object to the center of the cluster to which it belongs by introducing the term w_i^o:

$$score_C(o) = \sum_{o \in (C_i, S_i)} w_i^o \left[\alpha \times \left(\frac{|C_i|}{C_{max}} \right) + (1-\alpha) \times \left(\frac{|S_i|}{S_{max}} \right) \right], \quad (4.4)$$

where $|C_i|$ is the number of objects in cluster C_i, and $|S_i|$ is the number of attributes used to describe the subspace cluster; C_{max} is the maximum cluster size in the subspace cluster result, and S_{max} the maximal dimensionality in the subspace cluster result. w_i^o

considers the distance between object and the center of the subspace cluster to which it belongs (based on node attribute values), and is the weight given to object o belonging to cluster i (a function of distance from object o to center of cluster i), and $d^o_{C_i}$ is the actual distance of object o to the center of cluster C_i.

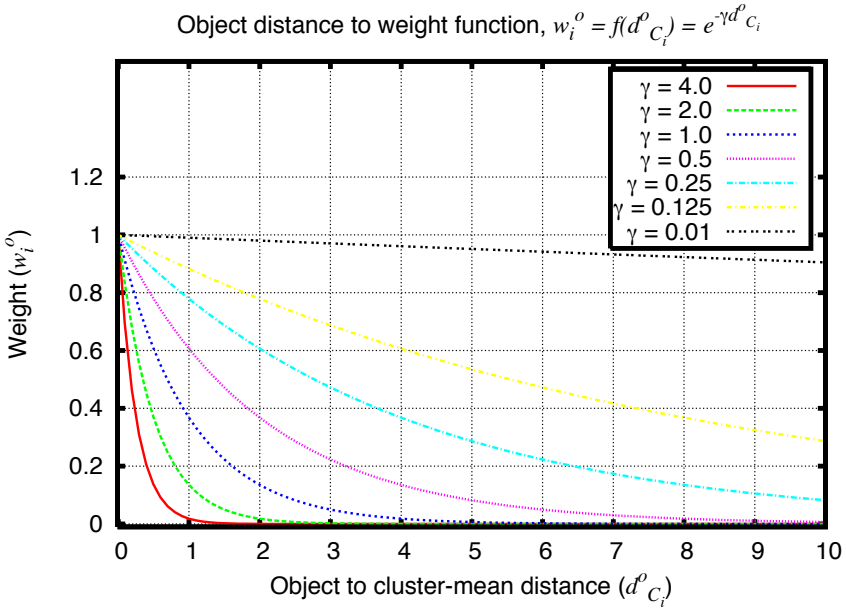

Figure 4.7: Weight function for different γ values

We use a point-cluster distance measure to measure the distance between an object and the center of the cluster(s) that to which it belongs. The Euclidian distance between an object and cluster C_i is denoted $d(\mathbf{o}, C_i)$ and is written as follows:

$$d^o_{C_i} \quad = \quad d(\mathbf{o}, C_i) = d(\mathbf{o}, \bar{\mathbf{X}}_{C_i}) = \sqrt{(o_1 - \mu_{i1})^2 + (o_2 - \mu_{i2})^2 + \cdots + (o_n - \mu_{in})^2},$$

where $n_i = |C_i|$ and where μ_i is the mean of points in the cluster C_i. In order to give a lower regularity score to the nodes that are farther from the center of the cluster(s) to which it belongs, and a greater score to those nodes that are closer to the center of the cluster(s) to which they belong, we have the following formulation:

$$w_i^o = f(d_{C_i}^o) = e^{-\gamma \times d_{C_i}^o}, \tag{4.5}$$

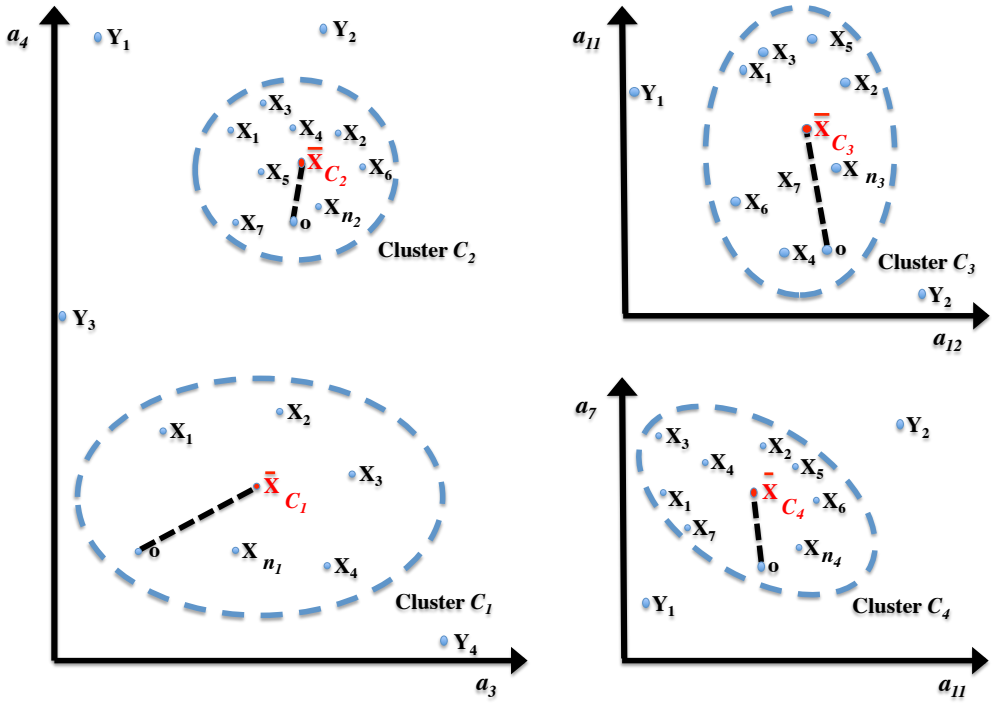

Figure 4.8: Example of four clusterings for different attribute subspaces, and point cluster distance for object o and Cluster C_i

so that the greater the distance from an object to the center of a cluster, the less the weight for object o in cluster i, as seen in Figure 4.7.

In Figure 4.8 Clusters C_1 and C_2 arise from the same attribute subspace, while Clusters C_3 and C_4 each arise from their respective attribute subspace. The axis labels indicate the relevant attribute subspace utilized. In this figure, we show two dimensions for ease of viewing. The idea can be expanded to n dimensions. We use the cluster-based method developed in this section for our integrated outlier ranking formulation so that:

$$OS_C(o) = score_C(o). \tag{4.6}$$

The second portion of the score function will build on $score_C(o)$ score function, and integrates a graph structure-based outlier score from Section 4.4.3. The integrated score function is given in the below Equation (4.7):

$$\begin{aligned} OS_I(o) &= (w_C \times OS_C(o)) + (w_G \times OS_G(o)) \\ &= w_C \times \sum_{o \in (C_i, S_i)} w_i^o \left[\alpha \times \left(\frac{|C_i|}{C_{max}} \right) + (1-\alpha) \times \left(\frac{|S_i|}{S_{max}} \right) \right] + w_G \times \frac{c(o)}{c_{max}}, \end{aligned}$$

where w_C and w_G are the weights for the cluster-based term and the graph structure-based term, respectively. The graph structure-based outlier score is given in the next subsection.

4.4.3 Graph structure-based methods

A node in a network should not be considered an outlier if it is central to the network, or if it is similar to other nodes in the network. Our proposed graph-based outlier score leverages both of these aspects by combining a centrality score and a similarity score in the outlier score function, which is given by:

$$c(o) = \sum_{k=1}^{N} \left[A_{ok} + C_{ok} \right],$$

where $\mathbf{A}$ is the adjacency matrix and $\mathbf{C}$ is the normalized multi-stage co-citation similarity matrix where the similarity between nodes x and y is given by [69]:

$$\mathbf{C} = C^M(x,y)_{normalized} = \frac{C^M(x,y)}{\mathrm{C}^\infty(x,x) + \mathrm{C}^\infty(y,y)}.$$

In this way our outlier score function considers the similarity of patents using indirect links and the co-citation relationship, which is of high importance in PCNs. The

combination of our proposed approaches is seen in Figure 4.9, which shows a flowchart of the entire outlier score process.

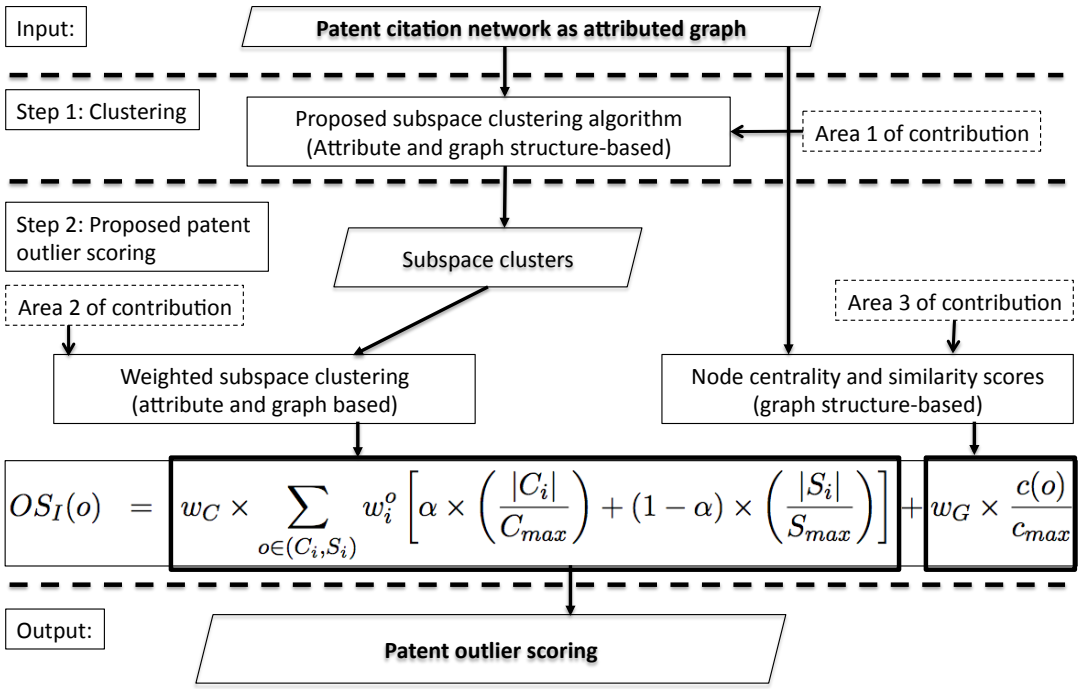

Figure 4.9: Proposed patent outlier ranking flowchart

4.5 Experimental results

In this section we first present experimental results based on three artificial datasets. We use the small artificial datasets to demonstrate the working of our approach. Since true outlier labels are not available for real-life data, we first use the artificial datasets in which node outliers are easy to identify in order to show the working of our methods. The first artificial dataset is a small attributed graph that mimics a real-life patent citation network. The second artificial dataset is a *subspace clustering result* for an attributed graph to demonstrate contribution of the weighted subspace clustering approach. The third artificial dataset contains only graph structure data and highlights the graph-based portion of our outlier score function contribution. In this way, we are able to show the performance individually for each of our contributions

from earlier sections. In the later part of this section, we will present results based on a real-life attributed patent citation network dataset, using all contributions in a combined manner.

4.5.1 Data description: Example patent citation networks

In this section we provide details on three artificial datasets and one real-life dataset. We will use each of the datasets to demonstrate the performance of our proposed methods. First, we use artificial dataset 1 to show the advantage of our subspace clustering algorithm result when used in cluster-based outlier score function. Next, we use artificial dataset 2 to show the advantage of weighted subspace clustering to distinguish outlierness of nodes that belong to the same cluster. Finally, we use artificial dataset 3 to show the advantage of our graph-based outlier score function. After providing results for artificial datasets, we apply our proposed methods to a real-life patent citation network.

Artificial dataset 1: 6-node attributed patent citation network

First we present the example attributed graph from Section 4.3, originally used to demonstrate the proposed subspace clustering algorithm. We compare our experimental results for node outlier ranking with the existing approach. In this dataset, there are six nodes with three attributes describing each node. Attribute values are integers greater than or equal to zero, like the actual attribute values will be in a real-life attributed PCN. The attributes that describe the nodes then are interpreted as class code counts, as were presented in Section 4.2.4.

For this example patent citation network, we provide a "Yes" label, depending on the expected outlierness for each node in the network. That is, if a node clearly is an outlier we mark it as such. If a node is clearly an outlier, then we mark the *Expected Outlier* column of Table 4.9 with "Yes." If it is unclear whether a node should or should not be an outlier, then we mark the column with a "-."

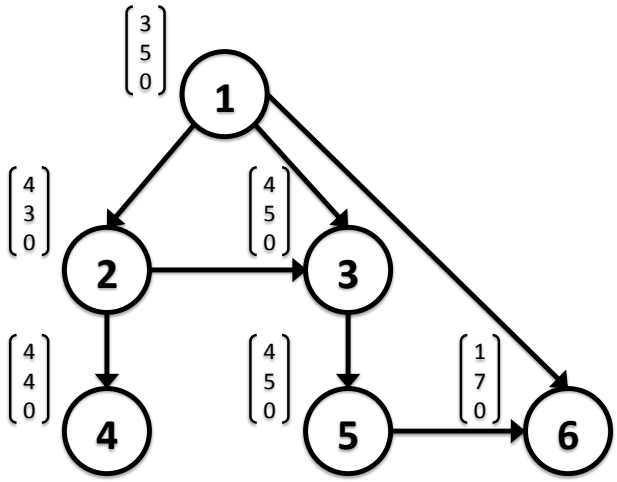

Figure 4.10: Artificial dataset 1 - example attributed PCN for node outlier ranking based on subspace clustering, where attributes indicate patent class counts

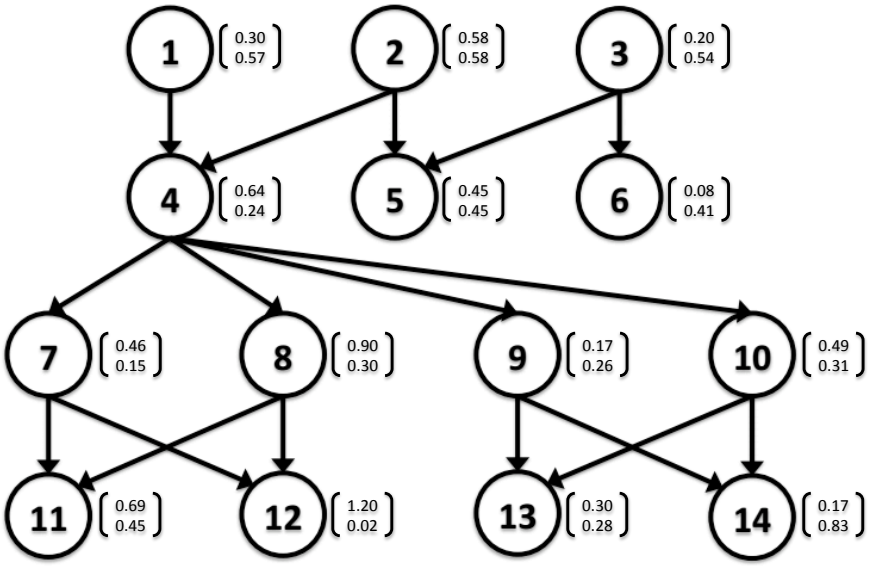

Figure 4.11: Artificial dataset 2 - example attributed graph for outlier ranking based on weighted subspace clustering; two attributes describe each node

Artificial dataset 2: 14-node attributed patent citation network

To test our developed weighted subspace clustering method, we will also use the 14-node example citation network seen in Figure 4.11. Note that this is a directed, acyclic graph, like the patent citation networks we have been working using throughout this work. In addition to the attributed graph, this example dataset also shows a resulting cluster, called C_1. The reason this dataset provides a cluster is so that we can explicitly apply our weighted subspace clustering approach to that cluster. We will use this same clustering result as input to both existing and our outlier score function. Note that the attribute values are different than the ones in our real-life data, but our idea of weighted subspace clustering still holds for this attribute data.

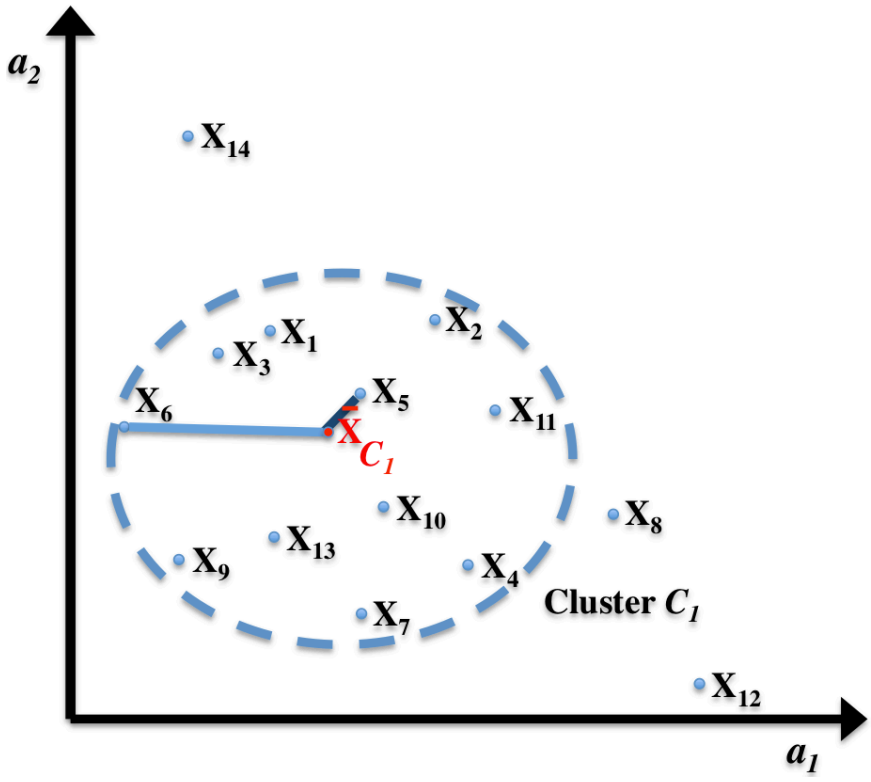

Figure 4.12: Artificial dataset 2 - given clustering of nodes using two attributes to describe nodes

Table 4.7: Attribute values for the 14 nodes of the artificial dataset 2, with attributes a_1 and a_2 describing each node

Node	a_1	a_2
1	0.30	0.57
2	0.58	0.58
3	0.20	0.54
4	0.64	0.24
5	0.45	0.45
6	0.08	0.41
7	0.46	0.15
8	0.90	0.30
9	0.17	0.26
10	0.49	0.31
11	0.69	0.45
12	1.20	0.02
13	0.30	0.28
14	0.17	0.83

Artificial dataset 3: 14-node patent citation network

The value of this example network is that we can systematically determine how we expect nodes in the network to be ranked in outlierness. As an example, we would not expect Node 4 to be an outlier in this network since it is cited by four other nodes (the most citations in this network) and it cites two other nodes (tied for the most in this network). Node 6 in comparison is not cited by any other node, and makes one citation. We will present our general expected outlier ranking for this graph, and compare an existing method to our proposed method.

Real-life patent citation network

Total patent citation network data consists of 4,142 nodes and 18,385 edges, and form a single connected tree structure. The citation network contains directed, unweighted edges. The dataset actually used for the experiments are U.S. patents in the area of information and security issued between 1994 and 2007. For this experiment, we take the top 1% most frequently cited patents from 1994 to 2007 as the nodes in the patent citation network. In order to have a single connected tree structure to which

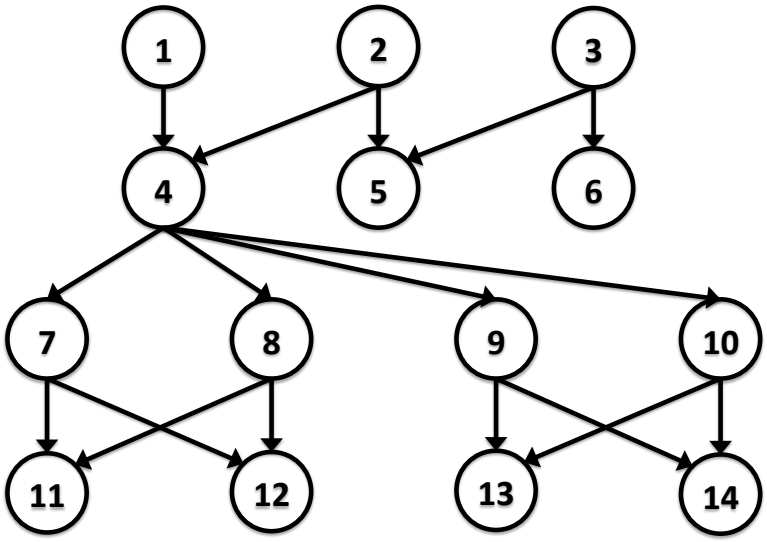

Figure 4.13: Artificial dataset 3 - example PCN graph data

to apply outlier detection method, we select the patents that cite, either directly or indirectly, the most cited patent from the original dataset, which is patent US-5349655. Our patent citation network then consist of 150 nodes and 215 edges. Overall, patent attributes include: patent class codes, year granted, country of origin, and so on. In this work we focus on the U.S. class codes, as was described in Section 4.2.4. This dataset then contains 42 attributes describing each patent.

4.5.2 Artificial dataset 1: 6-node attributed patent citation network

First, outlier scores for artificial dataset 1 using the our proposed subspace clustering and weighted subspace clustering methods (we do not show results for strictly graph-based methods here). Our new subspace clustering method is described in Sections 4.3, and our weighted subspace clustering approach is described in Section 4.4.2. Subspace clustering results for this 6-node PCN dataset was provided in the numerical example, and is summarized in Table 4.6. Table 4.8 provides outlier scores and ranking for artificial dataset 1. Notice in these results that our proposed weighted subspace

clustering is able to assign a unique outlier score to each node in the dataset. Also, node 6 is identified as the outlier which is validated by both its position in the network and its attribute values, as seen in Figure 4.10.

Table 4.8: Comparison of outlier scoring and ranking for artificial dataset 1 using proposed cluster-based method

Outlier rank	Node ID	Outlier score
1	6	0.00
2	2	0.22
3	1	0.38
4	5	0.79
5	3	1.04
6	4	1.14

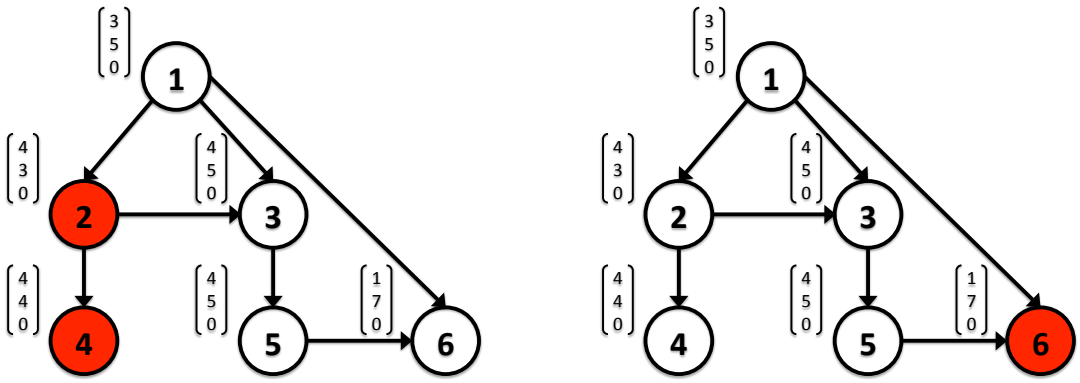

Figure 4.14: Artificial dataset 1 - example attributed patent citation network outlier ranking results using existing (left) and proposed (right) methods, where shaded node indicates outlier

We compare outlier ranking of our proposed methods with that of the existing GOurRank method. In Table 4.9, an outlier rank of 1st identifies the node that is most outlier, while outlier rank of 6th identifies the least outlier node (*i.e.*, most regular node). Node 6 has the outlier attribute values and is not well connected to the rest

Table 4.9: Comparison of outlier ranking for artificial dataset 1 using existing and proposed methods (rank 1st is the greatest outlier)

Node	GOutRank outlier rank	PCOR-based outlier rank	Expected outlier
1	3rd tied	3rd	-
2	1st tied	2nd	-
3	3rd tied	5th	-
4	1st tied	6th	-
5	3rd tied	4th	-
6	**3rd tied (node 6 not an outlier)**	**1st (node 6 is the outlier)**	Yes

of the PCN. Existing methods do not handle sparse data well, and will cluster Node 6 with other nodes because of the zero attribute value for attribute 3. Our proposed approach overcomes this drawback and will not cluster patents based on zero values in the class code attribute. Additionally, the existing subspace clustering method is not able to distinguish the outlier rank of nodes in the same cluster. For this reason, the existing method has two ranks of nodes (note we show competition rank in result tables).

4.5.3 Artificial dataset 2: 14-node attributed patent citation network

Table 4.10: Given subspace clustering result for artificial dataset 2

Subspace cluster	Cluster details			
	Num. attributes used	Attributes used	Num. nodes in cluster	Cluster members
1 of 1	2	a_1, a_2	11	1,2,3,4,5,6,7,9,10,11,13

Next we present our results for artificial dataset 2, which is an example of an attributed PCN, and a given subspace clustering result. In this experiment, we show the value of the weight term w_i^o as proposed in Equation 4.4. In this experiment, we do not find the subspace cluster, but we assume one is already given to us. From this point, we work to find the outlier score and ranking for nodes, based on the node attribute values, and the outlier score function. The major drawback of existing subspace clustering scoring is that all objects in a single cluster will have outlier same score. We demonstrated the ability of our proposed methods to provide a unique outlier score

in the previous section. In this section we provide an example to illustrate in detail how our proposed method works. For example, given the resulting subspace cluster in Table 4.10, using the existing score function for subspace clustering, Node 5 and Node 6 outlier score cannot be differentiated, simply because they belong to the same cluster. As a solution to this major drawback, in Section 4.4.2, we proposed calculating a weight based on each objects distance to cluster mean so that objects have unique score. Using our contribution, Node 5 is nearer to the center of the cluster than Node 6, thus receives a greater weight (to be used as a coefficient) than Node 6. Node 5 is more central to cluster than Node 6, thus Node 6 is more of an outlier, within the cluster to which it belongs. Table 4.11 present the weights for each node, which is a function of the distance to the center of the cluster to which it belongs, as was presented in an earlier section.

Table 4.11: Weights for the 11 nodes that are in cluster C_1

Node	w_1^o
1	0.6924
2	0.6149
3	0.6370
4	0.5953
5	**0.8878**
6	**0.5566**
7	0.6447
9	0.6257
10	0.8255
11	0.5755
13	0.7891

Table 4.12 demonstrates the advantage of our weighted subspace clustering. The attribute-based location (not graph location) of the node within the cluster is shown in Figure 4.12. The smallest weight of 0.5566 is given to Node 6 since it is farthest from the center of the cluster. The greatest weight of 0.8878 is given to Node 5 since it is nearest to the center of the cluster. Based on the subspace clustering result given, Nodes 8, 12, and 14 have rank 1 for outlierness. This ranking is because these nodes do not belong to any cluster in this given example, as seen in Figure 4.12 and Table 4.11.

Table 4.12: Comparison of outlier score and ranking based only on subspace clustering attribute term for 14 nodes of artificial dataset 2

Node	GOutRank score	GOutRank rank	Proposed score	Proposed rank
1	1	4	1.38	11
2	1	4	1.23	7
3	1	4	1.27	9
4	1	4	1.19	6
5	**1**	**4**	**1.78**	**14**
6	**1**	**4**	**1.11**	**4**
7	1	4	1.29	10
8	0	1	0.00	1
9	1	4	1.25	8
10	1	4	1.65	13
11	1	4	1.15	5
12	0	1	0.00	1
13	1	4	1.58	12
14	0	1	0.00	1

After those outlier nodes, the existing score function shown in Equation 4.2 is not able to distinguish outlierness of the remaining nodes. Notice how Nodes 5 and 6 have the same outlier rank in the existing rank column of Table 4.12 (bolded). In contrast, in the Proposed rank column, Node 6 is the most outlier among nodes that are in cluster C_1, and Node 5 is the least outlier among all nodes in the attributed PCN.

4.5.4 Artificial dataset 3: 14-node patent citation network

In this section we present our results for artificial dataset 3, which demonstrates the advantages of our graph-based scoring presented in Section 4.4.3. We present our outlier ranking results alongside the ranking results of the graph-based approach from GOutRank, and show how our approach outperforms outlier scoring and ranking from existing outlier ranking algorithms. The major drawback of existing graph-based approaches is that they do not consider indirect link relationships, which contain important information in patent citation networks. We demonstrate how our methods presented in Section 4.4.3 use the co-citation similarity measure to achieve better outlier ranking results. Table 4.13 and Figure 4.15 show the node outlier ranking for both existing and proposed approaches. Notice that the existing approach describe in

Equation 4.2 cannot distinguish the outlier rank of Nodes 1 and 6 using graph structure, while our proposed approach described in Section 4.4.3 can greatly distinguish outlierness of those nodes.

Table 4.13: Comparison of outlier ranking based only on graph structure term for 14 nodes of artificial dataset 3

Node	GOutRank rank	Proposed rank
1	**1**	**13**
2	3	14
3	3	7
4	14	12
5	3	2
6	**1**	**1**
7	10	8
8	10	8
9	10	8
10	10	8
11	3	2
12	3	2
13	3	2
14	3	2

We discuss the general expected outlier ranking of nodes in artificial dataset 3 using the graph structure. The value of this example network is that we can systematically determine how we expect nodes in the network to be ranked in outlierness. As an example, we would not expect Node 4 to be an outlier in this network since it is cited by four other nodes (the most citations in this network) and it cites two other nodes (tied for the most in this network). We present the nodes we expect to be outliers for this graph, and those we expect to not be outlier in this graph. Node 6 should have the highest outlier rank since it cites only one other patent, and is not itself cited by any patent. We expect Node 1 to have a low outlier rank since it is co-cited with Node 2 at level-0, level-1, and level-2, meaning the two patents are very similar, based on graph structure. Also, we expect Node 2 to have a low outlier rank since it is most co-cited throughout the PCN, indicating high similarity or relatedness to other nodes in the network.

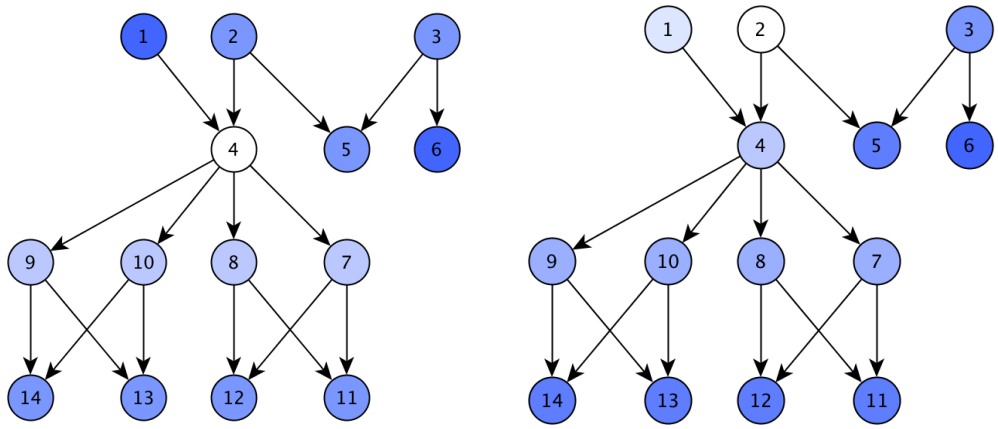

Figure 4.15: Artificial dataset 3 - example patent citation network and graph-based outlier ranking results for existing (left) and proposed (right) methods, where darker shade indicates greater outlierness

4.5.5 Real-life patent citation network

For these results, we use our combined contributions to outlier ranking in patent citation networks presented in earlier sections, and summarized in Equation 4.7 and Figure 4.9. For this experiment, we take 150 patents as the nodes in the patent citation network. In order to have a single connected tree structure to which to apply outlier detection method, we select the patents that cite, either directly or indirectly, the most cited patent from the original dataset, which is patent US-5349655. Note we included one patent that was not connected to the rest of the network for experimental reasons. Our patent citation network then consist of 150 nodes and 215 edges. Table 4.14 shows the top 5 outlier patents from the dataset based on our proposed contributions. Additionally, top outliers are shown in red in Figure 4.16. Outlier nodes are correctly characterized by their minimal connection to the rest of the network. Additionally, outlier nodes are characterized by being minimally clustered in the subspace clustering result.

Our approach identified patent US-6216183 as the top outlier. This patent concerns securing information entered on an input device, which is coupled to a universal serial bus (USB). The outlier rank is justified as this patent is actually not connect to the rest of the network by a citation, and is used as a control for the real-life dataset.

Table 4.14: Top five ranked outlier patents from the real-life PCN dataset

Outlier rank	Patent	Class codes
1	US-6216183	710/100; 711/163; 726/18
2	US-5930767	705/26.41; 705/27.2; 705/77
3	US-6026193	382/232; 348/473; 380/202; 386/E5.004; 704/E19.009; G9B/20.002
4	US-6038564	707/702; 707/966; 707/999.01; 707/999.2; 707/E17.032
5	US-6041412	726/3; 713/180; 713/186

A patent that is not connected to the rest of the dataset may indicate the entrance of a technology from a new area. The second ranked outlier patent is US-5930767, which concerns transaction methods, systems, and devices. This patent is found to be an outlier because it contains only three subclasses in a single class, 705, indicating that it is a narrow, specialized technology as compared to the other patents in the patent dataset. The third ranked outlier patent is US-6026193, which relates to video steganography. This patent has an unusual combination of class codes: 380, 382, and 386, which concern cryptography, image analysis, and motion video signal processing for recording or reproducing, respectively. This is the only patent in this dataset to have this combination of three class codes. The fourth ranked outlier patent is US-6038564, which concerns a method and apparatus for integrating distributed information. This patent contains five subclasses within class 707, again indicating that it is a specialized technology. This patent deals with programs for ensuring data integrity that is stored distributively in multiple processing devices. Finally, the fifth ranked outlier patent is US-6041412, which deals with apparatus and method for providing access to secured data or area. This patent is minimally connected to the PCN as it makes one citation, and is not itself cited, thus its identification as an outlier.

4.6 Conclusion and future work

In this work we present a new subspace clustering algorithm and new node outlier ranking methods that leverage both node attribute data and graph structure data found in attributed patent citation networks. The objective of this research is to develop advanced methods for outlier ranking geared specifically towards patent citation

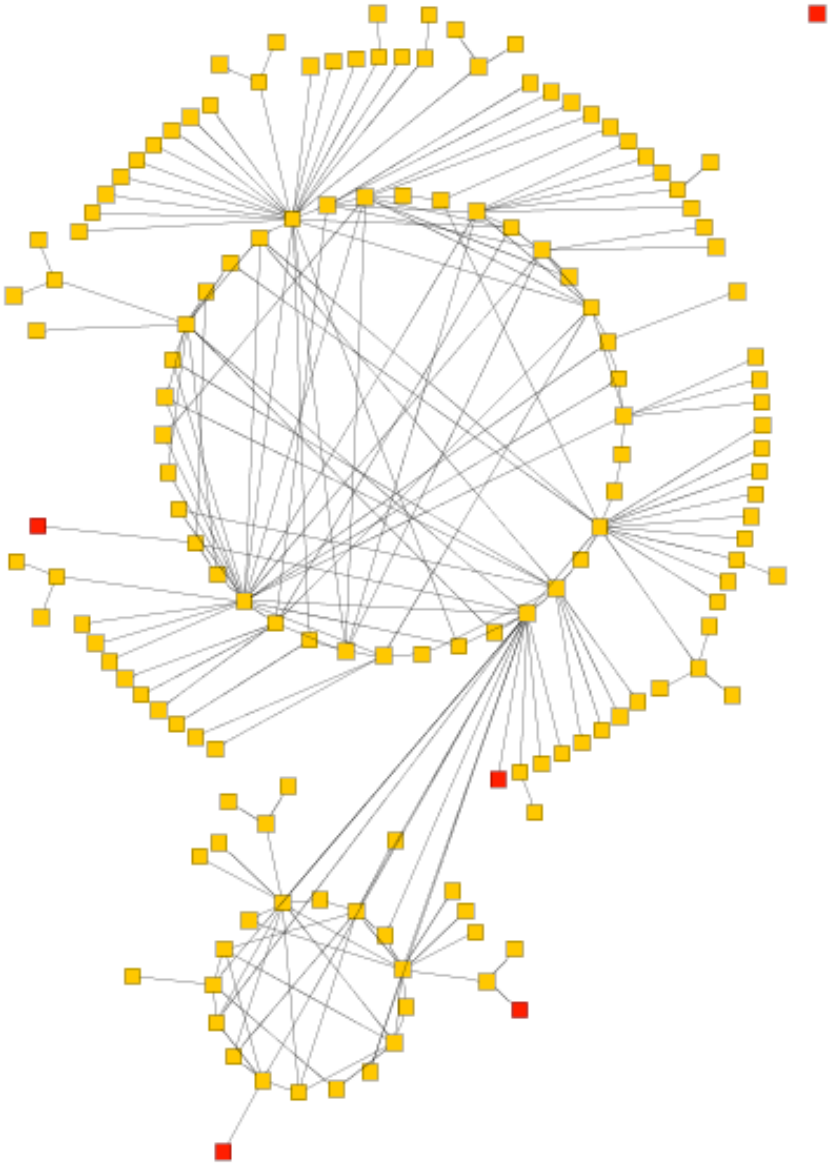

Figure 4.16: Real-life patent citation network outlier ranking results using score function that combines cluster-based and graph-based methods, where red nodes indicate patent outliers

networks. To this end, we presented patent outlier ranking methods based on citation graph structure considering subspace clusters to which a node belongs and based on graph structure, leveraging the multi-stage co-citation similarity measure and node centrality. Additionally, we are able to distinguish the outlierness of nodes belonging to the same cluster by considering the distance of a node to the center of the cluster to which it belongs, which was a major drawback of existing subspace clustering approaches.

This work is significant since, to the best of our knowledge, it is the first measure of its type developed specifically for patent citation networks, and the characteristics of that type of data. Experimental results show our approaches outperform other state-of-the-art approaches. It may be utilized for detection of outlier patents in patent citation networks that may be extended for identification of technology opportunities. Although we applied the approaches to patent citation networks, the developed methods may be applied to other types of attributed graph data data where the attribute data is sparse, and the co-citation relationship is meaningful.

One possible future work includes building on the subspace clustering-based models to consider categorical data. In this work, we focused on numerical values for class code attributes, but patent data also contains categorical data, which may further be leveraged to identify outliers. A second possible future work builds on graph-based models. In the background section of this work, regarding graph-based outlier approaches, we reviewed works that identify outliers by finding patents that do not belong to any cluster. As stated in [82], a challenge of this approach is often how to define the neighborhoods based on network structure. We may leverage the co-citation similarity matrix that we developed previously in order to help in defining the network structure. Rather than simply using common neighbors, we may leverage the rich multi-stage co-citation based similarity scores to construct an idea of a *logical neighborhood*. A third possible future work is to rank outliers in a time-evolving PCN, rather than a static attributed PCN. In this way, we can consider the the rate at which direct and indirect links are added.

Chapter 5

Concluding Remarks and Future Research

5.1 Concluding remarks

In this dissertation, we have proposed and developed several methodologies for patent influence measures, similarity measures, and outlier detection in patent citation networks. In Chapter 2, we proposed a measure of patent influence that leverages the powerful graph kernels. The new centrality measure is based on the change of the node similarity matrix after leveraging graph kernels. The proposed approach provides a more robust understanding of the identification of influential nodes, since it focuses on graph structure information by considering direct and indirect patent citations. This study leverages the concept that the change of similarity matrix that results from removing a given node indicates the importance of the node within its network, since each node makes a contribution to the similarity matrix among nodes. We calculate the change of the similarity matrix norms for a given node after we calculate the singular values for the case of the existence and the case of nonexistence of that node within in the network. Then, the node resulting in the largest change (*i.e.*, decrease) in the similarity matrix norm is considered to be the most influential node. We compare the performance of our proposed approach with other widely-used centrality measures using artificial data and real-life U.S. patent data. Experimental results show that our proposed approach performs better than existing methods, and provides robustness that existing approaches do not.

In Chapter 3, we moved from the analysis of citation to the analysis of co-citations, and we proposed a similarity measure between patents in a patent citation network using only the graph structure. In the past, techniques such as text mining and keyword

analysis have been applied for patent similarity calculation. The drawback of these approaches is that they depend on word choice and writing style of authors. In this work we propose two new similarity measures for patents in a patent citation network using only the patent citation network structure. The first proposed similarity measure uses co-citation links. The second developed similarity measure uses bibliographic coupling links. A challenge is when some patents are involved in a disproportionately large number of citations, thus are considered more similar to many other patents in the patent citation network. To overcome this challenge, we develop a normalization technique to account for the case where some pairs are ranked very similar to each other because they both are cited by, or cite, many other patents. We validate our proposed similarity measures using U.S. class codes for U.S. patents and the well-known Jacquard similarity index. Experiments show that the proposed methods perform well when compared to the Jaccard similarity index and considering Spearman correlation coefficient.

In Chapter 4, we proposed methods for ranking outlier patents and patent citations in a patent citation network. There is currently no work on patent citation network anomaly detection that considers attributes in patent citation networks, in addition to the citation network structure. Patent citation networks have the distinguishing characteristic that each patent is given carefully assigned attributes, such as classification codes at the time of the patent creation. We first develop a subspace clustering algorithm to cluster patents. Then we propose patent outlier score functions. One score function using the subspace clustering result, while the other score function uses graph-based measures, including those developed in Chapter 3. Experiments using artificial datasets show that the proposed methods outperform existing approaches, when applied to patent citation network data.

5.2 Future research

For patent influence measure research, potential future research opportunities include (1) considering assigning a weight of zero for indirect citations greater than some length c, so that older patents do not have the advantage of having many long indirect

citation paths that contribute to their influence score and (2) considering time information of a patent to account for the rate of citations made (*i.e.*, consider how the citation network evolves with time).

For patent similarity measure research, a meaningful integration of co-citation and bibliographic coupling similarity measures would be a possible extension.

Future studies on anomaly detection or patent outlier ranking are needed. One possible future work includes building on the subspace clustering-based models to consider categorical data. Additionally, little research has been done on validating the results of anomalous or outlier patents. A meaningful line of research would be to develop anomaly or outlier ranking validation methods, which are applicable to attributed graphs. Moreover, we can extend methods to consider additional attribute data, such as temporal data, in order to study the evolving patent citation network. Another future work would be to apply outlier detection approaches to other types of attributed graph data.

Appendix A

Proof of Proposition 1

Let $\mathbf{A}$ be the $n \times n$ adjacency matrix where a_{ij} is 1 if node i cites node j, and 0 otherwise. We know that the sum of the squared singular values σ_{ii} is equal to the sum of the squared values of the matrix, over all (i, j) pairs. The sum of squares of singular values for $\mathbf{A}$ is the number of 1s in the adjacency matrix since $\mathbf{A}$ is a {0,1} matrix and $1^2 = 1$ and $0^2 = 0$, so we have:

$$\begin{aligned}
\sum_{i=1}^{n}(\sigma_{ii})^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}(a_{ii})^2 \\
&= \sum_{i=1}^{n}\sum_{j=1\neq t}^{n}(a_{ii}) \\
&= \text{number of 1s in } \mathbf{A}.
\end{aligned}$$

This means that the difference of sum of the squares of the singular values between the case of the existence and the case of nonexistence of each node is:

$$\begin{aligned}
p_2(t) &= \sum_{i=1}^{n}(\sigma_{ii})^2 - \sum_{i=1}^{n}(\sigma_{(-t)_{ii}})^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}(a_{ii})^2 - \sum_{i=1}^{n}\sum_{j=1\neq t}^{n}(a_{ii})^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}(a_{ii}) - \sum_{i=1}^{n}\sum_{j=1\neq t}^{n}(a_{ii}) \\
&= \text{number of 1s in row } t \\
&= \text{out-degree of node } t.
\end{aligned}$$

And we have shown that using RR SVC scores starting with just the adjacency matrix, $\mathbf{A}$, is equal to out-degree centrality score.

Appendix B

Proof of Proposition 2

Recall that $\mathbf{K}_{VN} = \sum_{k=0}^{\infty} \alpha^k \mathbf{A}^k = (\mathbf{I} - \alpha\mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \ldots$. Let a_{ij}^k be the a_{ij} element of $\mathbf{A}^k$. Then,

$$e(t) \quad = \quad \mathbf{K} - \mathbf{K}_{(-t)}.$$

$$\begin{aligned} e(t) - 1 &= \sum_{i=1}^{n}\sum_{j=1}^{n}(a_{ii}) - \sum_{i=1}^{n}\sum_{j=1\neq t}^{n}(a_{ii}) \\ &= \text{number of 1s in row } t \text{ of } \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \ldots \\ &= \text{direct and indirect out-degree of node } t. \end{aligned}$$

where a_{ij}^k is the a_{ij} element of $\mathbf{A}^k$. And we have shown that one less than the centrality score using the graph kernel-based SVC with von Neumann kernel and parameter $\alpha = 1$, and using the entry-wise matrix norm, is equal to total number of paths of length 1 to m, for each node in the network.

References

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Advances in Knowledge Discovery and Data Mining*, pages 410–421. Springer, 2010.

[2] Michael B. Albert, Daniel Avery, Francis Narin, and Paul McAllister. Direct validation of citation counts as indicators of industrially important patents. *Research policy*, 20(3):251–259, 1991.

[3] Diego R. Amancio, Osvaldo N. Oliveira Jr., and Luciano da F. Costa. Structure–semantics interplay in complex networks and its effects on the predictability of similarity in texts. *Physica A: Statistical Mechanics and its Applications*, 391(18):4406 – 4419, 2012.

[4] Diego R Amancio, Osvaldo N Oliveira Jr, and Luciano da F. Costa. On the use of topological features and hierarchical characterization for disambiguating names in collaborative networks. *EPL (Europhysics Letters)*, 99(4):48002, 2012.

[5] Ery Arias-Castro, Emmanuel J. Candes, Arnaud Durand, et al. Detection of an anomalous cluster in a network. *The Annals of Statistics*, 39(1):278–304, 2011.

[6] Gamal Atallah and Gabriel Rodriguez. Indirect patent citations. *Scientometrics*, 67(3):437–465, 2006.

[7] Vic Barnett and Toby Lewis. *Outliers in statistical data*, volume 3. Wiley New York, 1994.

[8] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[9] Stephen P. Borgatti. Identifying sets of key players in a social network. *Computational and Mathematical Organizational Theory*, 12(21–34), 2006.

[10] Matthew Brand. Fast online svd revisions for lightweight recommender systems. In *SIAM Third International Conference on Data Mining*, pages 37–46, March 2003.

[11] Stefano Breschi, Francesco Lissoni, and Franco Malerba. Knowledge-relatedness in firm technological diversification. *Research Policy*, 32(1):69–87, 2003.

[12] Gaetano Cascini and Manuel Zini. Measuring patent similarity by comparing inventions functional trees. In Gaetano Cascini, editor, *Computer-Aided Innovation (CAI)*, volume 277 of *The International Federation for Information Processing*, pages 31–42. Springer US, 2008.

[13] Deepayan Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *Knowledge Discovery in Databases: PKDD 2004*, pages 112–124. Springer, 2004.

[14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[15] Duen Horng Chau, Shashank Pandit, and Christos Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. In *Knowledge Discovery in Databases: PKDD 2006*, pages 103–114. Springer, 2006.

[16] Daniel J. Cook and Lawrence B. Holder. *Mining Graph Data*. Wiley-Interscience, 2006.

[17] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[18] William Eberle and Lawrence Holder. Anomaly detection in data represented as graphs. *Intelligent Data Analysis*, 11(6):663–689, 2007.

[19] Leo Egghe and Ronald Rousseau. Introduction to informetrics: Quantitative methods in library, documentation and information science. 1990.

[20] Leo Egghe and Ronald Rousseau. Co-citation, bibliographic coupling and a characterization of lattice citation networks. *Scientometrics*, 55(3):349–361, 2002.

[21] P. Ellis, G. Hepburn, and C. Oppenhein. Studies on patent citation networks. *Journal of Documentation*, 34(1):12–20, 1978.

[22] Ernesto Estrada, Desmond J. Higham, and Naomichi Hatano. Communicability betweenness in complex networks. *Physica A: Statistical Mechanics and its Applications*, 388(5):764–774, 2009.

[23] Francois Fouss, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *Sixth International Conference on Data Mining*, pages 863–868, December 2006.

[24] Linton C. Freeman. Centrality in social network: Conceptual clarification. *Social Networks*, 1:215–239, 1978.

[25] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. On community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 813–822. ACM, 2010.

[26] Devi R. Gnyawali and Byung-Jin Robert Park. Co-opetition between giants: Collaboration with competitors for technological innovation. *Research Policy*, 40(5):650–663, 2011.

[27] Bernard Gress. Properties of the uspto patent citation network: 1963–2002. *World Patent Information*, 32(1):3–21, 2010.

[28] Stanislao Gualdi, Matúš Medo, and Y-C Zhang. Influence, originality and similarity in directed acyclic graphs. *EPL (Europhysics Letters)*, 96(1):18004, 2011.

[29] Stephan Gunnemann, Ines Farber, Brigitte Boden, and Thomas Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 845–850. IEEE, 2010.

[30] Victoria J. Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:2004, 2004.

[31] Lawrence B. Holder and Diane J. Cook. Graph-based data mining. *Encyclopedia of data warehousing and mining*, 2:943–949, 2009.

[32] Shiu-Wan Hung and An-Pang Wang. Examining the small world phenomenon in the patent citation network: a case study of the radio frequency identification (rfid) network. *Scientometrics*, 82(1):121–134, 2010.

[33] Dean L. Isaacson and Richard W. Madsen. *Markov chains, theory and applications*, volume 4. Wiley New York, 1976.

[34] Si Hyung Joo and Yeonbae Kim. Measuring relatedness between technological fields. *Scientometrics*, 83(2):435–454, 2010.

[35] Jaz Kandola, John Shawe-Taylor, and Nello Cristianini. Learning semantic similarity. In *Proceedings of the Neural Information Processing Systems*, pages 657–664. MIT Press, 2003.

[36] U Kang, Leman Akoglu, and Duen Horng Polo Chau. Big graph mining: Algorithms, anomaly detection, and applications. *Proceedings of the ACM ASONAM*, 13:25–28, 2013.

[37] Maxwell Mirton Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25, 1963.

[38] Byunghoon Kim, Gianluca Gazzola, Jae-Min Lee, Dohyun Kim, Kanghoe Kim, and Myong K. Jeong. Inter-cluster connectivity analysis for technology opportunity discovery. *Scientometrics*, 98:1811–1825, 2014.

[39] Chulhyun Kim and Hyeonju Seol. On a patent analysis method for identifying core technologies. In Junzo Watada, Toyohide Watanabe, Gloria Phillips-Wren, Robert J. Howlett, and Lakhmi C. Jain, editors, *Intelligent Decision Technologies*, volume 2, pages 441–448. Springer Berlin Heidelberg, 2012.

[40] Dohyun Kim, Bangrae Lee, Hyuck Jai Lee, Sang Pil Lee, Yeongho Moon, and Myong K. Jeong. Automated detection of influential patents using singular values. *IEEE Transactions on Automation Science and Engineering*, 9(4):723–733, October 2012.

[41] Euiseok Kim, Yongrae Cho, and Wonjoon Kim. Dynamic patterns of technological convergence in printed electronics technologies: patent citation network. *Scientometrics*, 98(2):975–998, 2014.

[42] Young Gil Kim, Jong Hwan Suh, and Sang Chan Park. Visualization of patent analysis for emerging technology. *Expert Systems with Applications*, 34(3):1804 – 1812, 2008.

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

[44] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[45] Ohjin Kwon, Bangrae Lee, Jinny Seo, Kyeongran Noh, Jinjoo Lee, and Jinsuk Kim. A method to make the genealogical graph of core documents from the directed citation network. *INFORMATION-An International Interdiscplinary Journal*, 12(4):875–888, July 2009.

[46] Leah S. Larkey. A patent search and classification system. In *Proceedings of DL-99, 4th ACM Conference on Digital Libraries*, pages 179–187. ACM Press, 1999.

[47] Yan Lin, Jian Chen, and Yan Chen. Backbone of technology evolution in the modern era automobile industry: An analysis by the patents citation network. *Journal of Systems Science and Systems Engineering*, 20(4):416–442, 2011.

[48] Koji Maruhashi and Christos Faloutsos. Eigendiagnostics: Spotting connection patterns and outliers in large graphs. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1328–1337, Dec 2010.

[49] Bai Meng, Hu Ke, and Tang Yi. Link prediction based on a semi-local similarity index. *Chinese Physics B*, 20(12):128902, 2011.

[50] Martin Meyer. What is special about patent citations? differences between scientific and patent citations. *Scientometrics*, 49(1):93–123, 2000.

[51] Jacques Michel and Bernd Bettels. Patent citation analysis. a closer look at the basic input data from patent search reports. *Scientometrics*, 51(1):185–201, 2001.

[52] Martin G. Moehrle and Jan M. Gerken. Measuring textual patent similarity on the basis of combined concepts: design decisions and their consequences. *Scientometrics*, 91(3):805–826, 2012.

[53] Martin G. Moehrle, Lothar Walter, Anja Geritz, and Sandra Müller. Patent-based inventor profiles as a basis for human resource decisions in research and development. *R&D Management*, 35(5):513–524, 2005.

[54] Mary Ellen Mogee and Richard G Kolar. International patent analysis as a tool for corporate technology analysis and planning: Practitioners forum. *Technology Analysis & Strategic Management*, 6(4):485–504, 1994.

[55] H.D.K. Moonesinghe and Pang-Ning Tan. Outlier detection using random walks. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pages 532–539. IEEE, 2006.

[56] Emmanuel Muller, Patricia Iglesias Sánchez, Yvonne Mulle, and Klemens Bohm. Ranking outlier nodes in subspaces of attributed graphs. In *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*, pages 216–222. IEEE, 2013.

[57] Ramasuri Narayanam and Yadati Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 8(1):130–147, January 2011.

[58] Francis Narin. Patent bibliometrics. *Scientometrics*, 30(1):147–155, 1994.

[59] Mark E. J. Newman. *Networks: An Introduction*. Oxford University Press, Inc., 2010.

[60] Hyun Joung No and Yongtae Park. Trajectory patterns of technology fusion: Trend analysis and taxonomical grouping in nanobiotechnology. *Technological forecasting and social change*, 77(1):63–75, 2010.

[61] Caleb C. Noble and Diane J. Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2003.

[62] Jae D. Noh and Heiko Rieger. Random walks on complex networks. *Physical Review Letters*, 92, 2004.

[63] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, July 2010.

[64] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[65] Robert Piziak and Patrick L Odell. *Matrix theory: from generalized inverses to Jordan form*, volume 288. CRC Press, 2007.

[66] Stephen Ranshous, Shitian Shen, Danai Koutra, Christos Faloutsos, and Nagiza F. Samatova. Anomaly detection in dynamic networks: A survey. Technical report, Technical Report-Not held in TRLN member libraries, 2014.

[67] Andrew Rodriguez, Byunghoon Kim, Jae-Min Lee, Byoung-Yul Coh, and Myong K. Jeong. Graph kernel based measure for evaluating the influence of patents in a patent citation network. *Expert Systems with Applications*, 42(3):1479–1486, 2015.

[68] Andrew Rodriguez, Byunghoon Kim, Jae-Min Lee, Byung Y. Coh, and Myong K. Jeong. Graph kernel based centrality measure for evaluating the importance of patents in a patent citation network. *Technical Report*, 2014.

[69] Andrew Rodriguez, Byunghoon Kim, Mehmet Turkoz, Jae-Min Lee, Byoung-Youl Coh, and Myong K. Jeong. New multi-stage similarity measure for calculation of pairwise patent similarity in a patent citation network. *Scientometrics*, 102(3):1–17, 2015.

[70] Ronald Rousseau. The gozinto theorem: Using citations to determine influences on a scientific publication. *Scientometrics*, 11(3–4):217–229, 1987.

[71] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*. Addison-Wesley, 1989.

[72] Vasilios A. Siris and Fotini Papagalou. Application of anomaly detection algorithms for detecting syn flooding attacks. *Computer communications*, 29(9):1433–1442, 2006.

[73] Henry Small. Co-citation in the scientific literature: a new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24:265–269, 1973.

[74] Ashish Sood and Gerard J. Tellis. Technological evolution and radical innovation. *Journal of Marketing*, 69:152–168, July 2005.

[75] Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[76] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[77] Yuen-Hsien Tseng, Chi-Jen Lin, and Yu-I Lin. Text mining techniques for patent analysis. *Information Processing & Management*, 43(5):1216–1247, 2007.

[78] USPTO. Us patent full-text database number search, http://patft.uspto.gov/netahtml/pto/srchnum.htm, 2014.

[79] Maarten van Steen. *Graph Theory and Complex Networks: An Introduction*. Maarten van Steen, 2010.

[80] Iwan von Wartburg, Thorsten Teichert, and Katja Rost. Inventive progress measured by multi-stage patent citation analysis. *Research Policy*, 34:1591– 1607, 2005.

[81] Hsiao-Chun Wu, Hung-Yi Chen, Kung-Yen Lee, and Ying-Chieh Liu. A method for assessing patent similarity using direct and indirect citation links. In *Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on*, pages 149–152, 2010.

[82] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824–833. ACM, 2007.

[83] Byungun Yoon and Yongtae Park. A text-mining-based patent network: Analytical tool for high-technology trend. *The Journal of High Technology Management Research*, 15(1):37–50, 2004.

[84] Janghyeok Yoon and Kwangsoo Kim. Detecting signals of new technological opportunities using semantic patent analysis and outlier detection. *Scientometrics*, 90(2):445–461, 2012.