The Academic Social Network and Research Ranking System

FU, Zhengjia

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
September 2013

Abstract of thesis entitled:

The Academic Social Network and Research Ranking System
Submitted by FU, Zhengjia
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in September 2013

Through academic publications, the authors of these publications form a social network. Instead of sharing casual thoughts and photos (as in Facebook), authors pick co-authors and reference papers written by other authors. Thanks to various efforts (such as Microsoft Libra, DBLP and APS), the data necessary for analyzing the academic social network is becoming more available on the Internet. What type of information and queries would be useful for users to find out, beyond the search queries already available from services such as Google Scholar? In this thesis, we explore this question by defining a variety of ranking metrics on different entities - authors, publication venues and institutions. We go beyond traditional metrics such as paper counts, citations and h-index. Specifically, we define metrics such as *influence*, *connections* and *exposure* for authors. An author gains influence by receiving more citations, but also citations from influential authors. An author increases his/her connections by co-authoring with other authors, and specially from other authors with high connections. An author receives exposure by publishing in selective venues where publications received high citations in the past, and the selectivity of these venues also depends on the influence of the authors who publish there. We discuss the computation aspects of these met-

rics, and similarity between different metrics. With additional information of author-institution relationships, we are able to study institution rankings based on the corresponding authors' rankings for each type of metric as well as different domains. We are prepared to demonstrate these ideas with a web site (`http://pubstat.org`) built from millions of publications and authors.

Another common challenge in bibliometrics studies is how to deal with incorrect or incomplete data. Given a large volume of data, however, there often exists certain relationships between data items that allow us to recover missing data items and correct erroneous data. In the latter part of the thesis, we study a particular problem of this sort - estimating the missing year information associated with publications (and hence authors' years of active publication). We first propose a simple algorithm that only makes use of the "direct" information, such as paper citation/reference relationships or paper-author relationships. The result of this simple algorithm is used as a benchmark for comparison. Our goal is to develop algorithms that increase both the coverage (the percentage of missing year papers recovered) and accuracy (mean absolute error of the estimated year to the real year). We propose some advanced algorithms that extend inference by information propagation. For each algorithm, we propose three versions according to the given academic social network type: a) Homogeneous (only contains paper citation links), b) Bipartite (only contains paper-author relations), and, c) Heterogeneous (both paper citation and paper-author relations). We carry out experiments on the three public data sets (Microsoft Libra, DBLP and APS), and evaluated by applying the K-fold cross validation method. We show that the advanced algorithms can improve both coverage and accuracy.

中文摘要

通过发表学术论文，论文作者们构成了一个学术社交网络（Academic Social Network）。不同于在脸书（Facebook）这样的社交网络中随意的交换想法和分享照片，作者们在学术社交网络中，只是选择合作者以及其他作者写的论文作为参考文献。受益于诸如（Microsoft Libra，DBLP和APS等）所作的各种努力，从网络中获取用来分析学术社交网络所必须的数据集合，正在变得越来越容易。除了像谷歌学术（Google Scholar）已经提供的学术类查询服务之外，什么类型的信息和查询是对用户最有用的？在本论文中，我们将通过定义和研究多种多样的、针对不同对象的（对象包括作者、论文、发表刊物和学术机构等）排名指标来探讨这个问题。我们将超越传统的诸如：论文数量、被引用次数和H因子（h-index）等指标。特别的，我们针对作者提出像影响力（influence）、社交性（connections）以及曝光度（exposure）这样的排名指标。一个作者一方面会因其论文被引用次数的增加而提升影响力；另一方面，也可以因为被有影响力的人所引用而提升影响力。类似的，一个作者的社交性会随着和其他作者合作次数的增多而提高，也同样可以通过和社交性很高的作者合作来提高。一个作者如果在比较有影响力的刊物上发表论文，那么其曝光度会得到增加，同时，影响刊物的（影响力）也正式那些论文的作者的影响力本身，这两者是相辅相成的。我们的讨论涉及这些指标的计算方法，以及指标间的相似程度。当我们能够获得额外的数据，例如作者所属的学术机构信息的时候，我们还能利用针对作者产生的各个指标的排名结果来对学术机构进行排名。基于一个非常大数据量的作者和论文的数据集合，我们准备将我们的这些想法，设计的指标和最终排名的结果，展示在以下这个网站上（http://pubstat.org）。

在学术社交网络和文献计量方面，另一个公认的具有挑战性的问题，便是如何处理包含错误的或者不完整的数据。对于一个数量比较庞大的数据集合，通常，数据与数据之间本身存在着一定的内在联系。这给我们提供了一个解决问题的方向。在论文的后半部分，我们将研究这类问题中的一个特殊问题，即恢复缺失的论文发表的年份信息（同样也涉及到作者的学术活跃时间段的信息）。我们首先提出了一个简单算法，这个算法只利用“直接（可见）”的信息，例如论文本身引用过的论文和被引用的论文信息，论文的作者信息等。这个简单算法得到的结果将用来作为性能评估的参考。我们的目标是能够设计出既能提高“恢复率”（能够被恢复的缺失年份信息的论文的比例），又能提高“准确度”（恢复出的年份和论文真实发表年份的差别）的算法。之后，我们又提出了进阶的算法，这个算法引入了信息传播的机制。对每一个算法，针对三种不同的学术社交网络的类型，我们都提出了相应的版本。三种学术社交网络的类型分别是：1、同构图（只包含论文以及论文之间的引用信息）；2、二分图（只包含论文和作者之间的归属信息）；3、异构图（既包含论文和论文引用的信息，又包含论文和作者的归属关系）。我们使用了三种公布在网上的数据集合（Microsoft Libra，DBLP 和 APS）来进行实验，并运用 K-fold 交叉验证法来对算法进行性能评估。结果表明，相对于基本算法，进阶算法能同时提高“恢复率”和“准确度”。

Acknowledgement

I would like to thank my supervisor Prof. Dah Ming Chiu for his great help in my PhD study. He is a nice and kind mentor. I have learned a lot from him, not only the ways of doing research and solving problems, but also the rigorous scientific attitudes one shall have.

I would like to express my sincere thanks to my "Gui Ren"s, Mr. Ke Ma and Dr. Richard T. B. Ma. They are very important people in my life, always bringing me good opportunities and good lucks at the key time.

I would like to thank Mr. Changmin Chen, who has given me great help in studying and using CSharp and .Net Framework. He has also contributed to some of the projects.

Over the past few years, I have had a lot of collaborative works with Dr. Yipeng Zhou. It is a nice experience and I would like to thank him for his help and patience.

I would like to thank Ms Qianqian Song and Qiufang Ying for their help in the work of academic research ranking problem and the missing year estimation problem, respectively.

I would like to thank all my friends for their understanding and fully support, especially Vivian Zhang, Jiaying Wang, Victor Liang, Jing Pan, Chichen Ling and etc.

Last but not least, family is always my backup at anytime, anywhere. I would like to express my thanks to all my family members for their attention.

To my lovely wife, Ruyu,
and our great parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Summary

In this chapter, we give the introduction of the academic research ranking problem in Section 1.1. We introduce the background, the motivation and our contribution of the missing year estimation problem in Section 1.2. We finally discuss the organization of this thesis in Section 1.3.

1.1 The Academic Research Ranking

In the academic community, it is customary to get a quick impression of an author's research from simple statistics about his/her publications. Such statistics include paper count, citations of papers, h-index and various other indices for counting papers and citations. Several services, such as ISI [1], Scopus [2], Google Scholar [3], Microsoft Academic Search [4, 67], CiteSeerX [5, 56], DBLP [6, 54] and American Physical Society (APS) [7], facilitate the retrieval of these statistics by maintaining databases indexing the metadata of academic publications. These databases are usually proprietary and the information

users can retrieve, sometimes on a paid basis, is limited to what these services choose to provide.

In recent years, some of these service providers [4, 5, 6] are making the database more publically accessible and are starting to provide additional information users can query (this is specially the case with Libra). This allows us to study the author community as a social network, analyzing not only the statistics about papers published by an author, individually at a time, but also an author's choice and extent in *connecting* to other authors (co-authoring) and an author's *influence* on other authors. Since citation is a *slow* indicator for evaluating an author's standing, we can also design metrics to measure an author's *exposure* in her research community, to estimate his/her future influence and connections in research.

Our approach is to design various social network types of metrics to measure the traits defined above. Since there is no ground-truth for validation, we justify our designs by the following methods: (1) Compare top ranked authors to those receiving awards for qualities similar to what we try to measure, e.g. influence; (2) Use similarity study to ensure any new metric can measure something different from that is indicated by other well-established metrics already; (3) Undertake case-studies of those authors scoring very differently under different metrics, in domains we are familiar with; (4) Let colleagues use our experimental website (`http://pubstat.org`) and get their feedback on its usefulness.

Our conclusion is that several of the metrics we designed, namely *Influence*, *Connections* and *Exposure*, can provide different rankings of authors, and together with Citation Count can give a fuller picture about authors. According to the author ranking results, combined with additional information on author-institution relationships, we further study and design approaches for conducting author-based institution ranking for

each of the various metrics as well as the subject domains.

1.2 The Missing Year Estimation Problem

Academic publication analysis has always been of interest to the research community. Earlier focus includes citation analysis, and journal impact factor analysis, to help evaluate research impact. In recent years, there is increasing interest in the social aspects of research, for example there are studies of patterns of collaborations, automatically inferring advisor-advisee relationships, and finding or predicting leaders and rising stars in research areas.

To such research, *data cleaning*, which is how to deal with the lack of data, or when data is available its incorrectness and incompleteness, is a general problem. In academic social network analysis, there are quite a few known challenges, e.g., author name ambiguity, author-affiliation errors, missing publication data and so on.

For the author name ambiguity problem, there are two possible errors: (1) more than one real person share the same name, e.g. many authors from China share the same full name represented by Pinyin. It becomes even worse if name abbreviation is used. (2) one real person has multiple name representations and hence considered as sperate authors. For example in DBLP service [6], Dah Ming Chiu has two separate entities with names "Dah Ming Chiu" and "Dah-Ming W. Chiu", respectively.

The ambiguity problem also happens to the affiliation names, e.g., CUHK has several name representations: "The Chinese University of Hong Kong", "Chinese Univ. of HK", etc. This is one of the reasons causing the author-affiliation errors. Authors usually change their jobs and work in different affiliations. However, the author-affiliation information we have is not the most updated, which leads to author-affiliation errors.

The occurrence of the missing publication data (e.g., publication year, published venue) in the bibliographic data can be caused by a variety of reasons. We think one reason is the cited papers are also included in the dataset, even if the original source is not available. References are sometimes incomplete, leading to missing and erroneous data. It is also possible that some papers are recovered from scanned source, and it is hard to extract all attributes.

However, since the data volume is large, and there exists all kind of relationships between data items, it is often possible to recover certain missing (or correct erroneous) data items from the data we have. In this thesis, we study a particular problem of this sort - estimating the missing year information associated with publications (and authors' years of active publication).

In the recent KDD Cup 2013, the two challenges are the Author-Paper Identification Challenge and the Author Disambiguation Challenge. For both challenges, the publishing year information of each paper is important background knowledge for the design of algorithms. However, the given data set [8] has a high *Missing Year Ratio*, $\frac{155784}{2257249} \approx 6.90\%$ (there are totally 2257249 papers, and out of which, 155784 are missing year papers).This is an important motivation for developing algorithms to recover the missing year attribute of publications, we called the Missing Year Estimation (MYE) problem.

We first propose a simple algorithm that only makes use of the "direct" information, such as paper citation/reference relationships or paper-author relationships. The result of this simple algorithm is used as a benchmark for comparison. Our goal is to develop sophisticated algorithms that increase both the coverage (measured by the percentage of missing year papers recovered) and accuracy (mean absolute error, or MAE, of the estimated year to the real year). The more advanced algorithms we propose and study involve information propagation rules so that

information which is multiple hops away can also be utilized. For each algorithm, we propose three versions according to the given academic social network type: a) Homogenous (only contains paper citation links), b) Bipartite (only contains paper-author relations), and, c) Heterogeneous (both paper citation and paper-author relations). We carry out experiments on the three public data sets (Microsoft Libra, DBLP and APS), by applying the K-fold cross validation method.

Our contributions are: we formulate the problem and introduce a basic (benchmark) algorithm that can already recover most of the missing years if both citation and author information are available. We then systematically developed improved algorithms based on methods in machine learning. These advanced algorithms further improve both coverage and accuracy (around 20% in the paper citation network, 8% in paper author bipartite network and heterogeneous network), over the benchmark algorithm. In addition, the coverage achieved by the advanced algorithms well matches the results of the analytical model.

1.3 Organization

The rest of the thesis is organized as follows. In Chapter 2, we discuss the research background and related works. In Chapter 3, we describe the definitions of the academic social network (ASN) and the various metrics for research ranking. In Chapter 4, we discuss the design, implementation and evaluation of the academic research rankings. In Chapter 5, we introduce the algorithms we designed for MYE problem in three different network types and the performance evaluation results by real data experiments. Finally we make the conclusion and discuss the future work in Chapter 6.

□ **End of chapter.**

Chapter 2

Background and Related Works

Summary

In this chapter, we discuss the background and related works.

The study of academic publication statistics is by no means a new topic. Previous attention focused mostly in different areas of science, especially physics. The most influential work was published in 1965 by Derek [34], in which he considered papers and citations as a network and noticed the citation distribution (degree distribution) followed the power law. A few years later, he tried to explain this phenomenon using a simple model called the *cumulative advantage* process [35, 59]. The skewness of the citation count distribution has since been validated by other studies on large scale datasets [69, 71]. In subsequent literature, later on, the model became better known as *preferential attachment* by [23] (i.e. a paper is more likely to cite another paper with more existing citations) and with good empirical evidence [49].

To determine the quality or *impact* of a paper by its citation count [42], while considered reasonable by many, has met with strong criticisms [62, 81]. Instead of using citation

count, it has been proposed that a ranking factor, calculated using the eigenvector-based methods such as PageRank [28, 53] or HITS [50], be adopted. Subsequently, a number of proposals of different variations to measure paper importance appeared, including eigenvector-based [76] or network traffic-like schemes [57, 80], or knowledge flow based [46]. Since it takes time for a paper to accumulate its share of citations, it is common practice to use the venue (journal) the paper is published in to predict the potential impact/importance of a paper. Thus, Journal Impact Factor (JIF) becomes an important indicator used in practice. The citation count of papers published in a journal within a certain time window is usually the basis for establishing the JIF of a journal [39]. In recent years, it is proposed to apply PageRank-like iterative eigenvector-based approach (or "Eigenfactor") to calculate the impact of journals [24, 25, 33].

For many years, the common practice to measure the impact and contribution of authors is based on simple measures such as paper count. The use of citation count has become more popular due to Google Scholar. A well-known example is the "Publish or Perish" tool [9]. More recently, some new indices, such as h-index [21, 47] and g-index [38] have been proposed to combine the use of citation count and paper count to measure the achievements of an author. Later on, some extensive works about h-index are further studied [22, 37]. Some more recent studies have also proposed to apply PageRank-type iterative algorithms to evaluate authors' contribution and impact, notably a scheme called SARA (Scientific Author Ranking Algorithm) to compute authors contributions [68]; and a model to rank both papers and authors [84]; and systems which are providing online web services, such as SCEAS (Scientific Collection Evaluator with Advanced Scoring) [72, 73] and Arnetminer (Academic Researcher Social Network Search)[10].

Besides the paper citations *earned* by authors, authors can

also be ranked based on their connections and popularity as a co-author. This way of evaluating authors is used in a series of studies by Newman *et al* on author collaboration networks [63, 64, 65, 66]. This approach and viewpoint is similar to that used in the study of social networks [36]. A number of recent papers studied social influence and their correlation to user actions [19, 20, 29, 32].

In network analysis, early studies focused on the structural characteristics of missing data, e.g., [52]. [27] studied the impact of the measurement errors on random Erdős-Rényi networks. A more recent work by [83] reclassifies measurement errors, separating missing data and false data, analyzes their efforts on different topology properties of an online social network and a publication citation network. But few works studies techniques to correct measurement errors.

Temporal information is frequently used in topics of academic network. [75] finds nearly all journals will reach a steady-state of citation distribution within a journal-specific time scale, thus proposed a model for the rank of paper impacts using citation counts. Publication time information is useful and considered in the research of academic ranking [44, 70]. To solve the tricky problem of name disambiguation in digital library, [77] utilizes the multi-hop co-author relationship and its special property of time-dependence. [82] proposes a time-constrained probabilistic factor graph model to mining the highly time-dependent advisor-advisee relationship on the collaboration network. [30] uses an iterative Belief Propagation Algorithm to identify malware from a large scale of files and machines. [85] studies the propagation of two or more competing labels on a graph, using semi-supervised learning methods.

The topic of evolution of communities also attracts much attention. [26] have used state space models on the natural parameters of the multinomial distributions to represent the dynamic

evolution of topics. [48] developed the continuous time dynamic model to mine the latent topics through a sequential collection of documents. [45] proposed an algorithm integrating clustering and evolution diagnosis of heterogeneous bibliographic information networks. [58]track the evolution of an arbitrary topic and reveal the latent diffusion paths of that topic in a social community. [55] addressed the community detection problem by integrating dynamics and communities into the topic modeling algorithms, and experimented on the Scholarly publications data set ArnetMiner [79].

Recently, data cleaning on academic social networks receives much attentions. In KDD Cup 2013, the two challenges are the Author-Paper Identification Challenge or the Author Disambiguation Challenge. For both challenges, the publishing year information of each paper is important background knowledge and affecting the design of the algorithms. However, the given data set [8] has a high *Missing Year Ratio*, $\eta = \frac{155784}{2257249} \approx 6.90\%$. This is one of the practical examples and usages which implies the importance of the MYE problems and a good motivation of this work.

Finally, the publication database plays a critical role in such bibliometrics and social network studies [11]. The well-known databases are: Google Scholar [3], Scopus [2], ISI [1], CiteSeerX [5, 56], Microsoft Libra [4, 67], DBLP [6, 54], or DBLP-Cit [12], which is created by a third party based on the original DBLP paper set with adding paper citation relationships through proper mining method [78, 79], American Physical Society (APS) [7], arXiv [13], IEEE [14], and ACM [15]. These databases, however, tend to contain different papersets. For example, CiteSeerX, DBLP, ACM focus mostly on computer science and related literature, but each has its own rules of which conferences/papers to include or not. Not all these databases have citation information (e.g. DBLP does not). Google Scholar

probably reference to widest set of publications, but the publications are not categorized (into research fields they belong), and there is no author disambiguation. In our project, we will mainly use the Libra database, which is large-scale, and already categorized and author-disambiguated.

□ **End of chapter.**

Chapter 3

Definition of ASN and Metrics

Summary

In this chapter, we first present the definition of the academic social network, which is composed of different types of nodes and relationships between them in Section 3.1. We then discuss the notations in Section 3.2 and briefly review the basic PageRank Algorithm in Section 3.3. Finally, we describe the metrics we studied in Section 3.4.

In a general academic social network, there are many types of nodes and edges. For example, node types can be papers, authors and publishing venues, etc; and edges can be citations (linking papers to the papers they cite; authorships (connecting authors to the papers they have written), and so on.

3.1 Network Types

All the metrics we studied can be defined by considering four types of nodes (a) papers, (b) authors, (c) venues and (d) institutions. The relationships between these nodes are captured by the following networks (graphs):

a) Paper citation network, denoted by a directed graph $G_P = (V_P, E_P)$, where V_P is the set of papers and E_P is the set of citations from one paper to another. Citations have directions, therefore, each citation can be represented by an ordered paper pair, $\forall e \in E_P, e = (t, f)$, where $t, f \in V_P$, meaning paper t is cited by paper f.[1]

b) Authorship bipartite network, denoted by $G_{AP} = (V_A \cup V_P, E_{AP})$, where V_A is the set of authors and edges in the set E_{AP} link each paper to its authors (authorship) and symmetrically each author to his/her publications (ownership).

c) Venueship bipartite network, denoted by $G_{VP} = (V_V \cup V_P, E_{VP})$, where V_V is the set of venues and the edges in E_{VP} connect each paper to its publishing venue. Topologically, G_{VP} is similar to G_{AP}. The main difference is that each paper can have multiple authors while it can only be published in one venue.

d) Author-institution bipartite network, denoted by $G_{AS} = (V_A \cup V_S, E_{AS})$, where V_S is the set of institutions and edges in the set E_{AS} affiliate authors to their working institutions.

Fig. 3.1 shows an example of the super-graph $G_{ASN} = (V, E)$ combining all the four networks together. In this case, $V = (V_P \cup V_A \cup V_V \cup V_S)$ and $E = (E_P \cup E_{AP} \cup E_{VP} \cup E_{AS})$. We also denote $n_P = |V_P|, n_A = |V_A|, n_V = |V_V|$ and $n_S = |V_S|$ as the number of papers, authors, venues and institutions, respectively.

[1]Throughout this thesis, we will adopt this special order of the paper pair for representing the citations. The reason is that we try to keep this order consistent with the increasing time line: a paper can only be cited by those later published papers (on the time line, the directed citation edge is originated from a right position and pointing to a left position).

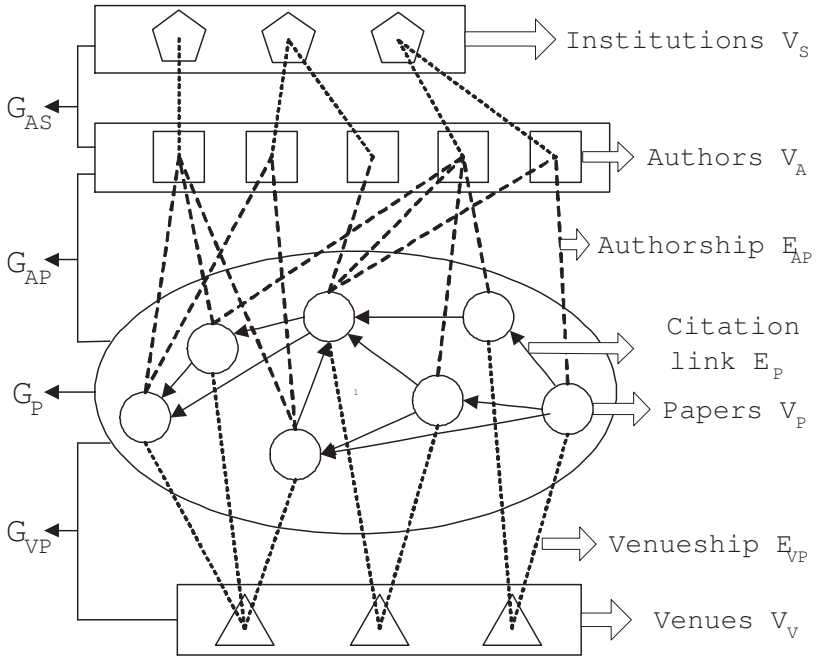


Figure 3.1: An example of the underlay topology of the Academic Social Network, G_{ASN}.

3.2 Notations

We summarize the notations in Table 3.1.

We use an example to illustrate these notations. As shown in Fig. 3.2, it is a simple ASN with 7 papers, $V_P = \{a, b, c, d, e, f, g\}$ and 2 authors, $V_A = \{h, i\}$. By definition, we have:

(1) the paper set cited by paper d, $F(d) = \{a, b, c\}$;
(2) the paper set that cite paper d, $T(d) = \{e, f, g\}$;
(3) the paper set written by author h, $P(h) = \{b, c, d\}$;
(4) the author set who wrote paper d, $A(d) = \{h, i\}$.

Table 3.1: Summary of notations.

n_P	total number of papers in V_P,
n_A	total number of authors in V_A,
n_V	total number of venues in V_V,
n_S	total number of institutions V_S,
$T(p),\ \forall\ p \in V_P,$	the set of papers that cite paper p, i.e., $T(p) = \{f \mid \forall f \in V_P, s.t., (p, f) \in E_P\}$, and $\lvert T(p)\rvert$ is known as the citation count of paper p.
$F(p),\ \forall\ p \in V_P$	the set of papers that are cited by paper p, i.e., $F(p) = \{t \mid \forall t \in V_P, s.t., (t, p) \in E_P\}$, and $\lvert F(p)\rvert$ is known as the reference count of paper p.
$P(a),\ \forall\ a \in V_A$	the paper set that are written by author a, and $\lvert P(a)\rvert$ is known as the publication count of author a.
$A(p),\ \forall\ p \in V_P$	the author set that have written paper p, and $\lvert A(p)\rvert$ is known as the coauthor count of paper p.
$(M)^*$	row normalization operation on any matrix M, i.e., $(M)^*_{ij} = \frac{M_{ij}}{\sum_k M_{ik}}$, for non-zero rows.
R	$n_P \times n_P$ paper-citation adjacent matrix, $R_{ij} = 1$, if paper i has cited paper j, else 0.
A	$n_P \times n_A$ paper-author adjacent matrix, $A_{ij} = 1$, if paper i is written by author j, else 0.
V	$n_P \times n_V$ paper-venue adjacent matrix, $V_{ij} = 1$, if paper i has published in venue j, else 0.
H	$n_A \times n_A$ author influencing matrix, $H = (A^T)^*(R)^*(A)^*$.
Y	$n_V \times n_V$ venue influencing matrix, $Y = (V^T)^*(R)^*(V)^*$.
F	$n_A \times n_A$ author following indicating matrix, $F_{ij} = 1$ if author i has cited author j's paper at least once, else 0.
N	$n_A \times n_A$ author collaboration matrix, $N = A^T A$.
T_{VA}	$n_V \times n_A$ value transition matrix, $T_{VA} = (V^T)^*(A)^*$.
T_{AV}	$n_A \times n_V$ value transition matrix, $T_{AV} = (A^T)^*(V)^*$.
X	$X^{(n_A+n_V)\times(n_A+n_V)} = \begin{pmatrix} \alpha(H)^* & (1-\alpha)T_{AV} \\ (1-\alpha)T_{VA} & \alpha(Y)^* \end{pmatrix}$.

Before we introduce the metrics, it is necessary for us to quickly and briefly explain the well-known PageRank algorithm in that the definition of some metrics involve the PageRank-like algorithm.

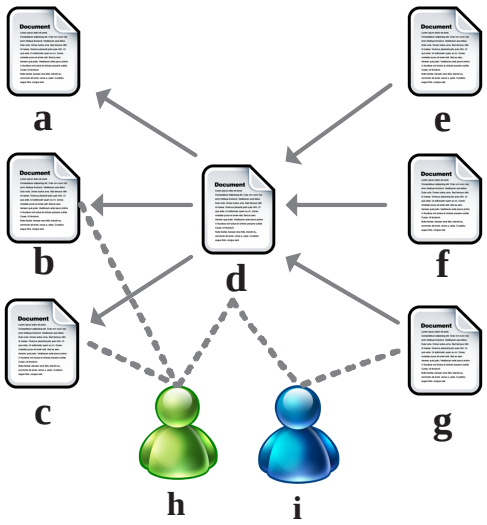


Figure 3.2: An example of a simple ASN with 7 papers $(a-g)$ and 2 authors (h, i).

3.3 The Page Rank Algorithm

Given a graph $G = (V, E)$, the PageRank Algorithm can be considered as a random walk starting from any node along the edges. After an infinite number of steps, the probability that a node is visited is the PageRank value of that node.

More formally, the probability distribution of visiting each node can be derived by solving a Markov Chain. The transition matrix C's entries c_{ij} $(i, j = 1, 2, \ldots, n)$ represent the transition probability that the random walk will visit node j next given that it is currently at node i. Thus, c_{ij} can be expressed as

$$c_{ij} = Prob(j|i) = \frac{e_{ij}}{\sum_k e_{ik}} \tag{3.1}$$

where e_{ij} is from the adjacency matrix for the graph G. If G is the citation graph, for example, then $e_{ij} = 1$ if paper i cites paper j; else $e_{ij} = 0$.

In general, C is a *substochastic* matrix with rows summing to either 0 (dangling nodes [28], for example, representing papers with citing no other papers) or 1 (normal nodes, or papers). For

each dangling node, the corresponding row is replaced by $\frac{1}{n}\mathbf{e}$, so that C becomes a *stochastic* matrix. In order to ensure the Markov Chain C is irreducible, hence a solution is guaranteed to exist, C is further transformed as follows:

$$\widetilde{C} = \alpha C + (1-\alpha)\mathbf{e}\mathbf{v}^T, \ \ \alpha \in (0,1). \tag{3.2}$$

Here, $\mathbf{e}$ is a special column vector with all 1s, and of dimension n.

In Eq. (3.2), $\mathbf{v} \in \mathcal{R}^n$ is a probability vector (i.e. its values are between 0 and 1, and sum to 1). It is referred to as the *teleportation vector*, which can be used to configure some bias into the random walk. For our purposes, we let $\mathbf{v} = 1/n\mathbf{e}$ as the default setting.

Now, according to the Perron-Frobenius Theorem [53, 60], matrix $\widetilde{C}$ is *stochastic, irreducible* and *aperiodic*, and the equation

$$\pi^T = \pi^T\widetilde{C} = \alpha\pi^T C + (1-\alpha)\frac{1}{n}\mathbf{e}^T, \ \ \alpha \in (0,1) \tag{3.3}$$

which can be solved by iteration methods in practice.

3.4 Metrics

We have designed a variety of metrics for all the four object types: (a) papers, (b) authors, (c) venues and (d) institutions. However, in this thesis, we only focus on the metrics of authors (venues[2]). For metrics of papers and institutions, we will not go deeper, but briefly discuss how we design and implement them for practical use in Chapter 4.

We group the metrics of authors we defined into three categories. A metric may be a simple count, such as citation count, or a value derived iteratively using a PageRank-like algorithm.

[2]Since topologically, G_{VP} is symmetric to G_{AP}, the metrics of venues automatically share the same physical meaning as authors. Unless otherwise noted, we use "metrics of authors" to represent the metrics for both authors and venues for simplicity.

3.4.1 Category I: Paper Based

In this case, each paper has a value defined by a metric (for paper). The value is distributed to the paper's authors in a way also determined by the metric. For this category, we study six metrics: **Citation count** (CC), **Balanced citation count** (BCC), **Average citation count** (ACC), **H-index**, **G-index** and **Citation value** (CV). For CC, BCC, ACC, H-index and G-index, the paper's value is simply the citation count, which is well-defined. The difference of them is how the paper value is distributed. While for CV, the paper's value is computed iteratively based on the citation graph G_P.

Citation Count (CC)

In CC's case, each co-author fully receives the paper's citation count. The CC of author a takes a summation of all the citation counts of his/her papers. Mathematically, we have Eq. (3.4):

$$CC(a) = \sum_{p \in P(a)} |T(p)|, \quad \forall a \in V_A. \tag{3.4}$$

Average Citation Count (ACC)

In ACC's case, each co-author again fully receives the paper's citation count. The ACC of author a takes the average of the citation counts of his/her papers. Mathematically, we have Eq. (3.5):

$$ACC(a) = \frac{\sum_{p \in P(a)} |T(p)|}{|P(a)|}, \quad \forall a \in V_A. \tag{3.5}$$

Balanced Citation Count (BCC)

In BCC's case, each co-author only receives an equal fraction of the paper's citation count. The BCC of author a takes a

summation on these received fraction of his/her papers. Mathematically, we have Eq. (3.6):

$$BCC(a) = \sum_{p \in P(a)} \frac{|T(p)|}{|A(p)|}, \quad \forall a \in V_A. \tag{3.6}$$

H-index

H-index is a popular and widely used metric, defined and suggested by Jorge E. Hirsch [21, 47]. Each co-author fully receives the paper's citation count. To calculate the H-index of author a, we first rank all his/her publications in the descending order of the paper citation count, i.e., we have the ordered paper list $p_1, p_2, \ldots, p_{|P(a)|} \in P(a)$, such that $|T(p_1)| \geq |T(p_2)| \geq \ldots \geq |T(p_{|P(a)|})|$. Then, the H-index of author a is the largest integer number satisfying Eq. (3.7):

$$h(a) = \arg\max_h |T(p_h)| \geq h, h \in \mathcal{N}. \tag{3.7}$$

G-index

G-index is another well known and widely used metric, proposed by Leo Egghe [38]. Paper's citation count is fully received by all the co-authors. Similar to H-index, the first step is also to get the ordered paper list of author a: $p_1, p_2, \ldots, p_{|P(a)|} \in P(a)$, such that $|T(p_1)| \geq |T(p_2)| \geq \ldots \geq |T(p_{|P(a)|})|$. Then, the G-index of author a is the largest integer number satisfying Eq. (3.8):

$$g(a) = \arg\max_g \frac{\sum_{i=1}^{g} |T(p_i)|}{g} \geq g, g \in \mathcal{N}. \tag{3.8}$$

Citation Value (CV)

For CV, the paper value is no longer the simple citation count, but a value computed iteratively based on the citation graph G_P

and then distributed to the co-authors in equal fractions. We use $r(p)$ to denote this value and get Eq. (3.9):

$$CV(a) = \sum_{p \in P(a)} \frac{r(p)}{|A(p)|}, \quad \forall a \in V_A. \tag{3.9}$$

Next we describe how we derive $r(p)$ in Eq. (3.9). The paper-citation adjacent matrix R (the matrix form of graph G_P) is an $n_P \times n_P$ squire matrix, whose entries $R_{ij} = 1$, if paper i cites paper j, or equivalently, $\exists e \in E_P, e = (i, j), i, j \in V_P$, otherwise, $R_{ij} = 0$. We define the row normalization operation $(M)^*$ on any input matrix M, as:

$$(M)^*_{ij} = \frac{M_{ij}}{\sum_k M_{ik}}, \quad \text{for non-zero rows.} \tag{3.10}$$

The paper value $r(p)$ is just the solution of applying PageRank algorithm on $(R)^*$.

3.4.2 Category II: Author Based

In this category, metrics are computed based on author-to-author relationships directly. For the three metrics we study in this category: **Influence** (Inf), **Followers** (Fol) and **Connections** (Con), we first define three matrices to represent the author-to-author relationships respectively. The metrics are then computed iteratively by applying the PageRank-like algorithm.

Influence (Inf)

For Influence, the author-to-author relationship is derived from the citation graph G_P and authorship graph G_{AP}. Every time author i cites author j's paper, author i's Influence is distributed to author j, split among the co-authors of j. According to this description, we can calculate the influence of author a in the

way as expressed in Eq. (3.11):

$$Inf(a) = \sum_{p \in P(a)} \frac{1}{|A(p)|} \sum_{t \in T(p)} \frac{1}{|F(t)|} \sum_{b \in A(t)} \frac{1}{|P(b)|} Inf(b). \quad (3.11)$$

Eq. (3.11) is quite complicated and unreadable, however, the equivalent matrix form will be much simpler and clearer. The paper-authorship adjacent matrix A (the matrix form of graph G_{AP}) is an $n_P \times n_A$ matrix, whose entries $A_{ij} = 1$, if paper i is written by author j, otherwise, $A_{ij} = 0$. The author influencing matrix H is defined in Eq. (3.12):

$$H = (A^T)^*(R)^*(A)^*, \quad (3.12)$$

where H is an $n_A \times n_A$ squire matrix. The author Influence value is the solution of applying PageRank algorithm on $(H)^*$.

Followers (Fol)

For Followers, the author-to-author relationship is also derived from the citation graph, but depended on whether author i cited author j instead of how many times. If author i cited author j, author i's Follower value is distributed to author j without splitting among author j's co-authors (which can be different for different papers). In particular, the author following indicating matrix F is an $n_A \times n_A$ matrix, whose entries $F_{ij} = 1$, if author i has cited author j's paper at least once, otherwise, $F_{ij} = 0$. Hence the author Followers value is the solution of applying PageRank algorithm on $(F)^*$.

Connections (Con)

The author-to-author relationship for Connections is defined only based on the authorship graph G_{AP}. If author i has co-authored a paper with author j, then author i's Connections value is distributed to author j and *vice versa*. Note, another

variation of Connections can also be defined so that every time author i co-authors with author j, they exchange their Connections value. Here we adopt the former one. By denoting N as the author collaboration matrix, we have Eq. (3.13):

$$N = A^T A, \tag{3.13}$$

where N is an $n_A \times n_A$ symmetric matrix. The author Connections value is the solution of applying PageRank algorithm on $(N)^*$.

3.4.3 Category III: Author-Venue Based

In this category, we define only one metric: **Exposure** (Exp).

Exposure (Exp)

This metric is computed by iterating on authors and venues together. It is easiest to think of venues also as a kind of author, thus we have an enlarged author set $V_A \cup V_V$.

The author-to-author relationship is defined in the same way as Influence; so is the relationship for venue-to-venue. The author-to-venue and venue-to-author relationships are defined intuitively as follows: each time an author i writes a paper published in venue k, author i distributes his/her influence to venue k; similarly, each time a venue k publishes a paper co-authored by i, author i shares a fraction of venue k's influence with i's co-authors for that paper.

Formally, we still use $H = (A^T)^*(R)^*(A)^*$ to denote the $n_A \times n_A$ author influencing matrix. Symmetrically, we define $Y = (V^T)^*(R)^*(V)^*$ to denote the $n_V \times n_V$ venue influencing matrix, where V is the $n_P \times n_V$ paper-venueship adjacent matrix, whose entries $V_{ij} = 1$, if paper i is published in venue j, otherwise, $V_{ij} = 0$. In addition, we define $T_{VA} = (V^T)^*(A)^*$

and $T_{AV} = (A^T)^*(V)^*$ to be the venues-to-authors and authors-to-venues influence transition matrix, respectively. Finally, we have Eq. (3.14):

$$X = \begin{pmatrix} \alpha(H)^* & (1-\alpha)T_{AV} \\ (1-\alpha)T_{VA} & \alpha(Y)^* \end{pmatrix}, \tag{3.14}$$

where X is the $(n_A + n_V) \times (n_A + n_V)$ author-venue exposure matrix. Therefore, the solution of applying PageRank algorithm on $(X)^*$ contains both the author Exposure value, and the venue Exposure value.

It is worth noting that all these metrics are defined so as to assign a value to each author, to indicate some characteristics of that author. Since citation count (CC) can be inflated by a large number of co-authored papers [31] (so are the H-index and G-index, although they are very popular and widely accepted), BCC and CV are alternative computations to assign citation credits to authors. The metrics Influence and Followers are intended to characterize an author's influence and impact on other authors. The metric Connections is used to measure an author's reach in the co-authorship network. Finally, Exposure is intended to bring in the impact of the venues to help characterize an author's potential influence that may not be reflected by citations if the author's papers are relatively recent. [74] has conducted an atomic study on the properties of different metrics and had some interesting findings.

□ **End of chapter.**

Chapter 4

Ranking Implementation and Evaluation

Summary

In this chapter, we discuss the implementation issues and evaluation results. We describe the available datasets and data preparation issues for ranking implementation in Section 4.1. We discuss the various ranking features and useful functions in Section 4.2. Finally, we present the approaches used to evaluate and validate our metrics and ranking methods in Section 4.3.

4.1 Data Preparation

4.1.1 Data Sources

For Social Network Analysis (SNA) study, there exist a bench of datasets for free access [11]. As a special type, some of them are for the Academic Social Network Analysis (ASNA), e.g., the citation networks of arXiv dataset in [11].

In addition, as a major component of the biblio/scientometric research, a considerable number of data sources are maintained

and provided for research on the ASN, including: DBLP [6, 54], Microsoft Academic Research Libra [4, 67], Google Scholar [3], ISI Web of Knowledge [1], Elsevier Scopus [2], CiteSeerX [5, 56], American Physical Society (APS) [7] and so on.

However, these data sources possess very different properties, e.g., some are free accessible while some are on a paid basis. We first introduce and explain the properties we are interested in. According to these properties, we then discuss how we categorize the data sources and select some of them for our research work.

Multidisciplinary/Sub-topics

Some data source is targeting at one specific research field, (e.g. DBLP focuses on Computer Science field and APS on Physics) while others are multidisciplinary (e.g. Libra and Scopus). In particular, some data source (like Libra) provides more detailed domain categorization information for each research field.

Data Availability

For this property, we define the following categories:

(1) Free Access and Friendly
Datasets are public available with no limitation. In particular, some dataset is packaged for downloading and public use (e.g. DBLP, APS) or providing well designed APIs (e.g. Libra, CiteSeerX).

(2) Free Access but Limited
Datasets are public available but under limited usage, (e.g. Google Scholar).

(3) Access with Charges
The remaining datasets usually charge a large amount of money for access, e.g., ISI and Scopus.

Information organization

We propose a 3-layer data model to categorize the available data information.

1) Raw text layer
   All types of nodes, such as papers, authors and venues etc, are identified by text strings. Most datasets are organized at this layer, e.g., DBLP, Scopus. The well-known and widely used data format "bibtex" is one of the most representative examples.

2) Object layer
   The object type includes: author, paper, conference venue, institution and so on. Each type of object possesses general properties such as a unique identifier, name and relationship to other objects (e.g. Libra). For example, if the object is a paper, then its properties include: paper ID, title, publication year, authorship and citations.

   Some datasets provide partial object-layer information. For example, only papers of APS and arXiv datasets are maintained as objects with paper ID and citations information.

3) Application layer
   The application layer includes information like rankings, comparisons, different grouping, statistics etc. Construction of this layer is the target of our work.

In Table 4.1, we list the categorization on the various datasets based on the properties discussed above.

4.1.2 Data Collection

The strategy for data collection depends on the availability property of each target dataset.

Table 4.1: Properties of datasets

Data set	Multidisciplinary	Availability	Organization
DBLP	Single	Free & Friendly	Text-layer/No Citations
DBLP-Cit	Single	Free & Friendly	Partial Object-layer
CiteSeerX	Single	Free & Friendly	Partial Object-layer
APS	Single	Free & Friendly	Partial Object-layer
ArXiv	Single	Free & Friendly	Partial Object-layer
Google Scholar	Multi/No Cate.	Free but Limited	Text-layer
ISI WoK	Multi/sub-domain	Charges	Partial Object-layer
Scopus	Multi/sub-domain	Charges	Partial Object-layer
Libra	Multi/sub-domain	Free & Friendly	Object-layer

For example, DBLP provides the formatted and packaged DBLP XML records [6, 54] for downloading, therefore, the only thing we need to do is to parse the XML records and manage them for future use. So are the DBLP-Cit [12] (created by a third party based on the original DBLP dataset with adding paper citation relationships through proper mining method [78, 79]), arXiv [13] and APS [7] datasets.

For those datasets with well designed APIs (e.g. Libra and CiteSeerX) provided, the parsing step is not necessary, only need to understand how to correctly use the APIs.

Finally, the traditional way of data collection, which is web-page crawling and parsing, can be considered as the default setting.

4.1.3 Data Management

In data management, the most difficult part is how to transfer data organization from the raw-text layer to the object layer, unless the dataset (like Libra) is already at the object layer. In fact, this involves many open challenges and hot research topics, e.g., the Author-Paper Identification Challenge and the Author Disambiguation Challenge in KDD Cup 2013 competition [8].

The second task is to design a unified data format for repre-

senting and storing different datasets after they are transferred to the object-layer. One straightforward way is to use database tool (e.g. MySql). In our implementation, before importing into database, we also use an XML format to store the object-layer data information. Fig. 4.1 shows an example.

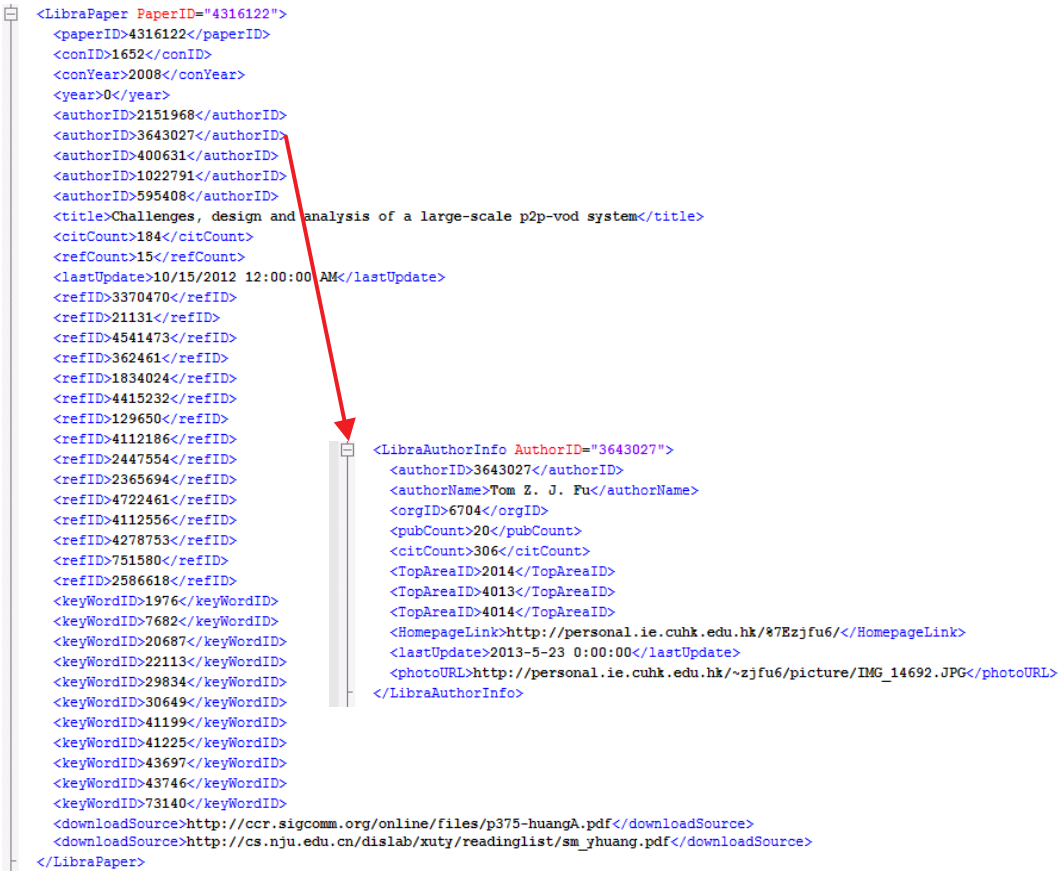


Figure 4.1: Illustration of how we store the object-layer information in XML format.

In reality, paper records are increasing with time, therefore we need a mechanism for appending new data records from time to time. For each paper record, it has two types of citation information, the paper list that this paper has referenced to and the paper list by which this paper is cited. Since the paper can only be cited by later published papers, we only need to keep the *reference to* paper list information for each newly added paper

record (those paper IDs included in those tags named "RefID" in Fig. 4.1).

4.1.4 The Libra Dataset

Considering the Microsoft Libra dataset is easy for collection (providing APIs) and has an object-layer organization [67], we decide to use it for experimental and evaluation purpose in our academic research ranking problem.

In fact, Libra has maintained a huge amount of data in a very wide range of research fields (15) and, for each field, it further categorizes the papers to belong to domains in that field. The data set we obtain for experimental purposes is for the Computer Science field, which includes 24 domains. Other facts we need to consider are: (a) an increasing proportion of these papers are published in more recent years, as shown in Fig. 4.2; (b) authors tend to be more collaborative on publishing papers in more recent years, as shown in Fig. 4.3. These have some ramifications for our analysis, as we discuss in a latter part. Despite the misgivings about the dataset we make many interesting observations.

4.1.5 Data Pre-processing

Libra dataset provides additional paper-domain categorization information in each research field. By using this domain information, we are able to conduct domain-specific ranking results for each metric.

To achieve this, data pre-processing is required. If we consider the graphs, G_P, G_{AP} and G_{VP} defined in the last chapter, are at the research field scale, in the following, we give the definition of subgraphs for those domains under this research field.

Given a subset of papers belonging to a domain D, denoted by $V_P^{D*} \subset V_P$, we construct the set V_P^D, which will finally be

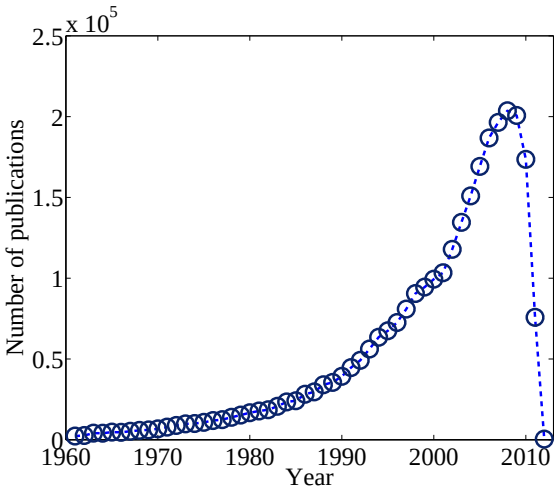


Figure 4.2: The number of papers in the Libra CS domain changing with time.

involved in the ranking calculation, by applying the constraints in Eq. (4.1):

$$p, q \in V_P^D \;\Leftrightarrow\; p, q \in V_P^{D*}, \text{either } e = (p, q) \in E_P, \text{ or } e = (q, p) \in E_P. \tag{4.1}$$

The remaining definitions are listed in Table 4.2.

Table 4.2: Remaining definitions of subgraphs for domain D.

Description	notation	Definition
Subset of citations	E_P^D	$e \in E_P^D \Leftrightarrow e \in E_P$ and $e = (t, f),\ t, f \in V_P^D$.
Subset of authors	V_A^D	$a \in V_A^D \Leftrightarrow a \in V_A, \exists e = (a, p) \in E_{AP},\ p \in V_P^D$.
Subset of authorship	E_{AP}^D	$e \in E_{AP}^D \Leftrightarrow e \in E_{AP}$ and $e = (a, p),\ a \in V_A^D, p \in V_P^D$.
Subset of venues	V_V^D	$v \in V_V^D \Leftrightarrow v \in V_V, \exists e = (v, p) \in E_{VP},\ p \in V_P^D$.
Subset of venueship	E_{VP}^D	$e \in E_{VP}^D \Leftrightarrow e \in E_{VP}$ and $e = (v, p),\ v \in V_V^D, p \in V_P^D$.
Citation subgraph	G_P^D	$G_P^D = (V_P^D, E_P^D)$
Paper-author subgraph	G_{AP}^D	$G_{AP}^D = (V_A^D \cup V_P^D, E_{AP}^D)$
Paper-venue subgraph	G_{VP}^D	$G_{VP}^D = (V_V^D \cup V_P^D, E_{VP}^D)$

Table 4.3 lists the name and the pre-processing results according to Eq. (4.1) and Table 4.2 (the number of papers in subset V_P^{D*}, the number of papers in V_P^D and the number of authors in V_A^D) in each of the 24 domains of the Computer Science Field.

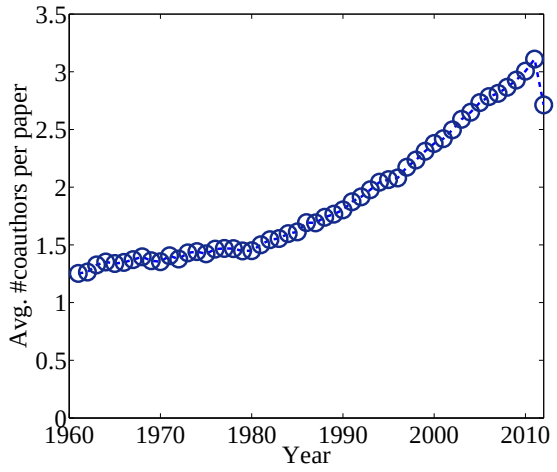


Figure 4.3: Average number of coauthors per paper in the Libra CS domain changing with time.

Since each author may publish papers in different domains, the sum of authors in all domains is significantly greater than the number of unique authors (941733). The number of papers in the database (3347795) is actually significantly greater than the sum from all domains (2449673). This is because many papers are not classified or have missing information.

4.2 Ranking Features

4.2.1 Ranking Types

Given the metrics we defined, we compute for each author his/her ranking for each metric. We have designed five types of ranking results:

i) Rank

This is the raw ordered position after ranking each author based on the values computed for each metric.

Table 4.3: The basic information of the Libra dataset we use: domain name, the number of papers in subset V_P^{D*}, the number of papers in V_P^D and the number of authors in V_A^D (according to Eq. (4.1) and Table 4.2), in each of the 24 domains of the Computer Science Field.

Domain Name	$\|V_P^{D*}\|$	$\|V_P^D\|$	$\|V_A^D\|$
Algorithms and Theory	270601	158540	96748
Security and Privacy	61957	38364	33910
Hardware and Architecture	150151	88992	81021
Software Engineering	174893	100221	85938
Artificial Intelligence	325109	224827	186976
Machine Learning and Pattern Recognition	108234	65808	66839
Data Mining	67485	43122	50958
Information Retrieval	51075	27317	30038
Natural Language and Speech	220227	104388	86670
Graphics	59880	37619	36548
Computer Vision	60806	46957	44969
Human-Computer Interaction	79909	48741	51548
Multimedia	80618	48088	59277
Network and Communications	235297	157912	138096
World Wide Web	35861	18439	25098
Distributed and Parallel Computing	117836	73714	69592
Operating System	25395	15091	18167
Databases	142421	83114	74125
Real-Time and Embedded System	33098	20813	21965
Simulation	27678	15169	18083
Bioinformatics and Computational Biology	55491	27251	48729
Scientific Computing	183878	99839	103982
Computer Education	49125	23679	29420
Programming Languages	70561	44019	33229
Computer Science Overall (24 domains)	2449673	1700637	941733
Computer Science Total Involved	3347795	2131645	1175052

ii) Rank percentile (RankPer)

The RankPer of each author equals to his/her rank position divided by the total number of authors ranked. Different to the absolute position (Rank), RankPer provides the relative ranking information in the whole author set.

iii) Cumulative value of contribution (CumValue)

A third choice is to view the ranking information in terms of the cumulative value of contribution by authors ranked ahead of the target author.

In particular, the CumValue of author a for one metric is:

$$CumValue(a) = \sum_{b \in \{i|Rank(i)<Rank(a), i \in V_A\}} Value(b), \qquad (4.2)$$

where $\{i|Rank(i) < Rank(a), i \in V_A\}$ is the set of authors who are ranked ahead of author a and $Value(b)$ denotes the derived value of author b for this metric.

To illustrate the ranking results of the above three types, here we show an example of an author "J Smith" (with the actual name anonymized) returned by our web service:

Table 4.4: An example of the different metric results returned by our web service in the "Network and Communications" domain with actual author name anonymized.

Value Type	Author	CC	BCC	CV	Inf	Fol	Con	Exp
Rank	J Smith	4786	2483	2996	4100	7647	2820	1805
RankPer	J Smith	3.5%	1.8%	2.2%	3.0%	5.5%	2.0%	1.3%
CumValue	J Smith	72.3%	63.7%	58.5%	56.5%	59.5%	18.2%	26.9%

As in Table 4.4, the 2nd row lists the Rank results while the 3rd and 4th rows list the results in terms of RankPer and CumValue, respectively. Actually, this ranking is for a specific domain ("Network and Communications") which has close to 138K authors of Libra dataset. So this author is ranked well within the top ten percentile of this domain he/she works in.

iv) Contribution based letter grading (Contri. Letter)

Besides, we consider it more appropriate to use a coarse granularity for such ranking information (especially for case study

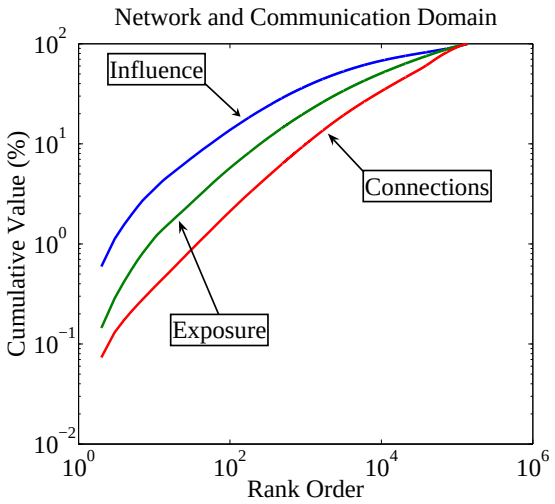


Figure 4.4: The loglog results of rank orders versus cumulative values of three metrics: Influence, Connections and Exposure.

purpose). There are two possible ways, one of which is based on cumulative value of contribution.

For this purpose, we decide to divide the CumValue range into five fixed intervals, and assign letter grade "ABCDE" as ranks. Lacking any better way to calibrate the partitioning, we simply use 20%, 40%, 60% and 80% as the thresholds. In this view, the above example becomes (Table 4.5):

Table 4.5: Example of the contribution based letter grades for each metric, where A:$(0-20\%)$, B:$(20\%-40\%)$, C:$(40\%-60\%)$, D:$(60\%-80\%)$ and E:$(80\%-100\%)$

Value Type	Author	CC	BCC	CV	Inf	Fol	Con	Exp
CumValue	J Smith	72.3%	63.7%	58.5%	56.5%	59.5%	18.2%	26.9%
Contri. Letter	J Smith	D	D	C	C	D	A	B

For most metrics, the distribution of contribution by authors ordered according to ranking follows Pareto-like distribution. For example, Fig. 4.4 shows the relationship between the rank order to the cumulative value of three metrics, Influence, Connections and Exposure, using a loglog plot.

So out of over 138K authors, the distribution of "ABCDE" for the different metrics are listed in Table 4.6.

Table 4.6: The distribution of contribution based letter assignment for different metrics of around 138k authors in "Network and Communications" domain of Libra dataset.

	CC	BCC	CV	Inf	Fol	Con	Exp
A	156	148	179	214	485	3386	940
B	558	513	752	994	1764	11516	3978
C	1629	1469	2366	4134	5646	32653	12251
D	5550	5059	9012	25916	26705	20866	31962
E	130203	130907	125787	106838	103496	69675	88965

v) Rank Percentile based letter grading (RankPer Letter)

The second way of letter assignment is based on rank percentile (RankPer). Since the cumulative curves of the metrics show the power-law property, we thus propose the power-based thresholds $(\mu^4, \mu^3, \mu^2, \mu)$ to assign letters according to the rank percentile, where parameter $\mu \in (0, 1)$ controls the skewness of the assignment results. Table 4.7 illustrates the letter assignment results when we set $\mu = 0.25$ for the experimental web site.

Table 4.7: The illustration of power-based letter assignment according to the rank percentile with parameter $\mu \in (0, 1)$.

RankPer Letter	Rank Percentile	$\nu = 0.25$
A	$(0 - \mu^4)$	$(0 - 0.39\%)$
B	$(\mu^4 - \mu^3)$	$(0.39\% - 1.56\%)$
C	$(\mu^3 - \mu^2)$	$(1.56\% - 6.25\%)$
D	$(\mu^2 - \mu)$	$(6.25\% - 25\%)$
E	$(\mu - 1)$	$(25\% - 100\%)$

The letter grades according to the rank percentile on the "J Smith" example are listed in Table 4.8.

It remains an open problem of how to find the best way of letter grades assignment, which we consider to be future work.

Table 4.8: Letter grades for each metric by power-based assignment according to the rank percentile on the example "J Smith", where $\mu = 0.25$.

Value Type	Author	CC	BCC	CV	Inf	Fol	Con	Exp
RankPer	J Smith	3.47%	1.8%	2.17%	2.97%	5.54%	2.04%	1.31%
RankPer Letter	J Smith	C	C	C	C	C	C	B

We briefly discuss the pros and cons of the two letter assignment methods proposed by us, contribution based vs. rank percentile based.

One notable difference is the metric-dependency of the letter count distribution. By definition, rank percentile based letter grading results in a consistent letter count distribution among different metrics (hence independent of metrics). However, it varies a lot among different metrics for letter count distribution generated by contribution based letter grading. For example, as shown in Table 4.6, there are 156 "A"s for the Citation Count (CC) metric when 3386 "A"s for the Connection (Con) metric. This is caused by the different skewness in the value distribution of authors' contribution for various metrics, which can also be inferred from the cumulative curves shown in Fig. 4.4.

On the other hand, any change in the total number of authors in a research domain (e.g. community expansion or rapid development) unavoidably affects the letter count distribution generated by rank percentile based letter grading, but it has very limited effects on the contribution based letter assignment results when the value distribution is very skewed (e.g. Influence and so on).

Later on, unless otherwise noted, we only show the letter assignment results by rank percentile based grading for space saving and fair comparison among various metrics.

4.2.2 Domain-specific vs Overall Ranking

As mentioned, the above example is the ranking for an author in a specific domain. Usually, an author works in several domains. Our web service shows the author's rankings in all the domains, as well as an overall score for his/her subject field (in this case "Computer Science"). The letter grades of the example "J Smith" are listed in Table 4.9).

Table 4.9: Letter grades of each metric in several involved domains of the example "J Smith".

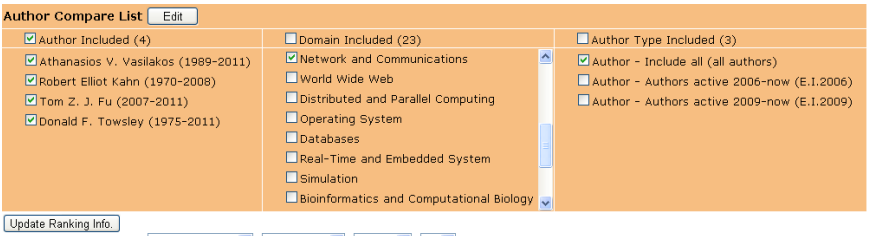
Domain	CC	BCC	CV	Inf	Fol	Con	Exp
Network and Communications	C	C	C	C	C	C	B
Security and Privacy	D	D	D	D	E	E	D
Computer Science Overall	C	C	C	C	C	C	B

This allows the person to be compared to others in his/her domain, as well as comparing him/her to a bigger set of people in a subject field.

The way to compute the overall score is difficult. We use the straightforward way of merging all the domains into one big domain and compared the results. This is more computationally demanding. Another possible way is to add up the authors ranking in each domain normalized by the size of each domain. The trade-off of different ways for computing the overall is something still under study.

4.2.3 Comparing Rankings

In our experimental web site, we have implemented different ways for authors to be compared. First of all, authors in the same domain can be looked up in ranking order, according to any metric. So it will be easy to look up top-ranked people according to one's favorite metric, whether it is Influence, Connections, or Exposure. This is often helpful.



AType	Domain	AuthorName	In/AllPub	#Cit Cnt	H-index	G-index	Inf	PVal	Expo	Conn	#Cit	H-in	G-in
Include All	Network and Communications	Athanasios V. Vasilakos (1989-2011)	60/188	87	6	9	D	C	C	A	C	C	C
Include All	Network and Communications	Donald F. Towsley (1975-2011)	439/725	10449	53	102	A	A	A	A	A	A	A
Include All	Network and Communications	Robert Elliot Kahn (1970-2008)	5/23	87	4	9	A	A	C	D	C	C	C
Include All	Network and Communications	Tom Z. J. Fu (2007-2011)	5/9	80	1	8	D	D	D	D	C	D	C

Figure 4.5: A snapshot of the implemented rank comparing function.

Second, we allow authors in the same institution to be looked up in ranking order, for a specific domain, or according to overall ranking. This will be useful in getting a feel as to how strong a particular institute is in a particular domain. It is also the rough way we justify our assignment of "ABCDE" to authors in different cumulative value percentiles or rank percentiles.

We also allow users to search for individual authors and keep them in a list for head-to-head comparison. This can be helpful for many different purposes. For example, we can use this method to collect a list of authors for a case study. An snapshot of the comparing function we have implemented in the website is shown in Fig. 4.5.

4.2.4 Author-based Institution Rankings

With the additional information of author-institution relationships (G_{AS} in Fig. 3.1), we can further provide institution rankings based on authors' ranking results. When ranking institutions, we use two granularities:

(1) We only count the number of authors assigned with "A";

(2) We compute a total score, counting "A" = 1, "B" = 0.5, "C" = 0.25, and "D" = "E" = 0.

For ranking authors, there are a number of various metrics (e.g., Influence, Connections, Exposure, etc.), two types of letter assignment (contribution based vs. rank percentile based) and the domain-specificity (e.g., 24 domains listed in Table 4.3), therefore the institution ranking automatically inherits these features.

4.3 Evaluation and Validation

4.3.1 Ranking Award Recipients

One way to justify our new metrics is to look at award recipients. In the computer science domain, the most prestigious award is the Turing Award. Since we are more familiar with the Network and Communications domain, we also look at the ACM Sigcomm Award recipients. The results are shown in the following three tables (Table 4.10, Table 4.11 and Table 4.12).

In both these cases, it is clear that citation count is not always a good measure, for these people obviously had tremendous contribution and impact in their fields. The Citation Value metric (CV) improved over CC and BCC. But Influence did much better - all the Turing Award winners scored at least "B". For these top people in their fields, the Followers metric was even more predictive. Though, as we will discuss later, we find Influence and Followers quite similar. Aside from trying to justify the Influence and Followers metrics, we can also appreciate the additional information provided by the Connections metric, in distinguishing those who tend to collaborate more from those who tend to work alone.

Since Sigcomm is a more applied community, the CC and BCC metrics perform even worse in comparison to Influence and Followers. This is perhaps because the Sigcomm community publication venues are more selective (hence have more

Table 4.10: Rankings received by Turing Award Recipients (1966 - 1991)

Year	Awardee	CC	BCC	CV	Inf	Fol	Con	Exp	Aff
1966	Alan J. Perlis	B	B	B	A	A	C	B	Yale
1967	Maurice V. Wilkes	B	B	A	A	A	D	A	Cambridge
1968	Richard W. Hamming	B	A	A	A	A	E	B	Naval PG Sch.
1969	Marvin Minsky	A	A	A	A	A	D	A	MIT
1970	James H. Wilkinson	B	A	A	A	A	D	A	NPL, UK
1971	John McCarthy	A	A	A	A	A	B	A	Princeton
1972	Edsger W. Dijkstra	A	A	A	A	A	C	A	UT Austin
1973	Charles W. Bachman	C	B	B	A	B	C	B	Bachman I.S.
1974	Donald E. Knuth	A	A	A	A	A	B	A	Stanford
1975	Allen Newell	A	A	A	A	A	A	A	CMU
	Herbert Simon	A	A	A	A	A	B	A	Illinois I.T.
1976	Michael O. Rabin	A	A	A	A	A	C	A	Columbia
	Dana Stewart Scott	A	A	A	A	A	C	A	CMU
1977	John W. Backus	A	A	A	A	A	C	A	IBM
1978	Robert W. Floyd	A	A	A	A	A	D	A	Illinois I.T.
1979	Kenneth E. Iverson	C	B	B	A	A	C	A	IBM
1980	C. A. R. Hoare	A	A	A	A	A	B	A	MSR
1981	Edgar Frank Codd	A	A	A	A	A	D	A	IBM
1982	Stephen A. Cook	A	A	A	A	A	B	A	U. Michigan
1983	Ken Thompson	A	A	A	A	A	C	A	Google
	Dennis M. Ritchie	A	A	A	A	A	D	A	Bell Labs
1984	Niklaus Emil Wirth	A	A	A	A	A	D	A	Xerox PARC
1985	Richard Manning Karp	A	A	A	A	A	A	A	IBM
1986	John Edward Hopcroft	A	A	A	A	A	B	A	Stanford
	Robert Endre Tarjan	A	A	A	A	A	A	A	HP
1987	John Cocke	A	A	A	A	A	C	A	IBM
1988	Ivan E. Sutherland	A	A	A	A	A	B	A	Portland S.U.
1989	William Morton Kahan	C	C	B	B	B	B	C	UC Berkeley
1990	Fernando Jose Corbato	C	C	B	B	A	D	C	MIT
1991	Robin Milner	A	A	A	A	A	B	A	Cambridge

Table 4.11: Rankings received by Turing Award Recipients (1992 - 2012)

Year	Awardee	CC	BCC	CV	Inf	Fol	Con	Exp	Aff
1992	Butler W. Lampson	A	A	A	A	A	B	A	MIT
1993	Juris Hartmanis	A	A	A	A	A	B	A	Cornell
	Richard Edwin Stearns	A	A	A	A	A	B	A	NYU at Albany
1994	Edward A. Feigenbaum	B	B	A	A	A	C	A	Stanford
	Raj Reddy	B	B	A	A	A	B	A	CMU
1995	Manuel Blum	A	A	A	A	A	B	A	CMU
1996	Amir Pnueli	A	A	A	A	A	A	A	NYU
1997	Douglas C. Engelbart	B	A	A	A	A	C	A	D.E. Ins.
1998	Jim Gray	A	A	A	A	A	A	A	MSR
1999	Fred Brooks	A	A	A	A	A	A	A	UNC
2000	Andrew Chi-chih Yao	A	A	A	A	A	B	A	Tsinghua Univ
2001	Ole-johan Dahl	B	B	A	A	A	C	B	Univ of Oslo
	Kristen Nygaard	B	B	A	A	A	C	B	Univ of Oslo
2002	Ronald L. Rivest	A	A	A	A	A	A	A	MIT
	Adi Shamir	A	A	A	A	A	A	A	Weizmann Ins.
	Leonard Max Adleman	A	A	A	A	A	B	A	MIT
2003	Alan Curtis Kay	B	B	B	B	B	C	B	HP Labs
2004	Vinton Gray Cerf	B	B	B	A	A	B	B	Google
	Robert Elliot Kahn	C	C	A	A	A	C	B	CNRI
2005	Peter Naur	C	B	A	A	A	D	A	U. Copenhagen
2006	Frances E. Allen	B	B	A	A	A	C	B	IBM
2007	Edmund Clarke	A	A	A	A	A	A	A	CMU
	E. Allen Emerson	A	A	A	A	A	B	A	UT Austin
	Joseph Sifakis	A	A	A	A	A	A	A	CNRS
2008	Barbara Liskov	A	A	A	A	A	A	A	MIT
2009	Charles P. Thacker	B	B	B	A	A	C	C	Microsoft
2010	Leslie Valiant	A	A	A	A	A	C	A	Harvard
2011	Judea Pearl	A	A	A	A	A	B	A	UCLA
2012	Shafi Goldwasser	A	A	A	A	A	B	A	Weizmann Ins.
	Silvio Micali	A	A	A	A	A	B	A	MIT

Table 4.12: Rankings received by Sigcomm Award recipients (1989 - 2012)

Year	Awardee	CC	BCC	CV	Inf	Fol	Con	Exp	Aff.
1989	Paul Baran	D	D	C	C	C	E	D	RAND Corp.
1990	Leonard Kleinrock	A	A	A	A	A	A	A	UCLA
	David D. Clark	B	B	B	A	A	B	B	MIT
1991	Hubert Zimmermann	C	B	B	B	B	E	B	Sun Micro.
1992	A. G. Fraser	C	C	B	A	B	E	B	Fraser Research
1993	Robert Elliot Kahn	C	C	A	A	A	C	B	CNRI
1994	Paul E. Green	C	B	B	B	B	C	B	Tellabs
1995	David J. Farber	B	B	B	B	A	B	B	CMU
1996	Vinton Gray Cerf	B	B	B	A	A	B	B	Google
1997	Jonathan B. Postel	A	A	A	A	A	B	A	USC
	Louis Pouzin	D	C	B	B	B	E	C	ITU
1998	Lawrence G. Roberts	B	A	A	A	A	D	A	Anagran Inc.
1999	Peter T. Kirstein	C	C	B	B	B	C	B	UC London
2000	Andre A. S. Danthine	C	C	C	C	C	C	C	Univ de Liége
2001	Van Jacobson	A	A	A	A	A	B	A	PARC
2002	Scott J. Shenker	A	A	A	A	A	A	A	UC Berkeley
2003	David Cheriton	A	A	A	A	A	B	A	Stanford
2004	Simon Lam	A	A	A	A	A	B	A	UT Austin
2005	Paul V. Mockapetris	B	A	A	A	A	E	A	Nominum
2006	Domenico Ferrari	A	A	A	A	A	B	A	UC Berkeley
2007	Sally Floyd	A	A	A	A	A	A	A	ICSI
2008	Donald F. Towsley	A	A	A	A	A	A	A	UMass
2009	Jon Crowcroft	A	A	A	A	A	A	A	Cambridge
2010	Radia J. Perlman	B	B	A	A	B	D	B	Intel
2011	Vern Paxson	A	A	A	A	A	A	A	UC Berkeley
2012	Nick W. Mckeown	A	A	A	A	A	A	A	Stanford

influence). We will discuss the differences between Influence, Followers and Exposure later.

4.3.2 Similarity between Metrics

For our similarity study, we choose to plot the cumulative value (essentially according to letter grades) of each author, for the two comparable metrics. For example, we first compare Citation Count (CC) with Influence as metrics. The former is the common metric used in practice, and the latter is something we proposed. The result is shown in Fig. 4.6. The two verti-

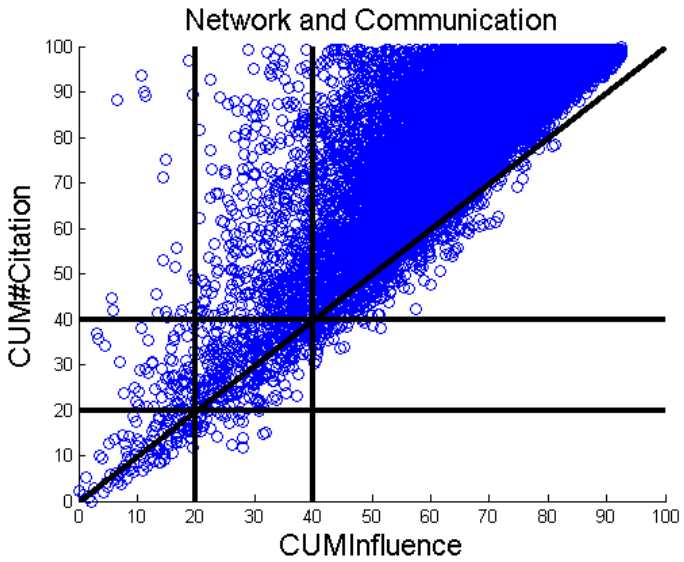


Figure 4.6: Comparison between Influence and Citation Count using cumulative value.

cal and horizontal lines give the boundaries separating "A" and "B" from the rest of the ranks. Any author on the diagonal line receives exactly the same ranking from both metrics. As we can see, there is correlation between Influence and CC - those with high CC ranking all have high Influence ranking as well. But the converse is not true - those with high Influence ranking may not have high CC ranking. This means we can use CC as a sufficient condition when estimating someone's influence, but

not a necessary condition. For this reason, we consider Influence is sufficiently different than CC, and shall be considered as a complementary metric.

The Citation Value (CV) metric is designed to be an alternative to CC. From our experience, an author's CV rank seems to be always between its CC rank and Influence rank. Fig. 4.7 compares CV against Influence. It is indeed similar to the com-

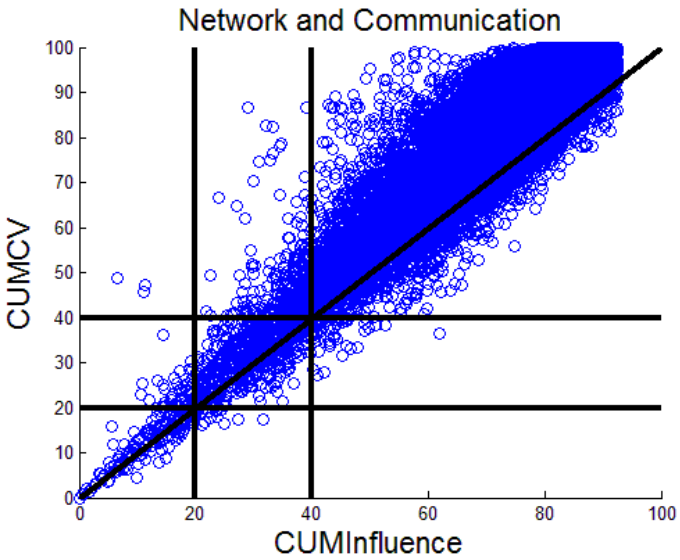


Figure 4.7: The comparison between Influence and Citation Value using cumulative value.

parison to CC, namely high CV implies high Influence but not *vice versa*. Thus, once we have CC and Influence, there is no strong reason to keep CV as an additional metric.

Now let us consider the Followers metric. As we observe in considering the Followers and Influence ranks for the Award recipients, those with a high influence rank tend to have even higher Followers ranks. But for the majority of the authors, these two ranks are very strongly correlated, and hence Followers seem to add little additional value to the Influence metric (as shown in Fig. 4.8).

As expected, the Connections metric has little correlation to

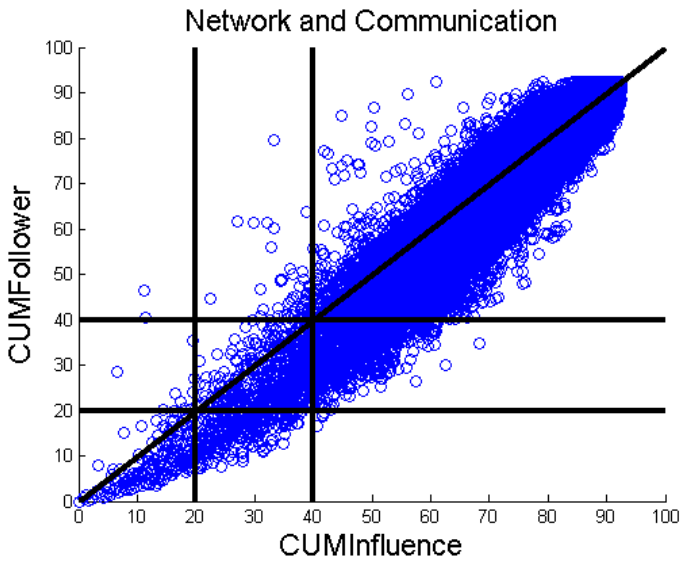


Figure 4.8: The comparison between Influence and Follower using cumulative value.

any of the other metrics. This is quite intuitive, so we have not included any similarity plots to save space.

Finally, we compare the Influence metric to the Exposure metric in Fig. 4.9. In this case,many authors with low Influence

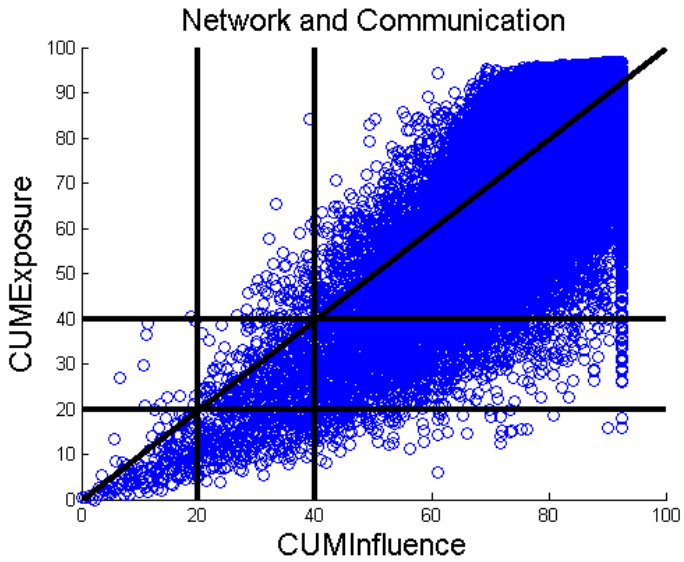


Figure 4.9: The comparison between Influence and Exposure using cumulative value.

values may have much higher ranks in Exposure. We suspect this is because this metric successfully identifies authors who are very active in publishing in high impact venues but have not had the time to build up their influence. It is difficult to tell how true this is - so we selected some real world examples for our case studies in the next subsection.

4.3.3 Case Studies

From the above similarity study, we conclude that, out of the five metrics based on iterative computation, i.e. CV, Influence, Followers, Connections and Exposure, the first three are sufficiently similar: we therefore choose to keep only Influence. Influence, Connections and Exposure are sufficiently different from each other, and from CC.

For case studies, we select and illustrate four cases in the Network and Communications domain: (a) authors with high Influence but low Citation Count (Table 4.13); (b) authors with high Exposure but low Influence (Table 4.14); (c) authors with high Influence but low Connections (Table 4.15); and (d) authors with high Connections but low Influence (Table 4.16).

(a) is the reason why we need Influence. As shown in Table 4.13, some very influential and important people (but probably has not obtained a lot of paper citations) are listed, e.g., Robert Elliot Kahn, one of the co-inventors of the TCP/IP Protocol; and Nathaniel S. Borenstein, one of the designers of the MIME protocol. (b) is the reason for keeping Exposure. (c) and (d) are just the opposite cases. People listed in Table 4.15 are influential but do not "like" having much co-authorship, e.g., Radia J. Perlman, the inventor of the spanning tree protocol (STP); and Robert G. Gallager, known for his work on information theory. On the contrary, authors in Table 4.16 have very powerful sociability in the community and favor of co-authoring

Table 4.13: Examples for High Influence and Low CC

Author	Influence	#Citation
Robert Elliot Kahn	A	C
J. M. Wozencraft	A	C
Jean-Jacques Werner	A	C
David G. Messerschmitt	A	C
Nathaniel S. Borenstein	A	C
James L. Massey	A	C
W. T. Webb	A	C
Takashi Fujio	A	D
Martin L. Shooman	A	D
Sedat Olcer	A	D
Massimo Marchiori	A	D
Roger A. Scantlebury	A	D

Table 4.14: Examples for High Exposure and Low Influence

Author	Influence	Exposure
Achille Pattavina	C	A
Herwig Bruneel	C	A
Yigal Bejerano	C	A
Torsten Braun	C	A
Kenneth J. Turner	C	A
Ioannis Stavrakakis	C	A
Emilio Leonardi	C	A
Luciano Lenzini	C	A
Dmitri Loguinov	C	A
Romano Fantacci	C	A
Hossam S. Hassanein	C	A
Azzedine Boukerche	C	A

with others. We take author named “Athanasios V. Vasilakos” as an example. From 1989-2011, he has totally published 172 papers and collaborated with 282 different collaborators. The yearly publication data is shown in Fig. 4.10. We can see from Fig. 4.10(a) and Fig. 4.10(b), the number published papers and the number of distinct collaborators per year has a rapid increase after year 2005. In the mean time, the average number of

Table 4.15: Examples for High Influence and Low Connections

Author	Influence	Connections
Robert G. Gallager	A	C
Tony, Tong Lee	A	C
Anthony S. Acampora	A	C
David Cheriton	A	C
Jonathan D. Rosenberg	A	C
Jack H. Winters	A	C
Sergio Verdu	A	C
Robert A. Scholtz	A	C
David Mills	A	D
Raymond Yeung	A	D
Radia J. Perlman	A	E

Table 4.16: Examples for High Connections and Low Influence

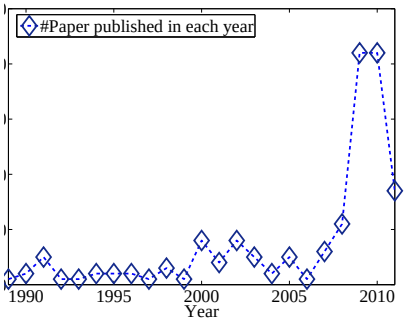
Author	Influence	Connections
Athanasios V. Vasilakos	D	A
Leonard Barolli	D	A
Christos Bouras	D	A
Edmundo Monteiro	D	A
Merouane Debbah	D	A
Djamel Fawzi Hadj Sadok	D	A
Marcelo Dias De Amorim	D	A
Hyunseung Choo	D	A
Han-chieh Chao	D	A
Madjid Merabti	D	A

coauthors for each paper is also gradually increasing with time, Fig. 4.10(c).

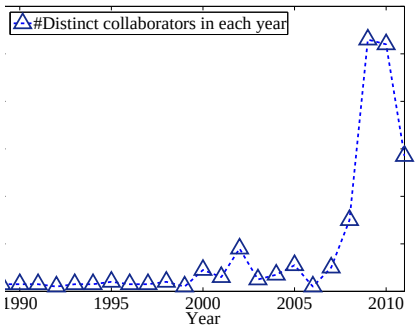
4.3.4 Relation of Ranking to Publication Years

Finally, we are curious to find out the relationship between how an author ranked and his/her first (or last) year of publication. Fig. 4.11 plots the authors' Influence ranks against their first year of publication.

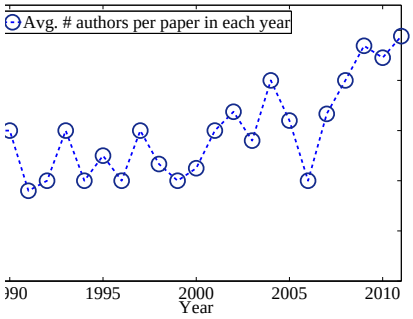
It is worth noting that it takes time to build up Influence.



(a) #Paper published every year



(b) #Distinct collaborators every year



(c) Avg.# coauthors per paper every year

Figure 4.10: The detailed yearly publication information of the example of case (d).

Authors ranked as "A" in Influence started publishing in the 1990s or earlier; "B" authors started publishing in the early 2000s or earlier, and so on (here applies the contribution based letter grading assignment).

Next, we plot an author's last year of publication against Influence (Fig. 4.12), and then against Citation Count (Fig. 4.13), for comparison. Note, for CC, the high ranking people are mostly still active, because we have been seeing paper and citation inflation over years. For Influence, however, there is more *memory*, in the sense that more people who are no longer active also enjoy high Influence. This is because an author's influence propagates, by definition of the Influence metric.

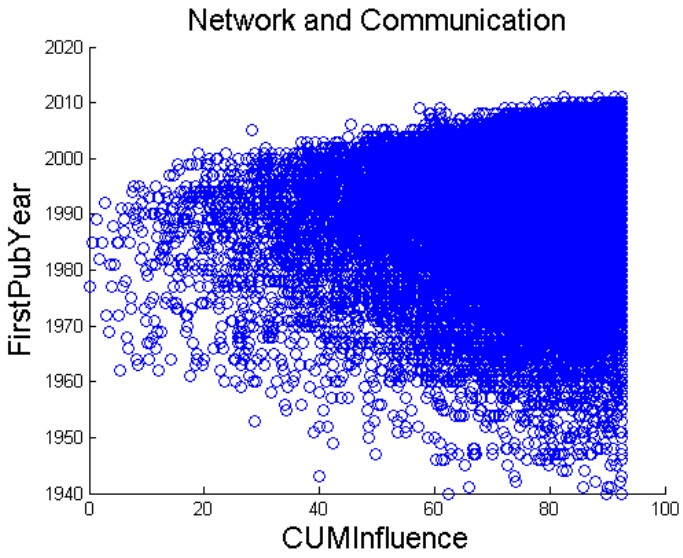


Figure 4.11: Comparison between Influence and the year of first publication.

4.3.5 Ranking Institutions

In Table 4.17, we illustrate the possibility of institutional ranking according to authors' rankings in various metrics. We select 30 well-known universities and apply two counting granularities on authors' letter grades of overall "Computer Science" rankings of three metrics, Citation Counts (CC), Influence (Inf) and Exposure (Exp).

We find that the ranking results by different metrics are similar at the institution level. The noise at the author ranking results are cancelled out to a certain extent after they are aggregated for scores. When we use the two granularities: (1) count the number of authors assigned with "A" and (2) compute the total score, counting "A"=1, "B"=0.5, "C"=0.25 and "D"="E"=0, for method (2), the size of an institution is influential; whereas for method (1), smaller schools also have a chance to rank very high. For example, in Table 4.17, Princeton University is ranked 28th by method (2), but 13th by only counting the number of "A" authors, i.e. by method (1).

Next we show three sets of similarity study between differ-

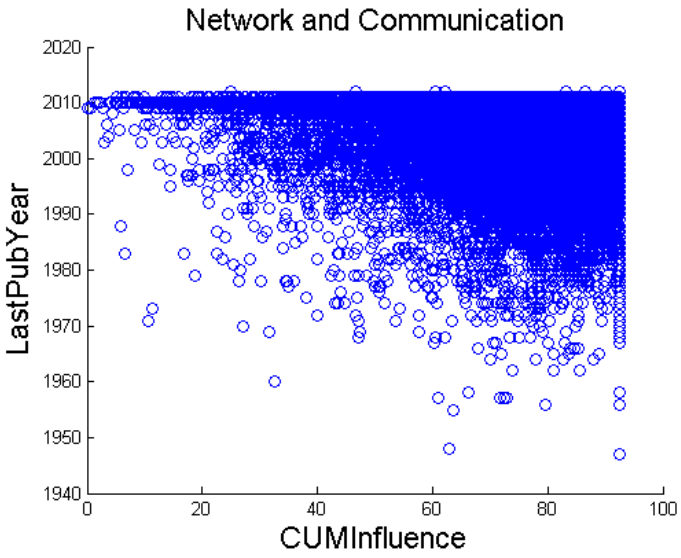


Figure 4.12: Comparison between Influence and the year of last publication.

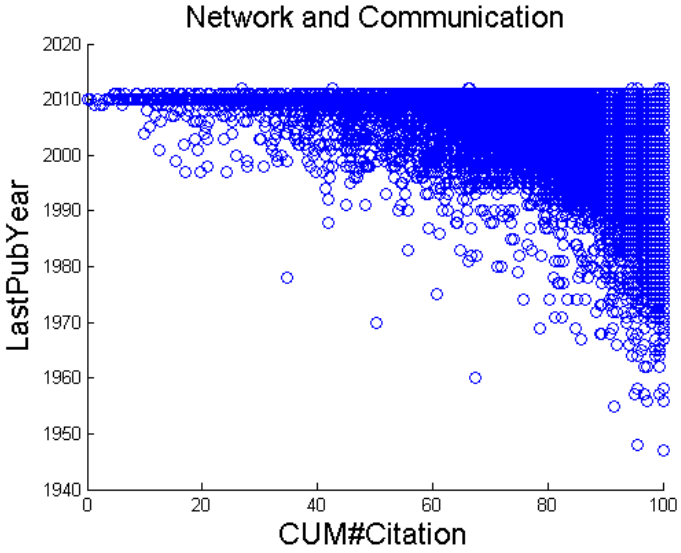


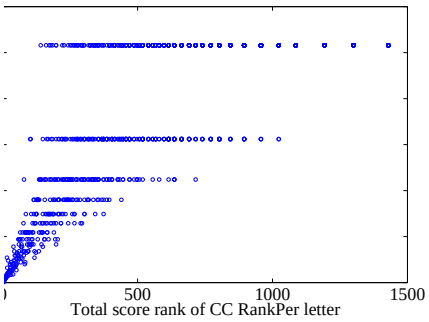
Figure 4.13: Comparison between Citation Count and the year of last publication.

ent institution ranking results, mainly focused on three selected metrics: Citation Count (CC), Influence (Inf) and Exposure (Exp). In the first set, we compare the ranking results at two different granularities, count number of "A" authors versus compute total score, based on the rank percentile based letter grades, as shown in Fig. 4.14(b).

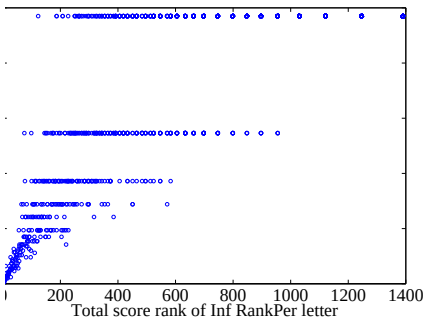
Table 4.17: Illustration of Institution Rankings on 30 selected top universities of three metrics (#Citations, Influence and Exposure) at two granularities based on authors' overall ranking in "Computer Science" domain.

Institution Name	Total Score Rank			#A Rank		
	CC	Inf	Exp	CC	Inf	Exp
Massachusetts Institute of Technology	2	1	2	1	1	2
Carnegie Mellon University	1	2	1	2	2	1
Stanford University	3	3	3	4	4	4
University of California Berkeley	4	4	4	3	3	3
University of Illinois Urbana Champaign	5	5	5	6	6	5
University of Southern California	6	6	7	5	5	6
Georgia Institute of Technology	7	6	6	8	7	8
University of California San Diego	11	8	9	7	8	7
University of Washington	10	9	14	10	9	13
University of Maryland	8	9	8	9	11	8
University of California Los Angeles	12	11	11	12	10	12
University of Texas Austin	9	11	10	11	14	10
University of Michigan	13	13	11	14	14	16
Cornell University	15	14	15	13	11	15
University of Cambridge	16	15	21	17	17	22
Columbia University	17	16	19	21	20	17
University of Wisconsin Madison	20	17	28	18	18	22
University of Toronto	18	18	16	16	16	13
The French National Institute for Research in Computer science and Control	14	19	11	24	26	22
University of Pennsylvania	22	20	27	21	21	22
Rutgers, The State University of New Jersey	23	21	22	29	18	22
Swiss Federal Institute of Technology Zurich	18	22	25	23	26	28
Harvard University	30	23	39	33	31	35
University of California Irvine	25	24	23	19	21	20
Purdue University	21	25	18	19	31	17
University of Minnesota	25	25	24	27	31	22
University of Massachusetts	24	25	26	31	36	30
Princeton University	27	28	29	14	13	17
Technion Israel Institute of Technology	31	29	19	24	23	10
University of Edinburgh	29	30	29	31	40	35

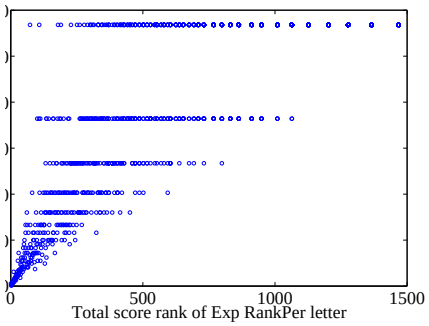
In the second set (Fig. 4.15), we investigate how the authors' letter grading methods (rank percentile based versus contribu-



(a) Citation Count



(b) Influence



(c) Exposure

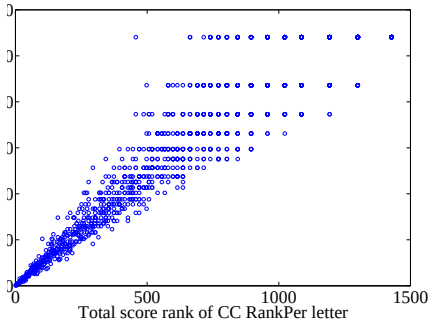
Figure 4.14: Comparison on institution ranking results between two granularity methods: (counting number of As, y-axis) versus (counting "A"=1, "B"=0.5, "C"=0.25 for total score, x-axis) for three metrics: CC, Inf and Exp, according to authors' rank percentile based letter grades.

tion based) affect the total scores (granularity method (2)) as well as the institution rankings.

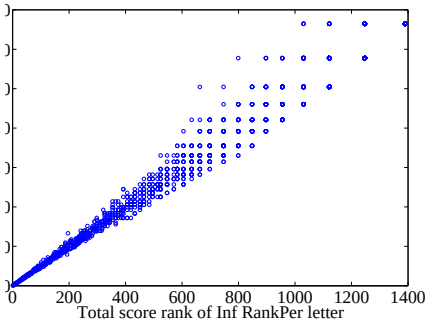
In the last set, we compare three metrics (Inf vs. CC in Fig. 4.16(a), Inf vs. Exp in Fig. 4.16(b)) while the rank percentile based letter grading scheme and the granularity method (2) of computing total score are used.

According to the above comparison results (Figures 4.14, 4.15 and 4.16), we make several observations:

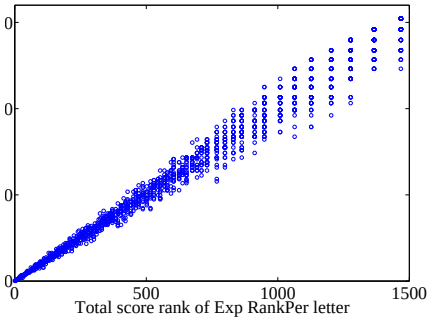
i. As shown in Fig. 4.14, for those highly ranked institutions (e.g. above 100th), the ranking results of the two granularities are very close. In addition, as mentioned before, when



(a) Citation Count



(b) Influence

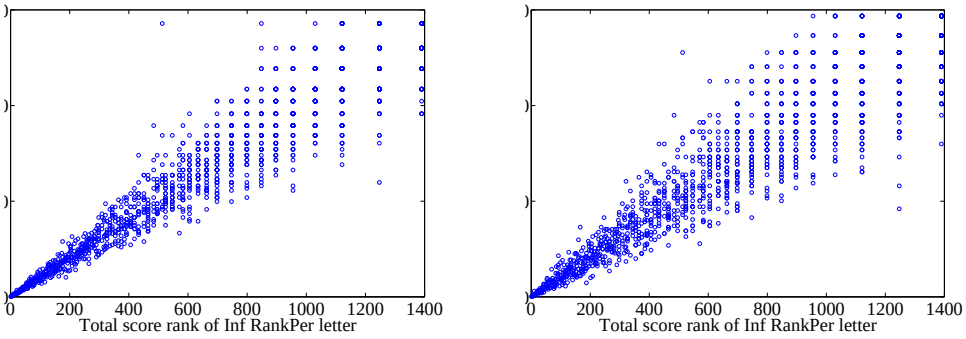


(c) Exposure

Figure 4.15: Comparison on institution ranking results between rank percentile based (x-axis) versus contribution based (y-axis) letter grading methods, using granularity method (2) for three metrics, CC, Inf and Exp.

the total scores (by counting "A"=1, "B"=0.5, "C"=0.25) are same, granularity method (1) can indicate the ratio of authors earning letter "A" (e.g. Princeton University in Table 4.17). On the other hand, counting the number of "A" authors only is ineffective in distinguishing institutions ranked below 100th (the number of institutions with the same number of "A" authors locate on horizontal lines).

ii. As shown in Fig. 4.15, although the rank percentile based and contribution based letter grading methods make pronounced differences on author rankings, they produce very



(a) Influence vs. Citation Count (b) Influence vs. Exposure

Figure 4.16: Comparison on institution ranking results among three metrics: CC, Inf and Exp, according to rank percentile based letter grading results using granularity method (2).

similar results on institution rankings.

iii. As shown in Fig. 4.16, institutions ranked above 100th have similar ranking results for these three metrics (CC, Inf and Exp); however, the points are spread out largely for those ranked below 100th under different metrics. This again validates the effectiveness of the definitions of the various metrics with practical interpretations.

Finally, we compare our institution ranking approach to the three established ranking systems. We show the top 30 universities in "Computer Science" domain ranked by each of these systems, together with the ranking results by ours (based on total score of Influence metric):

a) US News Ranking - The Best Graduate Schools in Computer Science ranked in 2010 [16]. Results are shown in Table 4.18.

b) The QS World University Rankings By Subject 2013 - Computer Science & Information Systems [17]. Results are shown in Table 4.19.

c) The Academic Ranking of World Universities (ARWU by SJTU) 2012 in Computer Science [18]. Results are shown in Table 4.20.

As in Tables 4.18-4.20, we find that the calculation of the overall score is the key factor leading to the deviation of the ranking results among different systems. In particular, the US New ranking system applied a subjective based approach [16] to calculate the total scores for each university. The QS ranking system calculated the overall score in the "Computer Science & Information Systems" subject based on the four objective factors: "Academic Reputation", "Employer Reputation", "Citations per Paper" and "H-index Citations" [17]. The ARWU ranking system, on the other hand, consider the overall score in "Computer Science" domain as the weighted average of the five metrics: "Alumni Turing Awards (10%)", "Staff Turing Award (15%)", "Highly Cited Researchers (25%)", "Papers indexed in SCI (25%)" and "Papers Published in Top Journals (25%)" [18]. Because of these factors (considering more reputation and recent work), the results of the three ranking systems tend to be quite volatile - the top universities change quite a bit from year to year. In our case of using total score of Influence metric, we are at least more stable and pure.

As Microsoft Libra also provides the institution ranking services [4], we make another comparison and the results are shown in Table 4.21. Since we are using the same dataset for calculation, it is not surprising that the ranking results are very similar.

Table 4.18: Top 30 Universities of "US News Ranking - The Best Graduate Schools in Computer Science ranked in 2010", compared to ours (total score of Influence metric)

University Name	Score	USNews	Inf TS
Carnegie Mellon University	5.0	1	2
Massachusetts Institute of Technology	5.0	1	1
Stanford University	5.0	1	3
University of California Berkeley	5.0	1	4
Cornell University	4.6	5	14
University of Illinois Urbana Champaign	4.6	5	5
University of Washington	4.5	7	9
Princeton University	4.4	8	28
University of Texas Austin	4.4	8	11
Georgia Institute of Technology	4.3	10	6
California Institute of Technology	4.2	11	33
University of Wisconsin Madison	4.2	11	17
University of Michigan	4.1	13	13
University of California Los Angeles	4.0	14	11
University of California San Diego	4.0	14	8
University of Maryland	4.0	14	9
Columbia University	3.9	17	16
Harvard University	3.9	17	23
University of Pennsylvania	3.9	17	20
Brown University	3.7	20	42
Purdue University	3.7	20	25
Rice University	3.7	20	47
University of Massachusetts	3.7	20	25
University of North Carolina-Chapel Hill	3.7	20	42
University of Southern California	3.7	20	6
Yale University	3.7	20	53
Duke University	3.6	27	59
Johns Hopkins University	3.4	28	44
New York University	3.4	28	33
Ohio State University	3.4	28	40
Pennsylvania State University	3.4	28	46
Rutgers, The State University of New Jersey	3.4	28	21
University of California Irvine	3.4	28	24
University of Virginia	3.4	28	68

Table 4.19: Top 30 Universities of "The QS World University Rankings By Subject 2013 - Computer Science & Information Systems", compared to ours (total score of Influence metric)

University Name	Score	QS	Inf TS
Massachusetts Institute of Technology	96.7	1	1
Stanford University	92.1	2	3
University of Oxford	92.0	3	21
Carnegie Mellon University	90.5	4	2
University of Cambridge	89.8	5	15
Harvard University	88.4	6	23
University of California Berkeley	88.0	7	4
National University of Singapore	87.2	8	57
Swiss Federal Institute of Technology Zurich	87.1	9	22
University of Hong Kong	84.0	10	165
Princeton University	83.7	11	28
The Hong Kong University of Science & Technology	83.6	12	113
The University of Melbourne	83.4	13	82
University of California Los Angeles	82.1	14	11
University of Edinburgh	81.5	15	30
University of Toronto	81.0	16	18
École Polytechnique Fédérale de Lausanne	80.2	17	36
Imperial College London	79.7	18	35
The Chinese University of Hong Kong	79.5	19	94
The University of Tokyo	79.4	20	50
Australian National University	78.9	21	107
Nanyang Technological University	78.5	22	91
University College London	78.0	23	47
The University of Sydney	77.9	24	146
The University of Queensland	77.8	25	107
Cornell University	77.6	26	14
Tsinghua University	77.5	27	107
University of Waterloo	77.5	27	32
The University of New South Wales	77.3	29	102
The University of Manchester	77.1	30	45

Table 4.20: Top 30 Universities of "The Academic Ranking of World Universities (ARWU by SJTU) 2012 in Computer Science, compared to ours (total score of Influence metric)

University Name	Score	SJTU	Inf TS
Stanford University	100	1	3
Massachusetts Institute of Technology	93.8	2	1
University of California Berkeley	85.3	3	4
Princeton University	78.7	4	28
Harvard University	77.7	5	23
Carnegie Mellon University	71.8	6	2
Cornell University	71.2	7	14
University of California Los Angeles	69.2	8	11
University of Texas Austin	68.3	9	11
University of Toronto	63.6	10	18
California Institute of Technology	63.5	11	33
Weizmann Institute of Science	63.3	12	89
University of Southern California	63.0	13	6
University of California San Diego	61.8	14	8
University of Illinois Urbana Champaign	61.7	15	5
University of Maryland	60.1	16	9
University of Michigan	58.9	17	13
Technion-Israel Institute of Technology	57.8	18	29
University of Oxford	56.7	19	31
Purdue University	54.5	20	25
University of Washington	54.2	21	9
Columbia University	53.8	22	16
Rutgers, The State University of New Jersey	53.5	23	21
Georgia Institute of Technology	53.0	24	6
Swiss Federal Institute of Technology Zurich	52.7	25	22
The Hong Kong University of Science & Technology	52.6	26	113
The Hebrew University of Jerusalem	52.5	27	77
Yale University	51.4	28	53
Tel Aviv University	50.9	29	36
The Chinese University of Hong Kong	50.7	30	94

Table 4.21: Top 30 Universities ranked by Libra in "Computer Science" domain, compared to ours (total score of Influence metric)

University Name	Field Rate	Libra	Inf TS
Stanford University	418	1	3
Massachusetts Institute of Technology	408	2	1
University of California Berkeley	404	3	4
Carnegie Mellon University	325	4	2
University of Illinois Urbana Champaign	268	5	5
Cornell University	260	6	14
University of Southern California	256	7	6
University of Washington	256	7	9
University of California San Diego	253	9	8
Princeton University	252	10	28
University of Texas Austin	248	11	11
University of California Los Angeles	243	12	11
University of Maryland	238	13	9
Georgia Institute of Technology	229	14	6
University of Michigan	224	15	13
University of Toronto	222	16	18
University of Cambridge	214	17	15
Harvard University	214	17	23
University of Wisconsin Madison	209	19	17
Columbia University	202	20	16
University of Pennsylvania	201	21	20
University of California Irvine	199	22	24
Rutgers, The State University of New Jersey	197	23	21
University of Oxford	197	23	31
University of Minnesota	195	25	25
Swiss Federal Institute of Technology Zurich	190	26	22
The French National Institute for Research in Computer science and Control	189	27	19
California Institute of Technology	189	27	33
Brown University	189	27	42
University of Massachusetts	189	27	25

4.3.6 Discussions

We want to have some discussions on all kinds of ranking results conducted and presented on our website.

Ranking results vs. research performance The interpretation of the ranking results is essentially relevant to the definition of each metric. For example, the "Connections" metric reveals how actively authors collaborate with others. When author A gets higher "Connections" value/rank than author B, we can only say that author A is "better" than B, in terms of the activeness of the collaboration behaviors.

Since there is no standard or clear definition on research performance, we shall be careful when trying to apply our ranking results to evaluate authors' research performance.

Sometimes, we may consider the ranking results of the "Influence" metric as "good" candidates for representing the research performance, while questioning on the rationale of applying "Connections" results. This is because that we have implicitly considered the definition of research performance to be much closer to that of the "Influence" metric, rather than the "Connections" metric.

To summarize, we are trying our best to keep everything objective, e.g., how we define various metrics as well as how we calculate the ranking results and so on. However, it inevitably involves subjective opinions when applying these results to reflect research performance.

The overall ranking results We have conducted pair-wise similarity studies among these metrics. Some are very similar pairs, e.g., (Influence, Follower) and (Influence, Citation Value). Some are very different pairs such as (Influence, Connections). The pairs like (Influence, Citation Count) and (Influence, Exposure)

are in between.

The similarity study helps to tell the ranking results of which two metrics are heavily overlapped (e.g., Influence vs. Follower); or overlapped to some extent (e.g., Influence vs. Exposure).

In this thesis, we have not studied how to conduct overall ranking results by combining the results of multiple metrics and we will consider it as our future works. When dealing with this, here are some challenges:

1) How to select typical metrics from all different metrics for combining.

2) How to avoid "double" counting when two metrics are similar to some extent.

□ **End of chapter.**

Chapter 5

MYE: Missing Year Estimation in ASN

Summary

In this chapter, we describe the missing year estimation problem in the academic social network. In Section 5.1, we introduce the estimation methodology. We present the data sets we used and the experiment results in Section 5.2.

5.1 Methodology

In this section, we first revisit the three types of the academic social networks (discussed in Chapter 3) that we are dealing with. Next we introduce the notations which are complementary to those listed in Table 3.1. For each network type, we propose three corresponding missing year estimation (MYE) algorithms, with different complexity levels.

5.1.1 Network types revisit and notations

The network types and definitions of ASN have been discussed in Chapter 3. Here we briefly revisit the related concepts.

In the MYE problem, we are mainly interested in two node types: papers and authors; and two edge types: paper citations and paper authorships, which induce three academic social networks:

a) Paper citation network, denoted by a directed graph $G_P = (V_P, E_P)$, where V_P is the set of papers and E_P is the set of directed citation links: $\forall e \in E_P, e = (t, f)$, where $t, f \in V_P$, meaning paper t is cited by paper f.

b) Paper authorship network, denoted by $G_{AP} = (V_A \cup V_P, E_{AP})$, where V_A is the set of authors, V_P is the set of papers and edges in the set E_{AP} connecting authors to their produced papers (authorship). Hence G_{AP} is a bipartite graph and we have $\forall e = (a, p) \in E_{AP}$, where $a \in V_A$ and $p \in V_P$.

c) Heterogenous network consisting of both paper citation network and paper authorship network, denoted by $G = G_P \cup G_{AP} = (V_A \cup V_P, E_P \cup E_{AP})$.

   Note we are only interested in two node types: papers and authors in the MYE problem, therefore, the definition of the heterogenous network here $G = G_P \cup G_{AP} = (V_A \cup V_P, E_P \cup E_{AP})$ is different to the super-graph $G_{ASN} = (V, E) = (V_P \cup V_A \cup V_V \cup V_S, E_P \cup E_{AP} \cup E_{VP} \cup E_{AS})$ defined in Chapter 3 (hence, a different notation is used).

Papers are further categorized into two exclusive sets: with known year information V_P^K and unknown (missing) year information V_P^U. Hence we have $V_P = V_P^K \cup V_P^U$ and $V_P^K \cap V_P^U = \emptyset$. The remaining notations are listed in Table 5.1:

Table 5.1: List of notations complementary to Table 3.1.

V_P^K	paper set with known year information.
V_P^U	paper set with unknown (missing) year information.
$Y(p),\ \forall\ p \in V_P$	the real publishing year of paper p, note: $\forall\ p^U \in V_P^U, Y(p^U)$ is only used for validation purpose.
$\hat{Y}(p^U),\ \forall\ p^U \in V_P^U$	the estimation result for the missing year paper p^U.
$w(p,q),\ \forall p,q\ \in V_P$	the Consistent-Coauthor-Count between two papers, $w(p,q) = w(q,p) = \lvert A(p) \cap A(q)\rvert$
$\Omega(p),\ \forall\ p \in V_P$	the Consistent-Coauthor-Pair set of a paper $p \in V_P$, $\Omega(p) = \{q \lvert q \in V_P \text{ and } w(p,q) > 1\}$
$AW\_Min(a),\ AW\_Max(a)$, $\forall\ a \in V_A$	the lower and upper bounds of the active publishing time window of author a.
$\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)$, $\forall\ p^U \in V_P^U$	the lower and upper bounds of the year estimation window, derived in the paper citation network G_P.
$\hat{Y}_{AMin}(p^U),\ \hat{Y}_{AMax}(p^U)$, $\forall\ p^U \in V_P^U$	the lower and upper bounds of the year estimation window, derived in the paper authorship network G_{AP}
$\hat{Y}_{GMin}(p^U),\ \hat{Y}_{GMax}(p^U)$, $\forall\ p^U \in V_P^U$	the lower and upper bounds of the year estimation window, derived in the heterogenous network G

5.1.2 MYE for paper citation network G_P

We first look at a simple example of the missing year estimation problem in the paper citation network, shown in Fig. 5.1. In this example, there are 12 papers $(a - l)$ and 10 citation edges. 5 papers (a, b, e, i, j) have no year information (i.e. $\in V_P^U$) and the other 7 papers (c, d, f, g, h, k, l) have publishing years (i.e. $\in V_P^K$). Later on, we will use this example to demonstrate the three MYE algorithms designed for the citation network G_P.

The main idea of estimating the missing years in the citation network G_P is to make use of paper citing activities, stated as Assumption 5.1, together with the available information: a) the year information of those known papers; b) the citation relationships (edges of the G_P).

Assumption 1 *Normally*[1]*, a paper can only cite those papers*

[1]Since the exceptions are rare, we believe that ignoring such exceptions is reasonable and does not harm our algorithm design.

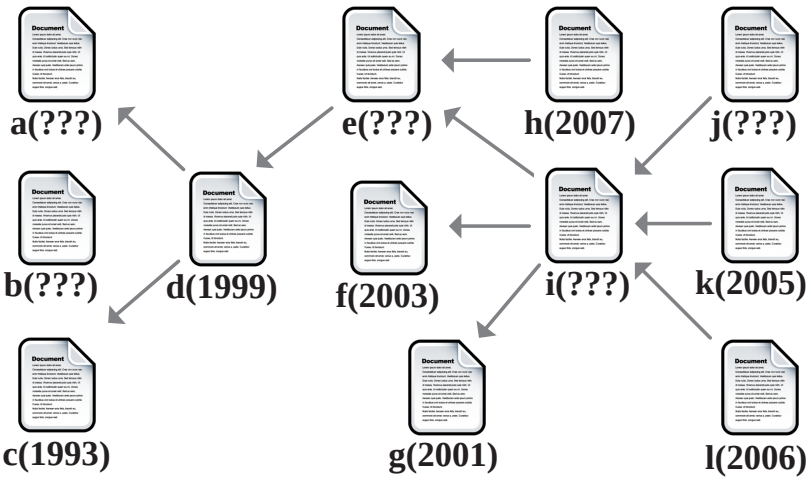


Figure 5.1: A simple example of a citation network with 12 papers $(a-l)$, where papers (a, b, e, i, j) are $\in V_P^U$ and the remaining (c, d, f, g, h, k, l) are $\in V_P^K$.

published before it, i.e., Eq. (5.1) is satisfied:

$$Y(t) \leq Y(f), \ \ \forall \ \ e = (t, f) \in E_P, \ \ t, f \in V_P. \tag{5.1}$$

Assumption 5.1 provides the way to determine either a possible upper bound of the target paper's missing year when it is cited by a known year paper (i.e., $t \in V_P^U$ and $f \in V_P^K$); or a possible lower bound of the target paper's missing year when it cites a known year paper (i.e., $t \in V_P^K$ and $f \in V_P^U$). For example, in Fig. 5.1, when we look at paper a (missing year) and d (published in 1999) with a citation link from d to a, we take 1999 as one possible upper bound of a's publishing year, i.e., $Y(a) \leq 1999$. Similarly, when we look at paper d and e, we get a lower bound of the real publishing year of e, i.e., $1999 \leq Y(e)$.

Following this logic, the missing year estimation task can be separated into two steps: (1) deriving the possible year estimation window (two bounds); (2) calculating the missing year value based on the derived window.

For each step, we propose two methods with different complexity, the simple ("Sim") version and the advanced ("Adv")

version. In the next three subsections, we will introduce the three algorithms designed for MYE in paper citation network G_P. The three algorithms are different combinations of the two methods in each step, listed in Table 5.2.

Table 5.2: Combination of the two proposed methods in each step, for the three algorithms for MYE in G_P.

Algorithm	Window derivation method	Year value calculation method
G_P-SS	Simple	Simple
G_P-AS	Advanced	Simple
G_P-AA	Advanced	Advanced

Algorithm for MYE in G_P: G_P-SS

We will first introduce the simple method for each of the two steps, and then show how G_P-SS works, by demonstrating the results on the example shown in Fig. 5.1.

Simple Window Derivation Method: The simple version of the window (bounds) derivation method only involves "one round" (or in a "direct" manner), which means: (1) spatially, we only consider those papers that are one-hop to the target missing year paper; (2) temporally, we only consider immediate (given) information.

Putting together (1) and (2), mathematically, we are deriving the bounds of the missing year paper $p^U \in V_P^U$ through the subset of the papers: $F(p^U) \cap V_P^K$ (for the lower bound) and $T(p^U) \cap V_P^K$ (for the upper bound) as long as they are not empty. For example, if we look at paper i in Fig. 5.1, then only f and g (one-hop away from i and with year information) are used for deriving the lower bound, while only k and l for the upper bound. Intuitively, when there are multiple bounds, we will take the tightest one by applying Eqs. (5.2) and (5.3):

$$\hat{Y}_{CMin}(p^U) = \max_{f \in F(p^U) \cap V_P^K} Y(f), \text{ if } F(p^U) \cap V_P^K \neq \emptyset;$$

$$= \quad -\infty, \quad \text{otherwise;} \tag{5.2}$$

$$\hat{Y}_{CMax}(p^U) = \min_{t \in T(p^U) \cap V_P^K} Y(t), \text{ if } T(p^U) \cap V_P^K \neq \emptyset;$$

$$= \quad +\infty, \quad \text{otherwise,} \tag{5.3}$$

where $\hat{Y}_{CMin}(p^U)$ denotes the largest possible lower bound of paper p^U and $\hat{Y}_{CMax}(p^U)$ denotes the smallest possible upper bound. Here the $-\infty$ and $+\infty$ have no practical meaning, used just to represent the non-existent bounds. In the real implementation, they can be assigned to some pre-defined constant variables such as "Default_Win_Min" and "Default_Win_Max".

Together with the conditions of non-existent bounds, we thus have four types of possible year estimation windows:

$$\begin{array}{ll} \text{Type-1:} & [\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]; \\ \text{Type-2:} & [\ \hat{Y}_{CMin}(p^U), \quad +\infty \quad); \\ \text{Type-3:} & (\quad -\infty, \quad \hat{Y}_{CMax}(p^U)\ ]; \\ \text{Type-4:} & (\quad -\infty \quad , \quad +\infty \quad). \end{array}$$

Actually, Type-4 window contains no information for estimation, hence we define *Uncovered Paper* to be those missing year papers with a Type-4 estimation window. On the other hand, it is possible to make a proper estimation on the year value for the missing year papers with Type-1, Type-2 or Type-3 estimation window.

Simple Year Value Calculation Method: Based on the derived possible year estimation window for each missing year paper p^U, the next step is to make a guess on its real publishing year. The simple calculation method works in a straightforward way, Eqs. (5.4)-(5.7):

$$\text{Type-1:} \quad \hat{Y}(p^U) = \frac{\hat{Y}_{CMin}(p^U) + \hat{Y}_{CMax}(p^U)}{2}, \tag{5.4}$$

$$\text{Type-2:} \quad \hat{Y}(p^U) = \hat{Y}_{CMin}(p^U), \tag{5.5}$$

$$\text{Type-3:} \quad \hat{Y}(p^U) = \hat{Y}_{CMax}(p^U), \tag{5.6}$$

Type-4: *Uncovered.* (5.7)

In summary, if both bounds exist (Type-1), we take the average of the two bounds, Eq. (5.4) (assuming that $Y(p^U)$ follows any symmetric discrete distribution centered at the middle point of the possible estimation window). If only one bound exists (Type-2 or Type-3), we take the bound value as the calculation result. Otherwise (Type-4), instead of making any random guess, we label it as (*Uncovered*), which means, the year of such paper cannot be estimated properly. Later on, in the performance evaluation part, we will consider the uncovered ratio ($= \frac{\text{Total \# Uncoverd}}{|V_P^U|}$) of all the proposed algorithms as one of the performance metrics

Considering the example in Fig. 5.1, we list both the intermediate and final estimation results conducted by apply G_P-SS in Table 5.3.

p^U in Fig.5.1	a	b	e	i	j
$F(p^U)$	$\emptyset$	$\emptyset$	d	e, f, g	i
$F(p^U) \cap V_P^K$	$\emptyset$	$\emptyset$	d	f, g	$\emptyset$
$T(p^U)$	d	$\emptyset$	h, i	j, k, l	$\emptyset$
$T(p^U) \cap V_P^K$	d	$\emptyset$	h	k, l	$\emptyset$
$\hat{Y}_{CMin}(p^U)$	$-\infty$	$-\infty$	1999	2003	$-\infty$
$\hat{Y}_{CMax}(p^U)$	1999	$+\infty$	2007	2005	$+\infty$
$\hat{Y}(p^U)$	1999	*Uncovered*	2003	2004	*Uncovered*

Table 5.3: The intermediate and estimation results obtained through G_P-SS algorithm running on the example of Fig. 5.1

In Table 5.3, the first row lists all the 5 papers belonging to V_P^U. The second and third rows list the paper set cited by each of the 5 papers, where the third row only contains papers with year information, e.g., for paper i, it cites three papers $F(i) = \{e, f, g\}$ and only two of them have year information, $F(i) \cap V_P^K = \{f, g\}$. The fourth and fifth rows list the papers that cite each of the 5 papers, where the fifth row only contains

papers belonging to V_P^K. The next two rows are the two bounds of the possible estimation window by applying Eqs. (5.2) and (5.3), e.g., $\hat{Y}_{CMin}(i) = \max\{Y(f), Y(g)\} = \max\{2003, 2001\} = 2003$. The last row shows the results derived by the simple year calculation scheme, Eqs. (5.4)-(5.7).

The G_P-SS is simple, quick and easy for both implementation and understanding, but its limitation is also obvious. It has not fully utilized the available information, which leaves with a high uncovered ratio (= 2/5 shown in Table 5.3) and looser bounds. Considering this question, can the information (derived bounds or estimated results after running G_P-SS) of paper i be useful for its missing year neighbor papers j and e? The answer is positive and the next algorithm is designed for dealing with this.

Algorithm for MYE in G_P: G_P-AS

Comparing to G_P-SS, G_P-AS applies the same simple version of year value calculation method, Eqs. (5.4)-(5.7), but an advanced method for window derivation with information propagations.

A quick way of extending G_P-SS is to simply repeat running it. In this way, the estimated result for a missing year paper (e.g. i in Fig. 5.1) in the previous rounds can be used to derive bounds for its neighbor missing year paper (e.g. j and e in Fig. 5.1) in the subsequent rounds. However, since the estimated year result for i can be inaccurate, this kind of repeating will definitely propagate and even amplify the inaccuracy.

Advanced Window Derivation Method: Generally in G_P, for each citation edge linking two papers, there can be three possible conditions: (a) both papers have year information ($\in V_P^K$); or (b) both papers are missing year ($\in V_P^U$); or (c) one has year information while the other has not. The limitation of simple window derivation method is that it only works under condition (c). By rephrasing Eq. (5.1) as Eq. (5.8), the advanced window derivation method relaxes this limitation without induc-

ing any inaccuracy in the propagation.

$$\hat{Y}_{CMin}(t) \leq Y(t) \leq Y(f) \leq \hat{Y}_{CMax}(f). \tag{5.8}$$

The rationale behind Eq. (5.8) is to extend the bound transmission rule between two missing year papers: (a) if $\hat{Y}_{CMin}(t)$ exists, it is also a lower bound of f; (b) if $\hat{Y}_{CMax}(f)$ exists, it is also an upper bound of t. The pseudo code of the advanced window derivation method is included below.

Algorithm 1 The pseudo code of advanced window derivation method

```
1: repeat
2:   UpCnt ← 0;
3:   for all e = (t, f) ∈ E_P, t, f ∈ V_P do
4:     f_CMin_Before← Ŷ_CMin(f);
5:     t_CMax_Before← Ŷ_CMax(t);
6:     if t, f ∈ V_P^U then
7:       Ŷ_CMin(f) ← max{Ŷ_CMin(f), Ŷ_CMin(t)};
8:       Ŷ_CMax(t) ← min{Ŷ_CMax(t), Ŷ_CMax(f)};
9:     else if t ∈ V_P^K, f ∈ V_P^U then
10:      Ŷ_CMin(f) ← max{Ŷ_CMin(f), Y(t)};
11:    else if t ∈ V_P^U, f ∈ V_P^K then
12:      Ŷ_CMax(t) ← min{Ŷ_CMax(t), Y(f)};
13:    end if
14:    /* Check update counts. */;
15:    if Ŷ_CMin(f) ≠ f_CMin_Before then
16:      UpCnt ← UpCnt +1;
17:    end if
18:    if Ŷ_CMax(t) ≠ t_CMax_Before then
19:      UpCnt ← UpCnt +1;
20:    end if
21:  end for
22: until UpCnt = 0; /* When no update happens, loop ends. */
```

In Algorithm 1, we first initialize a local variable "UpCnt" which records the total number of bound updates in each loop (Line 2). Lines 3-21 are steps in a loop of processing each citation link of G_P, where Lines 9-13 are the same as the simple window derivation method, Eq. (5.2) and Eq. (5.3), while Lines

6-8 are the essential part that differs from the simple version (also the implementation of the two bound transmission rules of Eq. (5.8)).

In Table 5.4, we list both the intermediate and estimation results of applying G_P-AS on the example of Fig. 5.1.

p^U in Fig.5.1	Round 1	Round 2	Round 3	$\hat{Y}(p^U)$
a	$(-\infty, 1999)$	$(-\infty, 1999)$	$(-\infty, 1999)$	1999
b	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	*NotCovered*
e	$(1999, 2007)$	$(1999, 2005)$	$(1999, 2005)$	2002
i	$(2003, 2005)$	$(2003, 2005)$	$(2003, 2005)$	2004
j	$(-\infty, +\infty)$	$(2003, +\infty)$	$(2003, +\infty)$	2003
	UpCnt = 5	UpCnt =2	UpCnt =0	

Table 5.4: The intermediate and estimation results of applying G_P-AS on the example shown in Fig. 5.1

From Table 5.4, we can see that the advanced window estimation takes two rounds (no updates happen in round 3) and the last column is the year estimation results by applying the simple year value calculation method based on the derived bounds. Comparing to Table 5.3, the improvement is obvious even for this simple example: (1) paper j is no longer labeled as (*Uncovered*, hence, the uncovered ratio decreases to 1/5; (2) paper e gets a tighter possible estimation window.

So far, we are doing our best to deal with the possible window derivation problem (apparently, paper b in Fig. 5.1 has no chance to get a good estimate, and we will discuss the relationship between the uncovered ratio and the structure of the given citation graph G_P mathematically in Section 5.2). In the next algorithm, we investigate how the year value calculation method can be further improved.

Algorithm for MYE in G_P: G_P-AA

Given the derived estimation window $[\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]$ for a missing year paper p^U, recall Eqs. (5.4)-(5.7)(how simple year

value calculation method works): (1) if both bounds exist (Type-1), the calculation result is the mean of the two bounds; or (2) if only one bound exists (Type-2 or Type-3), the calculation result equals to the value of the existing bound; or (3) if neither bound exists, then the paper is labeled as *Uncovered*, representing no proper estimation result.

The year estimation results for cases (1) and (2) affect the accuracy metrics, such as Mean Absolute Error (MAE), while case (3) only affects the uncovered ratio, irrelevant to other metrics. For case (1), it is rational to take the average of the two bounds, since the citing-to activity and cited-by activity can be considered symmetric. But for case (2), it needs more investigation. The physical interpretation of case (2) is based on the assumption that the missing year paper has the same publishing time as the earliest paper that cites it (the upper bound exists), or the latest paper cited by it (the lower bound exists). In reality, this seldom happens. The best guess for (Type-2 or Type-3) window case may be correlated to the bound value, not just a fixed distance to the bound (e.g. the simple calculation method takes a fixed zero distance). Therefore, the solution for this problem is to find a proper function $\hat{y}(p^U) = d(WinType(p^U), BoundVal(p^U))$ to calculate $\hat{y}(p^U)$ for each missing year paper p^U, based on its derived estimation window type, denoted by $WinType(p^U)$ (which takes value of either Type-2 or Type-3), and the value of bound, denoted by $BoundVal(p^U)$.

To achieve this, we need a separate data set, denoted by $\mathcal{T}$, containing a series of 3-tuples, $t = \{y_t, WinType_t, BoundVal_t\} \in \mathcal{T}$ for training purpose. Each 3-tuple data corresponds to a missing year paper t in this training set, where y_t is the validated real publishing year, $WinType_t$ is the derived estimation window type and $BoundVal_t$ is the bound value. If we denote $\mathcal{T}_{p^U}$ as the subset of $\mathcal{T}$ with respect to p^U and $\mathcal{T}_{p^U} = \{t | t \in \mathcal{T}, WinType_t =$

$WinType(p^U), BoundVal_t = BoundVal(p^U)\}$, then we get the following form for $d(\cdot)$ corresponding with $\mathcal{T}$:

$$\hat{y}(p^U) = d_{\mathcal{T}}(WinType(p^U), BoundVal(p^U)) = \frac{\sum_{t \in \mathcal{T}_{p^U}} y_t}{|\mathcal{T}_{p^U}|}, \quad (5.9)$$

where $|\mathcal{T}_{p^U}|$ is the element count of the set $\mathcal{T}_{p^U}$.

The idea of Eq. (5.9) is to take the expectation of the real publishing years of those papers having the same window type and bound value as P^U in the training set $\mathcal{T}$. However it is not trivial to find a proper training set satisfying: (1) a citation graph with similar property and structure to the given G_P; (2) the $BoundVal$ of this training set covers a wider range than that of $BoundVal(p^U), \forall p^U \in V_P^U$.

Advanced Year Value Calculation Method: We first propose a way to find a suitable training set $\mathcal{T}$ which can satisfy both (1) and (2) mentioned above. After that, the estimation results can be calculated through Eq. (5.9).

One of the most suitable training sets is just inside the given citation network G_P. In fact, each paper with known year ($\forall p^K \in V_P^K$) can also be used to derive a possible estimation window (by pretending itself to be a missing year paper). Consider the example in Fig. 5.1, for paper $d(1999)$, the simple window derivation method generates $[1993, +\infty)$. Since this is independent of deriving windows for missing year papers, these two procedure can be merged together to save the running time. The modified advanced window derivation method for G_P-AA is shown in Algorithm 2.

Comparing to Algorithm 1, the pseudo code in Algorithm 2 has added 4 lines (Lines 11, 13, 16 and 17) for preparing the training set. These four lines are still satisfying Eq. (5.8) for avoiding inducing inaccuracy, but the information is propagated towards papers in set V_P^K. Table 5.5 list the intermediate and final results of the example training set $\mathcal{T}$ in Fig. 5.1.

Algorithm 2 The modified advanced window derivation method for G_P-AA

repeat
  UpCnt $\leftarrow 0$;
  for all $e = (t, f) \in E_P, t, f \ \in V_P$ **do**
    f_CMin_Before$\leftarrow \hat{Y}_{CMin}(f)$;
    t_CMax_Before$\leftarrow \hat{Y}_{CMax}(t)$;
    if $t, f \in V_P^U$ **then**
      $\hat{Y}_{CMin}(f) \leftarrow \max\{\hat{Y}_{CMin}(f), \hat{Y}_{CMin}(t)\}$;
      $\hat{Y}_{CMax}(t) \leftarrow \min\{\hat{Y}_{CMax}(t), \hat{Y}_{CMax}(f)\}$;
    else if $t \in V_P^K$, $f \in V_P^U$ **then**
      $\hat{Y}_{CMin}(f) \leftarrow \max\{\hat{Y}_{CMin}(f), Y(t)\}$;
      $\hat{Y}_{CMax}(t) \leftarrow \min\{\hat{Y}_{CMax}(t), \hat{Y}_{CMax}(f)\}$; /* for training set $\mathcal{T}$. */
    else if $t \in V_P^U$, $f \in V_P^K$ **then**
      $\hat{Y}_{CMin}(f) \leftarrow \max\{\hat{Y}_{CMin}(f), \hat{Y}_{CMin}(t)\}$; /* for training set $\mathcal{T}$. */
      $\hat{Y}_{CMax}(t) \leftarrow \min\{\hat{Y}_{CMax}(t), Y(f)\}$;
    else /* $t, f \in V_P^K$ */
      $\hat{Y}_{CMin}(f) \leftarrow \max\{\hat{Y}_{CMin}(f), Y(t)\}$; /* for training set $\mathcal{T}$. */
      $\hat{Y}_{CMax}(t) \leftarrow \min\{\hat{Y}_{CMax}(t), Y(f)\}$; /* for training set $\mathcal{T}$. */
    end if
    /* Check update counts. */;
    if $\hat{Y}_{CMin}(f) \neq$ f_CMin_Before **then**
      UpCnt $\leftarrow$ UpCnt +1;
    end if
    if $\hat{Y}_{CMax}(t) \neq$ t_CMax_Before **then**
      UpCnt $\leftarrow$ UpCnt +1;
    end if
  end for
until UpCnt = 0; /* When no update happens, loop ends. */

p^K in Fig.5.1	Round 1	Round 2	Round 3	Round 4	$WinType$
$c(1993)$	$(-\infty, 1999)$	$(-\infty, 1999)$	$(-\infty, 1999)$	$(-\infty, 1999)$	Type-3
$d(1999)$	$(1993, +\infty)$	$(1993, 2007)$	$(1993, 2005)$	$(1993, 2005)$	Type-1
$f(2003)$	$(-\infty, +\infty)$	$(-\infty, 2005)$	$(-\infty, 2005)$	$(-\infty, 2005)$	Type-3
$g(2001)$	$(-\infty, +\infty)$	$(-\infty, 2005)$	$(-\infty, 2005)$	$(-\infty, 2005)$	Type-3
$h(2007)$	$(-\infty, +\infty)$	$(1999, +\infty)$	$(1999, +\infty)$	$(1999, +\infty)$	Type-2
$k(2005)$	$(-\infty, +\infty)$	$(2003, +\infty)$	$(2003, +\infty)$	$(2003, +\infty)$	Type-2
$l(2006)$	$(-\infty, +\infty)$	$(2003, +\infty)$	$(2003, +\infty)$	$(2003, +\infty)$	Type-2
	UpCnt = 2	UpCnt = 6	UpCnt =1	UpCnt =0	

Table 5.5: The intermediate and final results of the example training set $\mathcal{T}$ in Fig. 5.1.

p^U in Fig.5.1	a	j
Derived Window	$(-\infty, 1999)$	$(2003, +\infty)$
$WinType/BoundVal$	Type-3/1999	Type-2/2003
$\hat{Y}(p^U)$ by G_P-AS	1999	2003
$\mathcal{T}_{p^U}$	c(1993, Type-3, 1999)	k(2005, Type-2, 2003) l(2006, Type-2, 2003)
$\hat{Y}(p^U)$ by G_P-AA	1993	2006 (2005.5)

Table 5.6: Comparison on the estimation results on papers a and j of the example in Fig. 5.1 by G_P-AS versus G_P-AA.

Recall Table 5.4, we notice that the estimation results of paper a and paper j will be affected by the advanced year value calculation method, according to the derived training set in Table 5.5 and Eq. (5.9). The comparison on the estimation results between G_P-AS and G_P-AA is listed in Table 5.6.

So far, we are only illustrating how the three algorithms work and how different the estimation results appear. In the experiment section (Section 5.2), we will see their performance evaluated on the real datasets.

5.1.3 MYE for paper authorship network G_{AP}

In this section, we move to the paper-author bipartite graph G_{AP}. An artificially created example of MYE problem in G_{AP} is shown in Fig. 5.2. In this example, there are 8 papers $(a-h)$ and 4 authors $(i-l)$, where papers a, b, d, e have year information $(\in V_P^K)$ while c, f, g, h are missing year $(\in V_P^U)$.

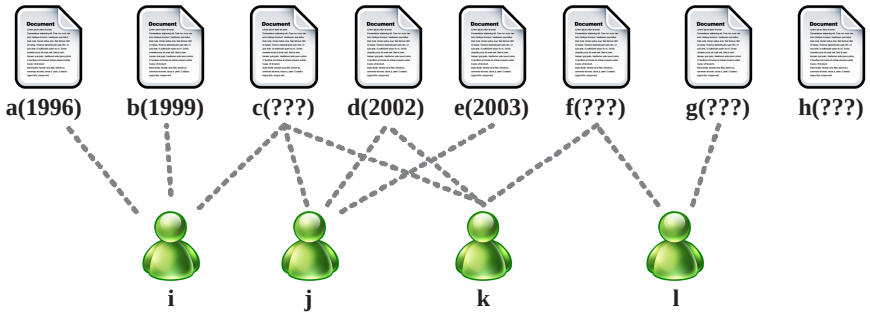


Figure 5.2: An example of a paper authorship network with 8 papers $(a-h)$ and 4 authors $(i-l)$, where papers (a, b, d, e) are $\in V_P^K$ and (c, f, g, h) are $\in V_P^U$.

For G_{AP}, we will also introduce three algorithms, namely G_{AP}-Ba, G_{AP}-Iter and G_{AP}-AdvIter, in an increasing complexity order.

Algorithm for MYE in G_{AP}: G_{AP}-Ba and G_{AP}-Iter

G_{AP}-Ba is the basic algorithm and G_{AP}-Iter is simply repeating G_{AP}-Ba until convergence, thus we introduce them together. The basic algorithm, G_{AP}-Ba, include three steps:

i) Derive author active publishing window.

   For each author, based on the graph topology and paper year information, we can derive an active paper publishing window. Eqs. (5.10) and (5.11) give the definition of the two bounds of this window:

$$AW\_Min(a) = \min_{p \in P(a) \cap V_P^K} Y(p), \tag{5.10}$$

$$AW\_Max(a) = \max_{p \in P(a) \cap V_P^K} Y(p), \tag{5.11}$$

where $P(a), \forall\ a\ \in V_A$ is the paper set written by author a. It is possible that $P(a) \cap V_P^K = \emptyset$, and we consider it as a non-existent bound. According to the above definition, the two bounds are either co-existent or non-existent.

ii) Derive paper possible year estimation window.

Based on the derived author active window, we can further define the paper possible year window:

$$\hat{Y}_{AMin}(p^U) = \min\{\max_{a \in A(p^U)} AW\_min(a), \min_{a \in A(p^U)} AW\_max(a)\}, \tag{5.12}$$

$$\hat{Y}_{AMax}(p^U) = \max\{\max_{a \in A(p^U)} AW\_min(a), \min_{a \in A(p^U)} AW\_max(a)\}, \tag{5.13}$$

where $A(p^U), \forall p^U \in V_P^U$ is the author set of paper p^U.

In most cases, $\hat{Y}_{AMin}(p^U) = \max_{a \in A(p^U)} AW\_min(a)$ and $\hat{Y}_{AMax}(p^U) = \min_{a \in A(p^U)} AW\_max(a)$. However, in case of the condition that authors' active windows have no intersection (this is possible because the author active window dose not take the missing year papers into account), we rewrite them to be Eqs. (5.12)-(5.13). For example, we look at paper c in Fig. 5.2. The author set of paper c is $A(c) = \{i(1996, 1999), j(2002, 2003), k(2002, 2002)\}$, where the author active windows are inside parentheses. According to the definition, we thus have:

1) $\max_{a \in A(c)} AW\_Min(a) = \max\{1996, 2002, 2002\} = 2002$,

2) $\min_{a \in A(c)} AW\_Max(a) = \min\{1999, 2003, 2002\} = 1999$.

Therefore, according to Eqs. (5.12)-(5.13), we obtain the possible year estimation window of paper c : $[1999, 2002]$.

iii) Calculate year value.

In this algorithm, we apply the simple year value calculation method, the same one as in the G_P-SS algorithm. There is only a small difference that in G_P-SS, there are four types of the year estimation window, whereas in G_{AP}, there are only two possible types, both bounds exist (Type-1) or neither exists (Type-4). Therefore, the estimated year value is either $\frac{\hat{Y}_{AMin}(p^U)+\hat{Y}_{AMax}(p^U)}{2}$ or labeled as *Uncovered*.

Note the rationale of the design of the basic algorithm is based on an observation that most authors are continuously active in publishing papers. Hence, the publishing years of his/her papers are usually within a continuous window. If we obtain the windows of all the coauthors of a missing year paper, the intersection of these windows will be an interval that with high confidence the real publishing year falls in.

The pseudo code for G_{AP}-Iter (including G_{AP}-Ba) is shown in Algorithm 3. Lines 2-19 are the steps of G_{AP}-Ba and G_{AP}-Iter is simply repeating G_{AP}-Ba (Line 1). The estimation results in the previous rounds affect the subsequent rounds, because each author's active publishing window will be re-calculated according to all the paper year information (given or estimated in the last round, Lines 7-8). Lines 13-17 are the implementation of Eqs. (5.12) and (5.13). The intermediate and final estimation results by running G_{AP}-Iter on the example of Fig. 5.2 are listed in Table 5.7.

In Table 5.7, the G_{AP}-Iter repeats 3 rounds until convergence. We show the intermediate results of the author active windows for authors (nodes i, j, k, l), the possible paper publishing windows for missing year papers (nodes c, f, g, h), and their estima-

Algorithm 3 The pseudo code of G_{AP}-Iter

repeat
  for all $e = (a, p) \in E_{AP}, a \in V_A, p \in V_P$ **do**
    if $p \in V_P^K$ **then**
      $AW\_Min(a) \leftarrow \min\{Y(p), AW\_Min(a)\}$
      $AW\_Max(a) \leftarrow \max\{Y(p), AW\_Max(a)\}$
    else if $\hat{Y}(p)$ exists **then** /* $p \in V_P^U$ */
      $AW\_Min(a) \leftarrow \min\{\hat{Y}(p), AW\_Min(a)\}$
      $AW\_Max(a) \leftarrow \max\{\hat{Y}(p), AW\_Max(a)\}$
    end if
  end for
  for all $p^U \in V_P^U$ **do**
    for all $a \in A(p^U)$ **do**
      $maxMin \leftarrow \max\{AW\_Min(a), maxMin\}$
      $minMax \leftarrow \min\{AW\_Max(a), minMax\}$
    end for
    $\hat{Y}_{AMin}(p^U) \leftarrow \min\{maxMin, minMax\}$
    $\hat{Y}_{AMax}(p^U) \leftarrow \max\{maxMin, minMax\}$
    $\hat{Y}(p^U) \leftarrow \frac{\hat{Y}_{AMin}(p^U)+\hat{Y}_{AMax}(p^U)}{2}$
  end for
until No update happens

Node	Type	Round 1	Round 2	Round 3
i	Author	$(1996, 1999)$	$(1996, 2001)$	$(1996, 2001)$
j	Author	$(2002, 2003)$	$(2001, 2003)$	$(2001, 2003)$
k	Author	$(2002, 2002)$	$(2001, 2002)$	$(2001, 2002)$
l	Author	$(-\infty, +\infty)$	$(2002, 2002)$	$(2002, 2002)$
c	Paper	$(1999, 2002)$	$(2001, 2001)$	$(2001, 2001)$
$\hat{Y}(c)$		2001 (2000.5)	2001	2001
f	Paper	$(2002, 2002)$	$(2002, 2002)$	$(2002, 2002)$
$\hat{Y}(f)$		2002	2002	2002
g	Paper	$(-\infty, +\infty)$	$(2002, 2002)$	$(2002, 2002)$
$\hat{Y}(g)$		*Uncovered*	2002	2002
h	Paper	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$\hat{Y}(h)$		*Uncovered*	*Uncovered*	*Uncovered*

Table 5.7: The intermediate and final estimation results obtained by running G_{AP}-Ba and G_{AP}-Iter on the example shown in Fig. 5.2

tion results ($\hat{Y}(p^U), p^U \in \{c, f, g, h\}$) in each round. The column labeled as "Round 1" shows the results generated by algorithm G_{AP}-Ba. Comparing to G_{AP}-Ba, G_{AP}-Iter helps to share information through the co-author relationships, like author l in Table 5.7. Therefore, G_{AP}-Iter obtains a lower uncovered ratio (1/4) than G_{AP}-Ba (2/4).

We need to note that G_{AP}-Iter may add inaccuracy during the information propagation, i.e., the estimation results in the previous rounds affect the derivation of both the author active windows and estimation results in the subsequent rounds. For example, $\hat{Y}(c)$ after Round 1 is 2001. In Round 2, the active windows of all the coauthors of paper c, $P(c) = \{i, j, k\}$ get updated, and hence the related paper year estimation windows get updated too. Although G_{AP}-Iter helps to decrease the uncovered ratio, it may not improve estimation accuracy like MAE (under certain situation, can be even worse than G_{AP}-Ba).

In order to compensate the weakness of G_{AP}-Iter so that both uncovered ratio and estimation accuracy can be improved, we propose the G_{AP}-AdvIter, which has an advanced iteration pro-

cedure to reduce the propagation of inaccurate information.

Algorithm for MYE in G_{AP}: G_{AP}-AdvIter

According to the previous discussion, the key point of improving the estimation accuracy in G_{AP} is to propagate as much "good" information as possible. Hence, we propose a heuristic algorithm, G_{AP}-AdvIter to achieve this. Here are some definitions:

1. Consistent-Coauthor-Count between two papers: the number of common coauthors of the two papers. We denote it by function $w(\cdot)$. Given any two papers, we can calculate their Consistent-Coauthor-Count by the following expression:

$$\forall\, p,\ q\ \in\ V_P,\ w(p,q) = w(q,p) = |A(p) \cap A(q)|, \quad (5.14)$$

   where $w(\cdot)$ is a non-negative integer and equals to zero only when the two papers have no common coauthors.

2. w-Consistent-Coauthor-Pair relationship: if any two papers, $\forall p, q \in V_P$, satisfy: $w(p,q) = w(q,p) > 1$, then we call them w-Consistent-Coauthor-Pair.

3. Consistent-Coauthor-Pair set of a paper $p \in V_P$, denoted by $\Omega(p)$:

$$\Omega(p) = \{q | q \in V_P \text{ and } w(p,q) > 1\} \quad (5.15)$$

We give some illustrations of these definitions using the example in Fig. 5.2: $w(a,g) = |\emptyset| = 0$ and $w(c,d) = |\{j,k\}| = 2$, thus, paper c, d have the 2-Consistent-Coauthor-Pair relationship. Except this, there is no more Consistent-Coauthor-Pairs in Fig. 5.2. Therefore, we obtain $\Omega(c) = \{d\}$, $\Omega(d) = \{c\}$ and $\Omega(p) = \emptyset, \forall p \in \{a,b,e,f,g,h\}$.

It is a reasonable assumption that if more authors work together and publish papers, it is more probable that these papers

are published within a small time window. For example, students worked together with their supervisors/group members and published certain papers during their Master/PhD study. Note this is only a sufficient condition, the reverse may not be true.

The above assumption implies that if two papers have w-Consistent-Coauthor-Pair relationship, then with high probability that their publishing years are close. In addition, this probability is positively correlated to the value of w. We conjecture that the estimated year values by utilizing the w-Consistent-Coauthor-Pair relationship must be "better" information for propagation.

The pseudo code of G_{AP}-AdvIter is listed in Algorithm 4, which shows how we make use of the more reliable information for propagation.

Comparing to Algorithm 3, we notice that Algorithm 4 only added Lines 1-3 and Lines 15-16. Lines 1-3 are the process to find $\Omega(p^U)$ for each missing year paper and this is done during initialization. Lines 15-16 show that we give higher priority to estimating year values if the w-Consistent-Coauthor-Pair relationship can help, than the basic procedure (Lines 17-25). The expression of the function W is in Eq. (5.16):

$$W(p^U, \gamma) = \frac{\sum_{q \in \Omega(p^U) \cap V_P^K} w(p^U, q)^\gamma \times Y(q)}{\sum_{q \in \Omega(p^U) \cap V_P^K} w(p^U, q)^\gamma}, \quad \text{if } \Omega(p^U) \cap V_P^K \neq \emptyset \tag{5.16}$$

The meaning of Eq. (5.16) is to take a γ-weighted average on the given year information of those papers in the set $\Omega(p^U) \cap V_P^K$. For example, if $\Omega(p^U) \cap V_P^K = \{q, r\}, w(p^U, q) = 2, w(p^U, r) = 3, Y(q) = 2000, Y(r) = 2002$, then $W(p^U, \gamma) = \frac{2^\gamma \times 2000 + 3^\gamma \times 2002}{2^\gamma + 3^\gamma}$. Here parameter γ is used to tune the importance we put on the values of w, e.g., if we set $\gamma = 0$, it implies that no weight is considered and the result is simply the average; and when $\gamma = 1$, it is a normal weighted average calculation; while $\gamma \to \infty$,

Algorithm 4 The pseudo code of G_{AP}-AdvIter

for all $p^U \in V_P^U$ **do**
  Derive the Consistent-Coauthor-Pair set, $\Omega(p^U)$.
end for
repeat
  for all $e = (a, p) \in E_{AP}, a \in V_A, p \in V_P$ **do**
    if $p \in V_P^K$ **then**
      $AW\_Min(a) \leftarrow \min\{Y(p), AW\_Min(a)\}$
      $AW\_Max(a) \leftarrow \max\{Y(p), AW\_Max(a)\}$
    else if $\hat{Y}(p)$ exists **then** /* $p \in V_P^U$ */
      $AW\_Min(a) \leftarrow \min\{\hat{Y}(p), AW\_Min(a)\}$
      $AW\_Max(a) \leftarrow \max\{\hat{Y}(p), AW\_Max(a)\}$
    end if
  end for
  for all $p^U \in V_P^U$ **do**
    if $\Omega(p^U) \cap V_P^K \neq \emptyset$ **then** /* for AdvIter */
      $\hat{Y}(p^U) \leftarrow W(p^U, \gamma)$
    else
      for all $a \in A(p^U)$ **do**
        $maxMin \leftarrow \max\{AW\_Min(a), maxMin\}$
        $minMax \leftarrow \min\{AW\_Max(a), minMax\}$
      end for
      $\hat{Y}_{AMin}(p^U) \leftarrow \min\{maxMin, minMax\}$
      $\hat{Y}_{AMax}(p^U) \leftarrow \max\{maxMin, minMax\}$
      $\hat{Y}(p^U) \leftarrow \frac{\hat{Y}_{AMin}(p^U) + \hat{Y}_{AMax}(p^U)}{2}$
    end if
  end for
until No update happens

it leads to the special case where only the papers in the set $\Omega(p^U) \cap V_P^K$ with the largest w are involved in the calculation. In addition, since it is meaningless for function W if $\Omega(p^U) \cap V_P^K = \emptyset$, we need to have a check beforehand (Line 15).

Node	Type	Round 1	Round 2	Round 3
i	Author	$(1996, 1999)$	$(1996, 2002)$	$(1996, 2002)$
j	Author	$(2002, 2003)$	$(2002, 2003)$	$(2002, 2003)$
k	Author	$(2002, 2002)$	$(2002, 2002)$	$(2002, 2002)$
l	Author	$(-\infty, +\infty)$	$(2002, 2002)$	$(2002, 2002)$
c	Paper	$(2002, 1999)$	$(2001, 2001)$	$(2001, 2001)$
$\hat{Y}(c)$	$W(c, 0)$	2002	2002	2002
f	Paper	$(2002, 2002)$	$(2002, 2002)$	$(2002, 2002)$
$\hat{Y}(f)$		2002	2002	2002
g	Paper	$(-\infty, +\infty)$	$(2002, 2002)$	$(2002, 2002)$
$\hat{Y}(g)$		*Uncovered*	2002	2002
h	Paper	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
$\hat{Y}(h)$		*Uncovered*	*Uncovered*	*Uncovered*

Table 5.8: The intermediate and final estimation results obtained by running G_{AP}-AdvIter on the example shown in Fig. 5.2

In Table 5.8, we list the intermediate and final estimation results obtained by running G_{AP}-AdvIter on the example shown in Fig. 5.2. As analyzed previously, $\Omega(c) = \{d\}$, $\Omega(d) = \{c\}$ and $\Omega(p) = \emptyset, \forall p \in \{a, b, e, f, g, h\}$, hence only $\hat{Y}(c) = Y(d) = 2002$ is affected by G_{AP}-AdvIter and also the related author active windows: $i : (1996, 2002)$, $j : (2002, 2003)$ and $k : (2002, 2002)$.

5.1.4 MYE for heterogenous network G

For a heterogeneous network, $G = (G_P \cup G_{AP})$, which consists of both G_P and G_{AP}, we make use of the proposed methods and results discussed in the previous two sections. Since for both G_P and G_{AP}, we proposed three algorithms of different complexity, there can be totally 9 different combinations. With careful

consideration, we pick out 3 typical combinations as MYE algorithms for G:

1) G-SSBa: combination of G_P-SS and G_{AP}-Ba

2) G-ASIter: combination of G_P-AS and G_{AP}-Iter

3) G-AdvIter: combination G_P-AA and G_{AP}-AdvIter

In fact, selecting the "combination" is not trivial, and let us explain it properly next. The common part of the two algorithms consists of these two steps: (a) derivation of possible year estimation window and (b) calculate the estimated year value based on the derived window.

No matter which combined algorithm for G is applied, for each missing year paper, two possible year estimation windows will be derived, one by the G_P part $[\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]$, and the other by the G_{AP} part $[\hat{Y}_{AMin}(p^U), \hat{Y}_{AMax}(p^U)]$, due to the independency of these two procedures.

Considering the four types of the derived estimation window from G_P and two types from G_{AP}, each missing year paper can end with the following four cases of which case (d) is most likely:

(a) $(\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)) = (\hat{Y}_{AMin}(p^U), \hat{Y}_{AMax}(p^U)) = (-\infty, +\infty)$, then it can only lead to the *Uncovered* estimation result;

(b) $(\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)) = (-\infty, +\infty)$ but $[\hat{Y}_{AMin}(p^U), \hat{Y}_{AMax}(p^U)]$ is not, then it is as if only the G_{AP} part algorithm is in action;

(c) $(\hat{Y}_{AMin}(p^U), \hat{Y}_{AMax}(p^U)) = (-\infty, +\infty)$ but $[\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]$ is not, then it is as if only the G_P part algorithm is in action;

(d) Neither window is $(-\infty, +\infty)$, we will have a detailed discussion for the three algorithms: G-SSBa, G-ASIter and G-AdvIter respectively.

For G-SSBa, G-ASIter and G-AdvIter, the way we do the combination follows a general criterion that we always give higher priority to the window derived from G_P than from G_{AP}. This is because the former is more reliable than the latter, as the latter may involve inaccuracy in information propagation.

Algorithm for MYE in G: G-SSBa and G-ASIter

Since the structures of G-SSBa and G-ASIter are similar, we try to merge their pseudo codes together for space saving and ease of description[2]. The pseudo code of G-SSBa and G-ASIter for case (d) is listed in Algorithm 5.

In Algorithm 5, we denote $\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U)$ to be the two bounds of the derived year estimation window in G. In the beginning, we derive $[\hat{Y}_{CMin}, \hat{Y}_{CMax}]$ by simple window derivation method for algorithm G-SSBa, or advanced window derivation method for algorithm G-ASIter (Lines 1-5).

Next, we derive $[\hat{Y}_{GMin}, \hat{Y}_{GMax}]$ depending on the type of the window in G_P, e.g., Lines 11-19 for Type-1, Lines 20-28 for Type-2 and Lines 29-37 for Type-3. The derivation follows the general criterion that if the intersection of $[\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]$ and $[\hat{Y}_{AMin}(p^U), \hat{Y}_{AMax}(p^U)]$ is not empty, we take this intersection window as $[\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U)]$; otherwise, we take $[\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]$. Line 41 is the same simple year value calculation method as in G_P-SS, G_P-AS, G_{AP}-Ba and G_{AP}-Iter. In fact, if conditions (Line 22 or Line 31) happen (i.e., the two windows do not intersect with each other), the operation (Lines 23-24 and Lines 32-33, together Line 41) is equivalent to Eq. (5.5)-Eq. (5.6), taking the bound values. For G-SSBa of which the combination includes G_{AP}-Ba, the basic procedure will only go through once (Lines 43-45); While for G-ASIter of which the combination includes G_{AP}-Iter, the $[\hat{Y}_{GMin}, \hat{Y}_{GMax}]$

[2]In real implementation, they are separated.

Algorithm 5 The pseudo code of G-SSBa and G-ASIter for case (d)

if G-SSBa **then**
  $[\hat{Y}_{CMin}, \hat{Y}_{CMax}] \leftarrow$ **Simple Window Derivation** in Eq.(5.2) and (5.3);
else if G-ASIter **then**
  $[\hat{Y}_{CMin}, \hat{Y}_{CMax}] \leftarrow$ **Advanced Window Derivation** in Algorithm 1;
end if
repeat
  $[\hat{Y}_{AMin}, \hat{Y}_{AMax}] \leftarrow$ by G_{AP}-Ba, Eqs. (5.10), (5.11), (5.12), (5.13);
  for all $p^U \in V_P^U$ **do**
    $\hat{Y}_{GMin}(p^U) \leftarrow -\infty$; /* Init */
    $\hat{Y}_{GMax}(p^U) \leftarrow +\infty$; /* Init */
    if $\hat{Y}_{CMin}(p^U) > -\infty$ And $\hat{Y}_{CMax}(p^U) < +\infty$ **then**
      /* Type-1 Window in G_P */
      if $\hat{Y}_{AMin}(p^U) < \hat{Y}_{CMin}(p^U)$ Or $\hat{Y}_{AMax}(p^U) > \hat{Y}_{CMax}(p^U)$ **then**
        $\hat{Y}_{GMin}(p^U) \leftarrow \hat{Y}_{CMin}(p^U)$;
        $\hat{Y}_{GMax}(p^U) \leftarrow \hat{Y}_{CMax}(p^U)$;
      else
        $\hat{Y}_{GMin}(p^U) \leftarrow \max\{\hat{Y}_{CMin}(p^U), \hat{Y}_{AMin}(p^U)\}$;
        $\hat{Y}_{GMax}(p^U) \leftarrow \min\{\hat{Y}_{CMax}(p^U), \hat{Y}_{AMax}(p^U)\}$;
      end if
    else if $\hat{Y}_{CMin}(p^U) > -\infty$ And $\hat{Y}_{CMax}(p^U) = +\infty$ **then**
      /* Type-2 Window in G_P */
      if $\hat{Y}_{AMax}(p^U) < \hat{Y}_{CMin}(p^U)$ **then**
        $\hat{Y}_{GMin}(p^U) \leftarrow \hat{Y}_{CMin}(p^U)$;
        $\hat{Y}_{GMax}(p^U) \leftarrow \hat{Y}_{CMin}(p^U)$;
      else
        $\hat{Y}_{GMin}(p^U) \leftarrow \max\{\hat{Y}_{CMin}(p^U), \hat{Y}_{AMin}(p^U)\}$;
        $\hat{Y}_{GMax}(p^U) \leftarrow \hat{Y}_{AMax}(p^U)$;
      end if
    else if $\hat{Y}_{CMin}(p^U) = -\infty$ And $\hat{Y}_{CMax}(p^U) < +\infty$ **then**
      /* Type-3 Window in G_P */
      if $\hat{Y}_{AMin}(p^U) > \hat{Y}_{CMax}(p^U)$ **then**
        $\hat{Y}_{GMin}(p^U) \leftarrow \hat{Y}_{CMax}(p^U)$;
        $\hat{Y}_{GMax}(p^U) \leftarrow \hat{Y}_{CMax}(p^U)$;
      else
        $\hat{Y}_{GMin}(p^U) \leftarrow \hat{Y}_{AMin}(p^U)$;
        $\hat{Y}_{GMax}(p^U) \leftarrow \min\{\hat{Y}_{CMax}(p^U), \hat{Y}_{AMax}(p^U)\}$;
      end if
    else
      Case (b);/* Type-4 Window in G_P */
    end if
    $\hat{Y}(p^U) \leftarrow \frac{\hat{Y}_{GMin}(p^U)+\hat{Y}_{GMax}(p^U)}{2}$; /* Simple Year Value Calculation */
  end for
  if G-SSBa **then**
    Break;
  end if
until No update happens

window will be propagated until convergence (Line 6 together with Line 46).

Algorithm for MYE in G: G-AdvIter

G-AdvIter is the combination of G_P-AA and G_{AP}-AdvIter, therefore, the concepts of training set $\mathcal{T}$ as well as the Consistent-Coauthor-Pair relationship will be involved. Algorithm 6 list the pseudo code of G-AdvIter for case (d):

In Algorithm 6, we omit the same code of deriving $\hat{Y}_{GMin}(p^U)$, $\hat{Y}_{GMax}(p^U)$ as in Algorithm 5 (Lines 8, 16, 30). At beginning (Line 1), we call the function G_P-AA (Algorithm 2) to derive $\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)$ and the training set $\mathcal{T}$, which is a series of 3-tuples $\{y_t, WinType_t, BoundVal_t\}$ from the papers with known year information. The preparation of the Consistent-Coauthor-Pair set for each missing year paper $\Omega(p^U)$, like G_{AP}-AdvIter, is also called (Line 2). The main difference between G-AdvIter and G-ASIter is the method of calculating year value. For all three types of window in G_P, we apply the $W_G(p^U, \gamma, y_l, y_r)$ function to calculate the year value:

$$\begin{aligned} \text{When} \quad \Omega_G(p^U) &= \{q | q \in \Omega(p^U), Y(q) \in (y_l, y_r)\}, \\ &\text{and} \quad \Omega_G(p^U) \cap V_P^K \neq \emptyset, \\ W_G(p^U, \gamma, y_l, y_r) &= \frac{\sum_{q \in \Omega_G(p^U) \cap V_P^K} w(p^U, q)^\gamma \times Y(q)}{\sum_{q \in \Omega_G(p^U) \cap V_P^K} w(p^U, q)^\gamma}; \\ \text{Otherwise} &= Null. \end{aligned} \tag{5.17}$$

In Eq. (5.17), the different part of W_G is that we pick out a subset of papers from $\Omega(p^U)$, denoted by $\Omega_G(p^U)$, satisfying the condition that the paper publishing years are within an input window $[y_l, y_r]$, i.e., $\Omega_G(p^U) = \{q | q \in \Omega(p^U), Y(q) \in [y_l, y_r]\}$. For Type-1 window of G_P, we choose the subset $\Omega_G(p^U)$ by setting the input window to be $[y_l = \hat{Y}_{CMin}(p^U), y_r = \hat{Y}_{CMax}(p^U)]$ for calculating $\hat{Y}(p^U)$ (Line 10). But if $\Omega_G(p^U) \cap V_P^K = \emptyset$, we change back to the default way (Lines 12-14).

Algorithm 6 The pseudo code of G-AdvIter for case (d)

Run Algorithm 2, derive $[\hat{Y}_{CMin}, \hat{Y}_{CMax}]$ and the training set $\mathcal{T}$.
Derive the Consistent-Coauthor-Pair set: $\Omega(p^U), \forall p^U \in V_P^U$.
repeat
  $[\hat{Y}_{AMin}, \hat{Y}_{AMax}] \leftarrow$ by G_{AP}-Ba, Eqs. (5.10), (5.11), (5.12), (5.13);
  for all $p^U \in V_P^U$ **do**
    $\hat{Y}(p^U) \leftarrow Null$; /* Init */
    if $\hat{Y}_{CMin}(p^U) > -\infty$ And $\hat{Y}_{CMax}(p^U) < +\infty$ **then** /* Type-1 Win */
      Derivation of $\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U)$; /* ∼Alg. 5, Lines 11-19 */
      if $\Omega(p^U) \cap V_P^K \neq \emptyset$ **then**
        $\hat{Y}(p^U) \leftarrow W_G(p^U, \gamma, \hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U))$
      end if
      if $\hat{Y}(p^U) = Null$ **then** /* In case W_G does not work */
        $\hat{Y}(p^U) \leftarrow \frac{\hat{Y}_{GMin}(p^U) + \hat{Y}_{GMax}(p^U)}{2}$;
      end if
    else if $\hat{Y}_{CMin}(p^U) > -\infty$ And $\hat{Y}_{CMax}(p^U) = +\infty$ **then** /* Type-2 */
      Derivation of $\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U)$; /* ∼Alg. 5, Lines 20-28 */
      $dResult \leftarrow d(\text{Type-2}, \hat{Y}_{CMin}(p^U))$;/*$d(WinType(p^U), BoundVal(p^U)$*/
      $\delta \leftarrow dResult - \hat{Y}_{CMin}(p^U)$;
      if $\Omega(p^U) \cap V_P^K \neq \emptyset$ **then**
        $\hat{Y}(p^U) \leftarrow W_G(p^U, \gamma, \hat{Y}_{CMin}(p^U), \hat{Y}_{CMin}(p^U) + 2\delta)$
      end if
      if $\hat{Y}(p^U) = Null$ **then** /* In case W_G does not work */
        if $\hat{Y}_{AMax}(p^U) < \hat{Y}_{CMin}(p^U)$ Or $dResult \in (\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U))$ **then**
          $\hat{Y}(p^U) \leftarrow dResult$
        else
          $\hat{Y}(p^U) \leftarrow \frac{\hat{Y}_{GMin}(p^U) + \hat{Y}_{GMax}(p^U)}{2}$;
        end if
      end if
    else if $\hat{Y}_{CMin}(p^U) = -\infty$ And $\hat{Y}_{CMax}(p^U) < +\infty$ **then** /* Type-3 */
      Derivation of $\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U)$; /* ∼Alg. 5, Lines 29-37 */
      $dResult \leftarrow d(\text{Type-3}, \hat{Y}_{CMax}(p^U))$;/*$d(WinType(p^U), BoundVal(p^U)$*/
      $\delta \leftarrow \hat{Y}_{CMax}(p^U) - dResult$;
      if $\Omega(p^U) \cap V_P^K \neq \emptyset$ **then**
        $\hat{Y}(p^U) \leftarrow W_G(p^U, \gamma, \hat{Y}_{CMax}(p^U) - 2\delta, \hat{Y}_{CMax}(p^U))$
      end if
      if $\hat{Y}(p^U) = Null$ **then** /* In case W_G does not work */
        if $\hat{Y}_{AMin}(p^U) > \hat{Y}_{CMax}(p^U)$ Or $dResult \in (\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U))$ **then**
          $\hat{Y}(p^U) \leftarrow dResult$
        else
          $\hat{Y}(p^U) \leftarrow \frac{\hat{Y}_{GMin}(p^U) + \hat{Y}_{GMax}(p^U)}{2}$;
        end if
      end if
    end if
  end for
until No update happens

The process for Type-2 or Type-3 window is a little more complicated. For Type-2 window, both $\Omega(p^U)$ and $\mathcal{T}$ are available tools. The following way is proposed: we first derive the estimation year value, denoted by *dResult*, through $d(\cdot)$ function expressed in Eq. (5.9). We use this *dResult* and the input parameter $\hat{Y}_{CMin}(p^U)$ to define a window $[y_l = \hat{Y}_{CMin}(p^U), y_r = \hat{Y}_{CMin}(p^U)+2\delta]$, of which the interval equals to twice of the distance from *dResult* to $\hat{Y}_{CMin}(p^U)$, $\delta = dResult - \hat{Y}_{CMin}(p^U)$. This window is then used to derive $\Omega_G(p^U)$ and calculate $\hat{Y}(p^U)$ (Lines 17-21). If $\Omega_G(p^U) \cap V_P^K = \emptyset$, we have a second choice which is *dResult* (Lines 22-25), if one of the following two conditions is met:

(a) $[\hat{Y}_{CMin}(p^U), \hat{Y}_{CMax}(p^U)]$ and $[\hat{Y}_{AMin}(p^U), \hat{Y}_{AMax}(p^U)]$ have no intersection part; or

(b) $dResult \in [\hat{Y}_{GMin}(p^U), \hat{Y}_{GMax}(p^U)]$.

Otherwise, we change back to the default way (Line 26).

The process for Type-3 window is symmetric to Type-2. The only difference is that the input window for deriving $\Omega_G(p^U)$ and $\hat{Y}_{CMin}(p^U)$ becomes $W_G(p^U, \gamma, y_l = \hat{Y}_{CMax}(p^U) - 2\delta, y_r = \hat{Y}_{CMax}(p^U))$ (Lines 31-35).

5.2 Experiment and Evaluation

In this section, we present the experiment settings and evaluation results. In the experiment, we test the proposed MYE algorithms in the last section by applying them to all the three types of the academic social networks, the paper citation network G_P, the paper authorship network G_{AP} and the heterogenous network G.

5.2.1 Data Sets

We have tried three different datasets: Microsoft Libra [4, 67], DBLP [6, 54] with additional citation information, DBLP-Cit [78, 79], and American Physical Society (APS) dataset [7]. The raw data sets are not perfect in that: (a) there exits a proportion of missing year papers; (b) Some citation links are pointing from early published papers (smaller year) to later ones (lager year), which breaks Assumption 5.1.

Since the performance evaluation needs ground truth knowledge, we have to do some preprocessing on the original data sets, including: a) remove these missing year papers and their relationships (citation links and paper-authorship links); b) remove those citation links breaking Assumption 5.1.

Table 5.9 lists the general information about the three data sets after preprocessing:

Data set	Microsoft Libra	DBLP-Cit	APS
Input Window	(1900 - 2013)	(1900 - 2013)	(1900 - 2013)
#papers	2323235	1558503	463347
#authors	1278407	914053	320964
#total citation links	10003121	2062896	4689142

Table 5.9: General information of the three data sets used after preprocessing.

As we can see in Table 5.9, the average number of citation links per paper of the three data sets are: 4.31 for Libra, 1.33 for DBLP-Cit and 10.34 for APS, which appears disparate. This probably reflects how well these three data sets are collected and managed. The APS data set is the most complete in terms of the paper citation information, and the DBLP-Cit is probably the least[3]. For DBLP-Cit, the job to find citation links for an existing paper set is a big challenge. The small number of average

[3]DBLP [6, 54] is a popular and well-managed data set, with complete and accurate meta information. But it does not provide paper citation information. DBLP-Cit [12] is created based on the original DBLP paper set with adding paper citation relationships through proper mining method [78, 79]

paper citation links shows that likely only a small proportion of the complete paper citation links are found.

The completeness and accuracy of the citation links will only affect those MYE algorithms that rely on citation information, e.g., the three algorithms for G_P.

5.2.2 Evaluation methodology

We apply a similar approach like K-fold cross validation [51, 61] to evaluate the MYE algorithms. For each date set after pre-processing, we randomly split the paper set into K mutually exclusive groups, i.e., $V_P = \cup_{k=1}^{K} V_{P_k}$, and $\forall i \neq j, V_{P_i} \cap V_{P_j} = \emptyset$. In addition, each group has approximately the same size, $|V_{P_k}| \approx \frac{|V_P|}{K}, k = 1, 2, \ldots, K$.

For a given parameter K, the experiment repeats K times. In the jth time, the year information of the papers in group V_{P_j} is artificially hidden, thus assumed to be the missing year paper set $V_P^U = V_{P_j}$, and the remaining groups become the paper set with known year information, i.e., $V_P^K = V_P \setminus V_{P_j}$. The overall performance metrics take the average of the results obtained in each of the K times.

Indirectly, the value of K controls the severity of the missing year phenomenon. For convenience, we define $\eta = \frac{|V_P^U|}{|V_P|} \approx \frac{1}{K}$ to be the *Missing Year Ratio* of the data set. Throughout the experiment, we have tried 5 different $\eta = \frac{1}{8}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}$.

5.2.3 Performance metrics

Three metrics are used to evaluate the performance of the MYE algorithms.

1) Coverage

   We have defined the uncovered ratio in Section 5.1. It equals to the number of those missing year papers finally

labeled as *Uncovered* by MYE algorithms, divided by the total number of missing year papers $|V_P^U|$. We use $N^U = |V_P^U| - \text{Total\#Uncovered}$ to denote the number of the covered part. In one experiment, the coverage metric is equal to $\frac{N^U}{|V_P^U|}$. With K-fold cross validation, the overall coverage becomes:

$$Coverage = \frac{1}{K}\sum_{k=1}^{K}\frac{N_k^U}{|V_{P_k}|}, \tag{5.18}$$

where the subscript k indicates the kth iteration and $V_P^U = V_{P_k}$.

2) Mean absolute error (MAE)

$$MAE = \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{1}{N_k^U}\sum_{i=1}^{N_k^U}|Y(p_i^U) - \hat{Y}(p_i^U)|\Big), \tag{5.19}$$

where in the kth iteration, $V_P^U = V_{P_k}$, $\hat{Y}(p_i^U)$ is the estimated year. $Y(p_i^U)$ is the real year of p_i^U, which we assumed to be unknown when running the MYE algorithms and used only for validation purposes.

3) Root mean square error (RMSE)

$$RMSE = \frac{1}{K}\sum_{k=1}^{K}\Big(\sqrt{\frac{1}{N_k^U}\sum_{i=1}^{N_k^U}[Y(p_i^U) - \hat{Y}(p_i^U)]^2}\ \Big). \tag{5.20}$$

In order to have a better understanding of the coverage metric, we propose an analytical model to calculate the expected coverage for an undirected graph $G = (V, E)$. According to the basic graph theory [36], G can be partitioned into S connected components $G = \cup_{i=1}^{S} G_i$, where $\forall i, j, G_i \cap G_j = \emptyset$.

The iteration mechanism of the MYE algorithms (e.g., G_{AP}-Iter, or G_{AP}-AdvIter) ensures that there can be only two possible outcomes for any connected component $G_i = (V_i, E_i)$ when propagation stops[4]:

[4]The outcome of G_P-AS and G_P-AA is a little complicated, we will discuss it later.

(I) All the missing year papers in this component have feasible estimated values (hence, $\neq$ *Uncovered*), if and only if there exits at least one paper with known year information in this component, i.e., $V_i \cap V_P^K \neq \emptyset$;

(II) Otherwise, all the missing year papers in this component are labeled as *Uncovered*.

If we assume the missing year paper is uniformly distributed among the whole paper set, then the expected coverage value can be calculated by Eq. (5.21):

$$Coverage(\eta, \bigcup_{i=1}^{S} V_i) = 1 - \frac{\sum_{i=1}^{S} \eta^{|V_i|} \cdot |V_i|}{\eta |V|} . \tag{5.21}$$

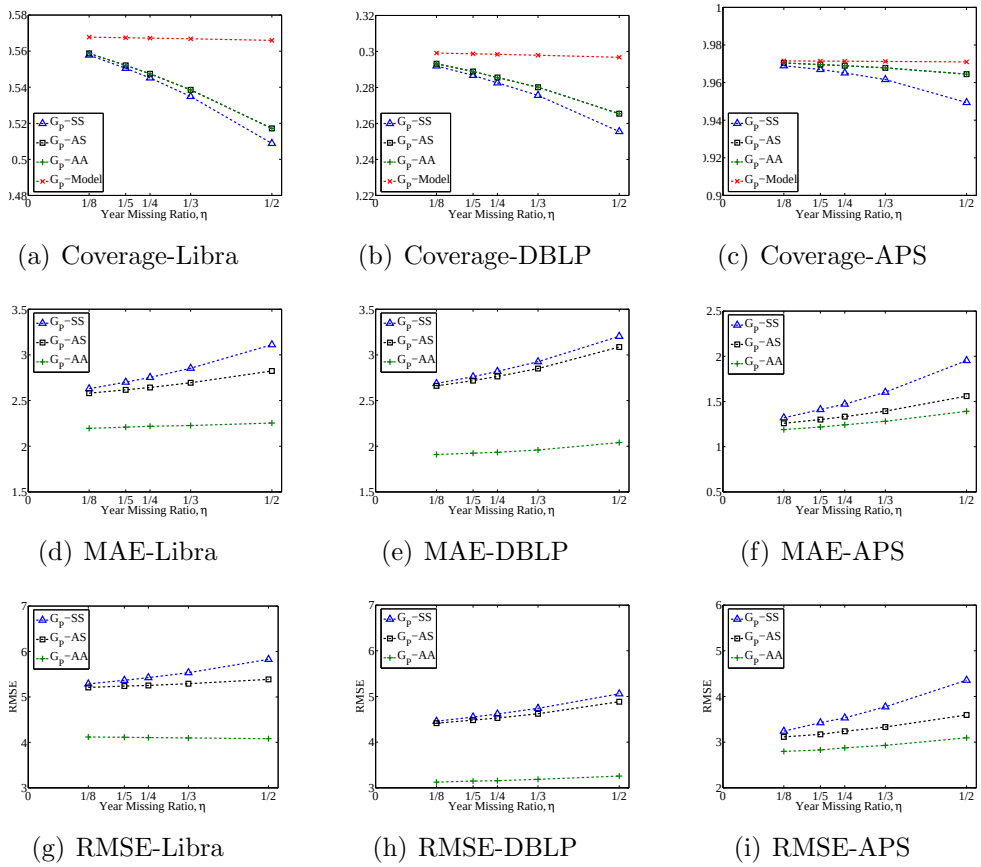
In Eq. (5.21), there are two inputs for this calculation: the year missing ratio η and the vertex partition $V = \cup_{i=1}^{S} V_i$. According to the uniform distribution assumption, each paper is selected to be the missing year paper with equal probability η. Thus the denominator equals to the expected number of missing year papers $|V_P^U| = \eta |V|$. For each component G_i, $\eta^{|V_i|}$ is the probability that all the papers in it are missing year papers and $\eta^{|V_i|} \cdot |V_i|$ is hence the expected number of papers that will be labeled as *Uncovered*.

For the three types of the academic social networks, the above model actually cannot be applied directly. To apply it, we have to make proper modifications: (1) based on the citation network $G_P = (V_P, E_P)$, we construct $G'_P = (V_P, E'_P)$ by implicitly considering all the citation edges as undirected edges, where E'_P is the undirected edge set. (2) based on the paper authorship network $G_{AP} = (V_A \cup V_P, E_{AP})$, we build a coauthor indicator graph $G'_{AP} = (V_P, E_{PP})$, where the existence of an edge between two papers in G'_{AP} indicates that they have at least one common author, i.e., $\forall e_{i,j} \in E_{PP}, i, j \in V_P \Leftrightarrow A(i) \cap A(j) \neq \emptyset$, where $A(i)$ is the author set of paper i. (3) For the heterogenous network G, by simply combining G'_P and G'_{AP}, we obtain

$G' = (V_P, E'_P \cup E_{PP})$. Now the analytical model can be applied on G'_P, G'_{AP} and G' to calculate the expected coverage.

5.2.4 Experiment results in G_P

The first set of experiments are conducted on the citation network $G_P = (V_P, E_P)$. The coverage, MAE and RMSE results of algorithms G_P-SS, G_P-AS and G_P-AA are plotted in Figure 5.3.



(a) Coverage-Libra (b) Coverage-DBLP (c) Coverage-APS

(d) MAE-Libra (e) MAE-DBLP (f) MAE-APS

(g) RMSE-Libra (h) RMSE-DBLP (i) RMSE-APS

Figure 5.3: The Coverage, MAE and RMSE of algorithms G_P-SS (Simple Window Derivation and Simple Value Calculation), G_P-AS (Advanced Window Derivation and Simple Value Calculation) and G_P-AA (Advanced Window Derivation and Advanced Value Calculation) in paper citation network G_P of three data sets

As shown in Figure 5.3, we have the following observations:

1) For all the three algorithms, when η increases, coverage decreases while both MAE and RMSE increase. This implies that more available information helps to get better estimation results, more coverage and less estimation error.

2) In Fig. 5.3(a)-5.3(c), the curve of G_P-AS overlaps with that of G_P-AA and they have better coverage than G_P-SS. This is consistent with what we have discussed in Section 5.1 (G_P-AS and G_P-AA use the same advanced window derivation method). However, it appears that all the three coverage curves have certain deviation from the curve (with nodes of red "X" in Fig. 5.3(a)-5.3(c)) obtained by the analytical model in Eq. (5.21).

   The reason is that the analytical model overestimates the number of covered papers for G_P-AS and G_P-AA. Recall in Section 5.1, the window propagation method in G_P is different to the iteration scheme of $G_{AP} - Iter$ and $G_{AP} - AdvIter$ in that it follows the bound transmission rules in Eq. (5.8) and does not utilize estimation results in the previous rounds. As a result, the outcome (I) discussed above may not be always true, while (II) remains true. We use a typical and simple example to illustrate. As shown in Fig. 5.4, there are three papers (a, b, c) and two citation links, where only one paper b has year information while the other two are missing year papers. Fig. 5.4 plots all the 7 possible topologies.

   According to outcome (I) of the analytical model, neither a nor c will be labeled as *Uncovered*. However, in Fig. 5.4, paper a in case (6) and paper c in case (7) get *Uncovered* result by applying the advanced window derivation method in Eq. (5.8). Building a more precise analytical model for

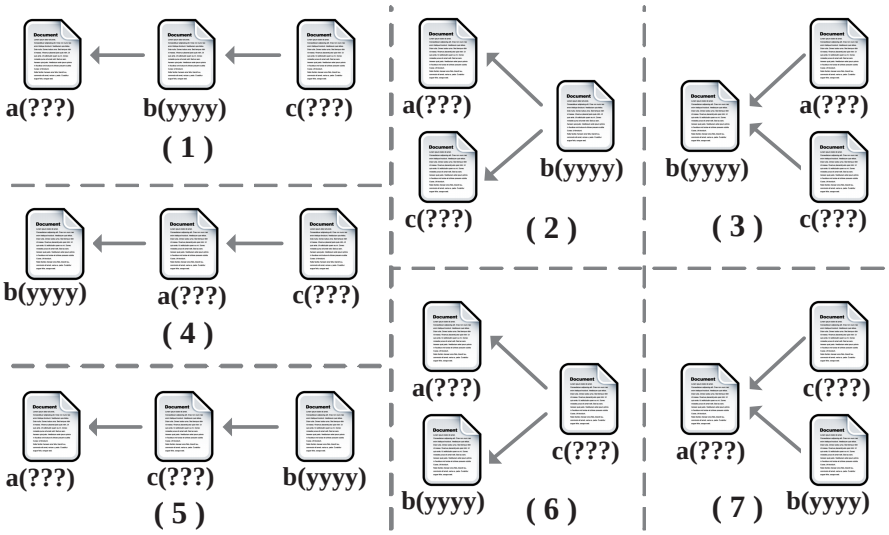


Figure 5.4: An example of paper citation network with three papers (a, b, c) and two citation links. When two papers are missing year (a and c), there are totally 7 possible topologies.

citation network, however, is too complicated. Therefore, we stick to use the current one in Eq. (5.21) as an upper bound for the coverage achieved in citation network G_P.

3) G_P-AA outperforms the other two for all network types and data sets in terms of both coverage and estimation accuracy, MAE and RMSE.

4) Comparing the three data sets, we find that the coverage on APS data is much higher than the other two and DBLP-Cit is the lowest. This is mainly caused by the completeness of the citation information of the three data sets, mentioned in the beginning of this section. Since APS maintains very complete and accurate citation information, this benefits both coverage and accuracy for the MYE in paper citation network (Fig. 5.3(c), Fig. 5.3(f), Fig. 5.3(i)).

5) In Fig. 5.3(a) and Fig. 5.3(b), the coverage on Libra case is higher than DBLP-Cit, however, its MAE and RMSE are

at similar level (or worse, e.g., G_P-AA in Fig. 5.3(d) and Fig. 5.3(e), all the curves in Fig. 5.3(g) versus Fig. 5.3(h)). We think one possible reason is that quantitatively, Libra has more complete paper citation information than DBLP-Cit, but qualitatively, the correctness of Libra data may be worse. We summarize this in Table 5.10.

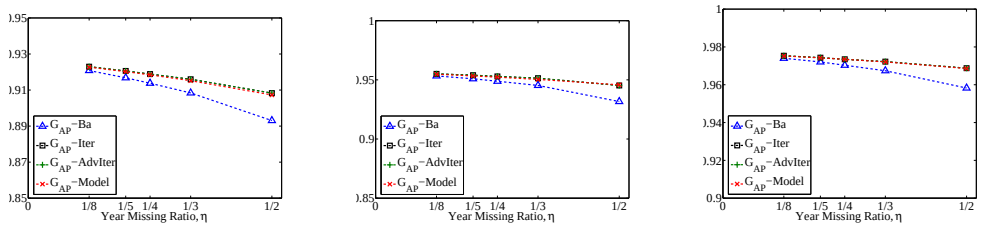
MYE performance in G_P	
Coverage	APS > Libra > DBLP
MAE/RMSE	APS < DBLP < Libra
Inferred data quality of paper citation information	
Completeness	APS > Libra > DBLP
Correctness	APS > DBLP > Libra

Table 5.10: Summary on data quality of paper citation information of three used datasets inferred from MYE performance in G_P.

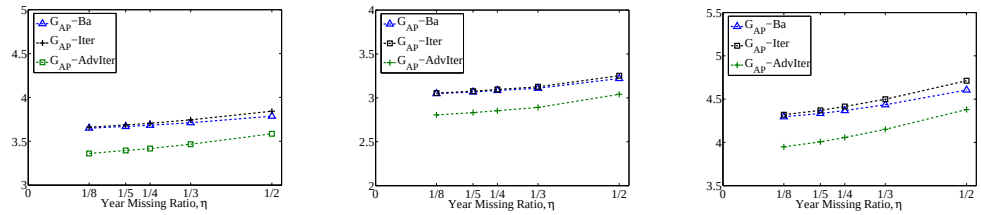
5.2.5 Experiment results in G_{AP}

The second set of experiments are conducted on the paper author bipartite network $G_{AP} = (V_A \cup V_P, E_{AP})$. The coverage, MAE and RMSE results of algorithms G_{AP}-Ba (the basic scheme), G_{AP}-Iter (Simple iteration of the basic scheme) and G_{AP}-AdvIter (Iteration with considering Consistent-Coauthor-Pair information) are plotted in Figure 5.5. Our observations are:

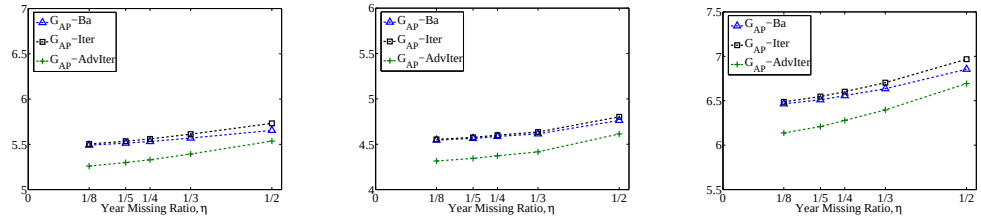
1) In Fig. 5.5(a)-5.5(c), the curve of G_{AP}-Iter overlaps with that of G_{AP}-AdvIter and they have better coverage than G_{AP}-Ba. As is discussed before (Section 5.1), G_{AP}-Iter and G_{AP}-AdvIter utilize the estimation results in the previous rounds for the later iterations (information propagation) which leads to the higher coverage results. In addition, the curves of G_{AP}-Iter and G_{AP}-Iter match quite well with the expected value generated by the analytical model.



(a) Coverage-Libra (b) Coverage-DBLP (c) Coverage-APS



(d) MAE-Libra (e) MAE-DBLP (f) MAE-APS



(g) RMSE-Libra (h) RMSE-DBLP (i) RMSE-ASP

Figure 5.5: The coverage, MAE and RMSE of algorithms G_{AP}-Ba, G_{AP}-Iter and G_{AP}-AdvIter in paper author bipartite network G_{AP} of the three data sets

2) In Fig. 5.5(d)-5.5(i), which concerns estimation accuracy, we find that G_{AP}-Iter obtains worse MAE than G_{AP}-Ba. This meets our anticipation (in Section 5.1 that the simple iteration scheme of G_{AP}-Iter spreads inaccuracy during the information propagation).

3) It shows that G_{AP}-AdvIter performs much better than the other two in both coverage and accuracy. For all different η, G_{AP}-AdvIter consistently makes around 10% improvement in MAE measures and 6% in RMSE measures.

4) If we compare the MAE curves of the three data sets in

Fig. 5.5(d)-5.5(f), the same algorithm generates the best MAE on DBLP-Cit data set, the worst on APS data set and intermediate on Libra data set. This result indirectly reflects the data quality (on paper-author relationship) of these three data sets, summarized in Table 5.11. As is widely known that, the original DBLP data set (with no citation information) is well managed and hence maintains the most complete and accurate paper-author/ paper-venue relationships [6, 54]. Libra is an object-level data set [4, 67], the process of the text-to-object transfer has been done before we obtain them. Different to the paper citation links, APS data set only provides pure text information of paper-author relationships, therefore, the text-to-object task is done by ourselves with some simple text-based matching scheme, which inevitably induces number of errors in G_{AP}. In fact, this involves several difficult and hot research problems in the community, for example the Author-Paper Identification Challenge and the Author Disambiguation Challenge in [8].

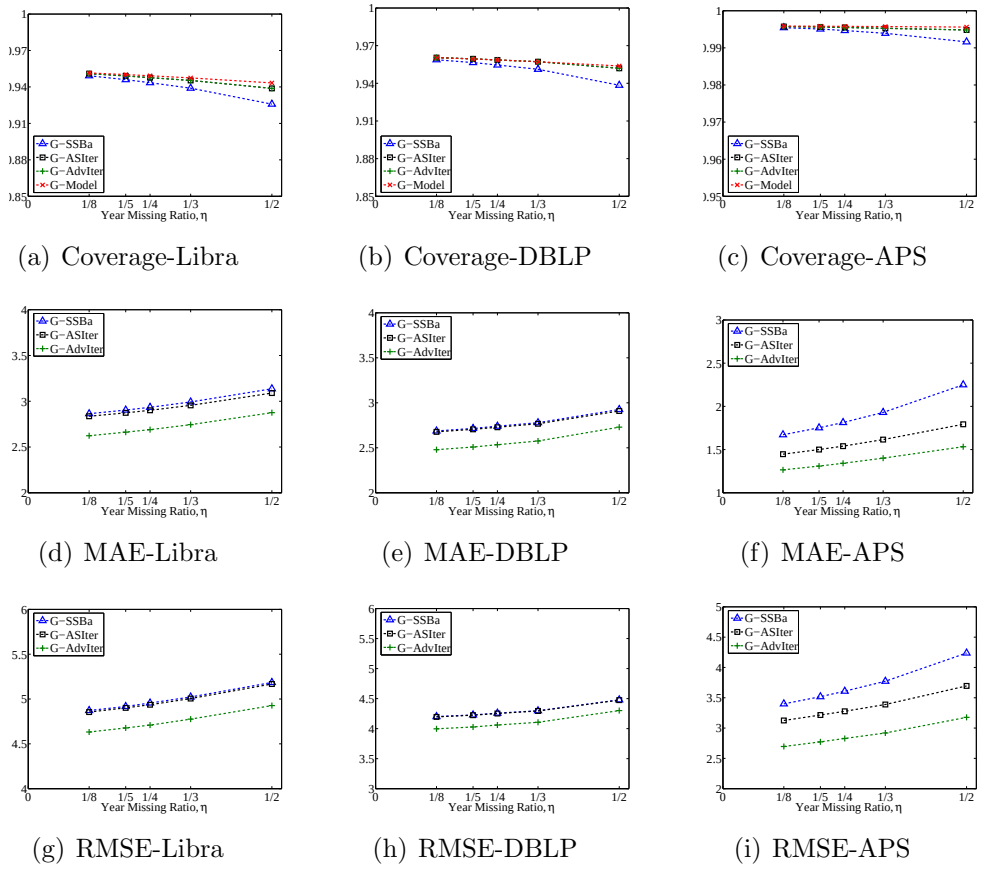
MYE performance in G_{AP}	
Coverage	DBLP $\approx$ Libra $\approx$ APS
MAE/RMSE	DBLP $<$ Libra $<$ APS
Inferred data quality of paper-author relationship	
Completeness	DBLP $\approx$ Libra $\approx$ APS
Correctness	DBLP $>$ Libra $>$ APS

Table 5.11: Summary on data quality of paper-author relationship of three used datasets inferred from MYE performance in G_{AP}.

5.2.6 Experiment results in G

The last set of experiments are conducted on the heterogeneous network $G = (G_P \cup G_{AP})$ which consists of both the paper citation network and the paper author bipartite network. The

coverage, MAE and RMSE results of algorithms G-SSBa (combination of G_P-SS and G_{AP}-Ba), G-ASIter (combination of G_P-AS and G_{AP}-Iter) and G-AdvIter (combination of G_P-AA and G_{AP}-AdvIter) are plotted in Figure 5.6.



(a) Coverage-Libra (b) Coverage-DBLP (c) Coverage-APS

(d) MAE-Libra (e) MAE-DBLP (f) MAE-APS

(g) RMSE-Libra (h) RMSE-DBLP (i) RMSE-APS

Figure 5.6: The Coverage, MAE and RMSE of algorithms G-SSBa, G-ASIter and G-AdvIter in the heterogenous network G of the three data sets.

We make three observations according to the results shown in Fig. 5.6:

1) All the curves have similar shapes as those in Fig.5.3 and Fig.5.5, but the results in Fig.5.6 have the highest coverage and smallest MAE and RMSE. This shows the advantage of the heterogeneous information (both paper citation and

paper author relationship) and the proper combination of the MYE algorithms in G_P and G_{AP}.

2) In Fig. 5.6(a)-5.6(c), there appears certain deviations (although milder than those in Fig. 5.3(a)-5.3(c)) from the coverage curves of G-ASIter and G-AdvIter to that generated by the analytical model. This is again due to the overestimation of the expected number of covered papers by the citation network information, since G-ASIter and G-AdvIter are the combinations from G_P-AS and G_P-AA respectively.

3) The G-AdvIter outperforms the other two for both coverage and accuracy (with around 8% improvement in MAE and 5% in RMSE for different η).

□ **End of chapter.**

Chapter 6

Conclusion and Future Work

Summary

In this chapter, we conclude our work and discuss the future work.

6.1 Conclusion

In this thesis work, we first present the design and experimental study of an Academic Social Network and Research Ranking system (`http://pubstat.org`) that we have built. It consists of several different non-conventional, social-network-like metrics we can use to rank authors and compare authors. In addition, it also provides author-based institution rankings by utilizing the author-institution relationship information. It has been demonstrated to numerous colleagues and collaborators, many of whom found it very useful.

Later on, we are dealing with the papers' missing publication year recovery problem in the academic social network. We have considered using three possible networks for estimating missing years: the paper citation network, the paper author bipartite network and the heterogenous network (the combination of

the previous two). In each network, we first propose a simple algorithm which is considered as a benchmark. Next another algorithm involving information propagation mechanism is proposed. The propagation mechanism helps to increase the estimation coverage ratio. Finally, an advanced propagation based algorithm is proposed, and in each of the three networks the advanced algorithm outperforms other algorithms and achieves at least 8% improvements on MAE and 5% on RMSE. In addition, the coverage achieved by the advanced algorithms well matches the results derived by the analytical model.

6.2 Future Work

Although we have had a working system for some time now, there are still many challenges to making it widely used. The publications database we have is not as complete as we would like; and we want to work out a way to continuously update it. We continue to discover new query types that users are interested in, and even new metrics.

As mentioned in section 4.1.5, the current paper-domain categorization information is provided by the Libra data set. The study on how to accurately and properly categorize papers into different domains is considered as a potential future direction. We can design approach based on the academic social network analysis results, or analysis on papers' keywords.

One possible extension in citation analysis is to consider the factor of citation roles, or citation behaviors [40, 41]. According to our own experience, when we write a paper, there are several reasons we cite existing papers, e.g., providing background studies, showing respect to the pioneers, etc. [40] had a detailed discussion on the various types of citation behaviors. To conduct this kind of study, the raw publication data (.ps or .pdf file) is required to collect so that we can extract additional cita-

tion information for investigating what type of citation behavior each citation belongs to.

The study of the dynamics of citation patterns and trends is a natural extension to the current results, which are based on a snapshot of publication database. There are many challenges, from the modeling and analysis of trends and dynamic patterns, to possible prediction of trends. How should we define journal impact factors? How do we measure the productivity of an author, a department or a university over time? Can we predict whether a hot research topic will turn out to be bubble? These are all interesting questions to explore.

The study of a group of authors [57] is another interesting dimension. As we mentioned before, many practical questions of interest involve evaluating groups of authors as an entity - for example a department, a research team, or a university. There are at least two approaches to explore: (a) aggregate the authors and their work as if it is a single author first, then analyze; (b) combine the scores of authors in some way (we have tried this in our thesis work). While (a) seems logically more sensible, (b) has its advantages as well, from the point of view of easy computation, or normalization.

At the aggregated level, e.g., a department or a research team, it is necessary to re-investigate the definition and properties of the existing ranking metrics. For example, "Connection" may not be a suitable metric to evaluate individual author's research performance, however, when applying at a group of authors, it can be a "good" indicator in identifying active research groups across different institutes, or inferring the role of an author in the group.

The data is still far from *clean*. Although we start to apply machine learning techniques to *recover* the missing publication year, the results are very preliminary. It is necessary to continue the work and design more sophisticated approaches to decrease

the estimation errors. For those MYE algorithms proposed in the thesis, we have only made use of the graph topology information. More textual and statistic information such as papers' keywords, the citing/cited half-life of the journals [43] in which papers have published and so on, can be further utilized and enhance the estimation accuracy and coverage.

In addition, the name disambiguation (e.g., author name, conference/journal name or institution name) problem is still a big challenge. More emphasizes and efforts shall be put on it. Another related problem which is to find advisor-advisee [82] relationships from the academic social network is also interesting and worth studying.

□ **End of chapter.**

Bibliography

[1] ISI Web of Knowledge, http://www.isiknowledge.com/.

[2] Elsevier Scopus, http://www.scopus.com/.

[3] Google Scholar, http://scholar.google.com/.

[4] Microsoft Academic Search, http://academic.research.microsoft.com/.

[5] CiteSeerX, http://citeseerx.ist.psu.edu/.

[6] DBLP - Computer Science Bibliography , http://www.informatik.uni-trier.de/~ley/db/.

[7] American Physical Society (APS), https://publish.aps.org/datasets/.

[8] KDDCup 2013 - Author-Paper Indentification Challenge, http://www.kdd.org/kdd2013/.

[9] Pulish or Perish, http://www.harzing.com/pop.htm/.

[10] Arnetminer, http://arnetminer.org/.

[11] Stanford Large Network Dataset Collection, http://snap.stanford.edu/data/.

[12] DBLP-Citation-network V5, http://arnetminer.org/citation/.

[13] ArXiv dataset for KDD Cup 2003, `http://www.cs.cornell.edu/projects/kddcup/datasets.html/`.

[14] IEEE Digital Library, http://dl.comsoc.org/comsocdl/.

[15] ACM Digital Library, http://portal.acm.org/dl.cfm/.

[16] US News Ranking - The Best Graduate Schools in Computer Science, `http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools/top-science-schools/computer-science-rankings/`.

[17] The QS World University Rankings By Subject 2013 - Computer Science & Information Systems, `http://www.topuniversities.com/university-rankings/university-subject-rankings/2013/computer-science-and-information-systems/`.

[18] The Academic Ranking of World Universities (ARWU by SJTU) 2012 in Computer Science, `http://www.shanghairanking.com/SubjectCS2012.html`.

[19] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proc. of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 7–15, 2008.

[20] E. Bakshy, B. Karrer, and L. A. Adamic. Social influence and the diffusion of user-created content. In *Proc. of the 10th ACM Conference on Electronic Commerce (EC)*, pages 325–334, 2009.

[21] P. Ball. Index aims for fair ranking of scientists. *Nature*, 436(7053):900–900, 2005.

[22] M. G. Banks. An extension of the hirsch index: Indexing scientific topics and compounds. *Scientometrics*, 69(1):161–168, 2006.

[23] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[24] C. T. Bergstrom. Eigenfactor: Measuring the value and prestige of scholarly journals. *College and Research Libraries News*, 68(5), 2007.

[25] C. T. Bergstrom, J. D. West, and M. A. Wiseman. The Eigenfactor^{TM} metrics. *The Journal of Neuroscience*, 28(45):11433–11434, 2008.

[26] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proc. of the 23rd international conference on Machine learning (ICML)*, pages 113–120. ACM, 2006.

[27] S. P. Borgatti, K. M. Carley, and D. Krackhardt. On the robustness of centrality measures under conditions of imperfect data. *Social Networks*, 28(2):124–136, 2006.

[28] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of the 7th international conference on Would Wide Web (WWW)*, 1998.

[29] S. Budalakoti and R. Bekkerman. Bimodal invitation-navigation fair bets model for authority identification in a social network. In *Proc. of the 21st international conference on World Wide Web (WWW)*, pages 709–718. ACM, 2012.

[30] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. In *Proc. of the SIAM inter-*

national conference on Data Mining (SDM), pages 131–142, 2011.

[31] D. M. Chiu and T. Z. J. Fu. "Publish or Perish" in the internet age: a study of publication statistics in computer networking research. *ACM Sigcomm Computer Communication Review (CCR)*, 40(1):34–43, 2010.

[32] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proc. of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 160–168, 2008.

[33] P. M. Davis. Eigenfactor: Does the principle of repeated improvement result in better estimates than raw citation counts? *Journal of the American Society for Information Science and Technology*, 59(13):2186–2188, 2008.

[34] D. J. de Solla Price. Networks of scientific papers. *Science*, 149(3683):510–515, 1965.

[35] D. J. de Solla Price. A general theory of bibliometric and other cumulative advantage process. *Journal of the American Society for Information Science*, 27(5):292–306, 1976.

[36] D. A. Easley and J. M. Kleinberg. *Networks, Crowds, and Markets - Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[37] L. Egghe. Dynamic h-index: The hirsch index in function of time. *Journal of the American Society for Information Science and Technology*, 58(3):452–454, 2006.

[38] L. Egghe. An improvement of the h-index: The g-index. *ISSI Newsletter*, 2(1):8–9, 2006.

[39] E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178(60):471–479, 1972.

[40] E. Garfield. When to cite. *Library, Quarterly*, 66(4):499–558, 1996.

[41] E. Garfield. Random thoughts on citationology its theory and practice. *Scientometrics*, 43(1):69–76, 1998.

[42] E. Garfield. Citation indexes for science. a new dimension in documentation through association of ideas. *International journal of epidemiology*, 35(5):1123–1127, 2006.

[43] E. Garfield. The evolution of the science citation index. *International microbiology*, 10(1):65–69, 2010.

[44] R. Ghosh, T.-T. Kuo, C.-N. Hsu, S.-D. Lin, and K. Lerman. Time-aware ranking in dynamic citation networks. In *Proc. of the 11th IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 373–380, 2011.

[45] M. Gupta, C. C. Aggarwal, J. Han, and Y. Sun. Evolutionary clustering and analysis of bibliographic networks. In *Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 63–70, 2011.

[46] S.-U. Hassan and P. Haddawy. Measuring internatinal knowledge flows and scholarly impact of scientific research. *Scientometrics*, 2013.

[47] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proc. of the National Academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[48] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *Proc. of the 16th ACM*

international conference on Knowledge discovery and data mining (SIGKDD), pages 663–672. ACM, 2010.

[49] H. Jeong, Z. Néda, and A.-L. Barabási. Measuring preferential attachment in evolving networks. *Europhysics Letters (EPL)*, 61(4):567–572, 2003.

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

[51] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the International joint Conference on artificial intelligence (IJCAI)*, volume 14, pages 1137–1145. Lawrence Erlbaum Associates Ltd, 1995.

[52] G. Kossinets. Effects of missing data in social networks. *Social networks*, 28(3):247–268, 2006.

[53] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, April 2009.

[54] M. Ley. DBLP: some lessons learned. *Proc. of the VLDB Endowment*, 2(2):1493–1500, 2009.

[55] D. Li, Y. Ding, X. Shuai, J. Bollen, J. Tang, S. Chen, J. Zhu, and G. Rocha. Adding community and dynamic to topic models. *Journal of Informetrics*, 6(2):237–253, 2012.

[56] H. Li, I. G. Councill, L. Bolelli, D. Zhou, Y. Song, W.-C. Lee, A. Sivasubramaniam, and C. L. Giles. Citeseer χ: a scalable autonomous scientific digital library. In *Proc. of the 1st international conference on Scalable information systems*, page 18. ACM, 2006.

[57] P. Li, J. X. Yu, H. Liu, and J. H. nd Xiaoyong Du. Ranking individuals and groups by influence propagation. In *Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 407–419, 2011.

[58] C. X. Lin, Q. Mei, J. Han, Y. Jiang, and M. Danilevsky. The joint inference of topic diffusion and evolution in social communities. In *Proc. of the 11th IEEE International Conference on Data Mining (ICDM)*, pages 378–387, 2011.

[59] R. K. Merton. The matthew effect in science. *Science*, 159(3810):56–63, 1968.

[60] C. D. Meyer. Matrix analysis and applied linear algebra. *SIAM Philadelphia*, 2000.

[61] F. Mosteller and J. W. Tukey. Data analysis, including statistics. *In Handbook of Social Psychology*, 1968.

[62] O. Mryglod, R. Kenna, Y. Holovatch, and B. Berche. Compparison of a citation-based indicator and peer review for absolute and specific measures of research-group excellence. *Scientometrics*, 2013.

[63] M. E. J. Newman. Coauthorship networks and patterns of scientific collaboration.

[64] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[65] M. E. J. Newman. The structure of scientific collaboration networks. *Proc. of the National Academy of Sciences of the USA*, 98(2):404–409, 2001.

[66] M. E. J. Newman. Who is the best connected scientist? a study of scientific coauthorship networks. *Complex Networks*, pages 337–370, 2004.

[67] Z. Nie, J.-R. Wen, and W.-Y. Ma. Object-level vertical search. In *Proc. of the Conference on Innovative Data Systems Research (CIDR)*, 2007.

[68] F. Radicchi, S. Fortunato, B. Markines, and A. Vespignani. Diffusion of scientific credits and the ranking of scientists. *Physical Review E*, 80(5):056103, 2009.

[69] S. Redner. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B-Condensed Matter and Complex Systems*, 4(2):131–134, 1998.

[70] H. Sayyadi and L. Getoor. FutureRank: Ranking scientific articles by predicting their future pagerank. In *Proc. of the 9th SIAM International Conference on Data Mining*, pages 533–544, 2009.

[71] P. O. Seglen. The skewness of science. *Journal of the American Society for Information Science*, 43(9):628–638, 1992.

[72] A. Sidiropoulos and Y. Manolopoulos. A citation-based system to assist prize awarding. *ACM SIGMOD Record*, 34(4):54–60, 2005.

[73] A. Sidiropoulos and Y. Manolopoulos. A new perspective to automatically rank scientific conferences using digital libraries. *Information processing and management*, 41(2):289–312, 2005.

[74] Q. Song. Similarity and comparison of academic ranking algorithms. *M.Phil Thesis*, 2012.

[75] M. J. Stringer, M. Sales-Pardo, and L. A. N. Amaral. Effectiveness of journal ranking schemes as a tool for locating information. *PLoS One*, 3(2):e1683, 2008.

[76] Y. Sun and C. L. Giles. Popularity weighted ranking for academic digital libraries. In *Proc. of the 29th European Conference on Information Retrieval Research (ECIR)*, 2007.

[77] J. Tang, A. C. M. Fong, B. Wang, and J. Zhang. A unified probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):975–987, 2012.

[78] J. Tang, D. Zhang, and L. Yao. Social network extraction of academic researchers.

[79] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *Proc. of the 14th ACM international conference on Knowledge discovery and data mining (SIGKDD)*, pages 990–998, 2008.

[80] D. Walker, H. Xie, K.-K. Yan, and S. Maslov. Ranking scientific publications using a model of network traffic. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06010, 2007.

[81] G. Walter, S. Bloch, G. Hunt, and K. Fisher. Counting on citations: a flawed way to measure quality? *Medical Journal of Australia*, 178(6):280–281, 2003.

[82] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *Proc. of the 16th ACM international conference on Knowledge discovery and data mining (SIGKDD)*, pages 203–212. ACM, 2010.

[83] D. Wang, X. Shi, D. McFarland, and J. Leskovec. Measurement error in network data: A re-classification. *Social Networks*, 2012.

[84] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles. Co-ranking authors and documents in a heterogeneous network. In *Proc. of IEEE International Conference on Data Mining (ICDM)*, 2007.

[85] X. Zhu, J. Lafferty, and R. Rosenfeld. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science.

Appendix A
Published Paper List

[1] L. Zhan, T. Z. J. Fu, D. M. Chiu, and Z. Lei. A framework for monitoring and measuring a large-scale distributed system in real time. In *Proc. of the HotPlanet Workshop*, 2013.

[2] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. On replication algorithms in p2p vod. *IEEE Transaction on Multimedia (TMM) Special Issue on Cloud-Based Mobile Media*, 15(4), 2013.

[3] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. An adaptive cloud downloading service. *IEEE/ACM Transaction on Networking (TON)*, 21(1):233–243, 2013.

[4] J. Park, T. Z. J. Fu, and D. M. Chiu. Networking, clustering and brodering keywords in the computer science research - analysis of the evolution using social network analysis. In *Proc. of the 7th IEEE International Conference on Digital Information Management (ICDIM)*, 2012.

[5] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. A unifying model and analysis of p2p vod replication and scheduling. In *Proc. of the IEEE INFOCOM*, pages 1530–1538, 2012.

[6] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. Server-assisted adaptive video replication for p2p vod. *Signal Processing:*

Image Communication Special Issue on Advances in 2D/3D video streaming over P2P networks, 27(5):484–495, 2012.

[7] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. Division-of-labor between server and p2p for streaming vod. In *Proc. of the 20th IEEE/ACM International Workshop on Quality of Service (IWQoS)*, pages 1–9, 2012.

[8] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. Statistical modeling and analysis of p2p replication to support vod service. In *Prof. of the IEEE INFOCOM*, pages 945–953, 2011.

[9] Y. Wang, T. Z. J. Fu, and D. M. Chiu. Design and evaluation of load balancing algorithms in p2p streaming protocols. *Computer Networks*, 55(18):4043–4054, 2011.

[10] J. Wang, T. Z. J. Fu, D. M. Chiu, and Z. Lei. Perceptual quality assessment on bd tradeoff of p2p assisted layered video streaming. In *Proc. of the IEEE Visual Communications and Image Processing (VCIP)*, 2011.

[11] T. Z. J. Fu, W. tung Leung, P. yin Lam, D. M. Chiu, and Z. Lei. Perceptual quality assessment of p2p assisted streaming video for chunk-level playback controller design. In *Proc. of the 18th International Packet Video Workshop (PV)*, pages 102–109, 2010.

[12] T. Z. J. Fu, D. M. Chiu, and Z. Lei. Designing qoe experiments to evaluate peer-to-peer streaming applications. In *Proc. of the IEEE Visual Communications and Image Processing (VCIP)*, 2010.

[13] D. M. Chiu and T. Z. J. Fu. Publish or Perish in the internet age: a study of publication statistics in computer networking research. *ACM SIGCOMM Computer Communication Review (CCR)*, 40(1):34–43, 2010.

[14] T. Z. J. Fu, Y. Hu, X. Shi, D. M. Chiu, and J. C. S. Lui. Pbs: Periodic behavioral spectrum of p2p applications. In *Proc. of the Passive and Active Network Measurement (PAM)*, pages 155–164. Springer, 2009.

[15] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang. Challenges, design and analysis of a large-scale p2p-vod system. In *Proc. of the ACM SIGCOMM conference on Data communication*, pages 375–388, 2008.

[16] Y. Wang, T. Z. J. Fu, and D. M. Chiu. Analysis of load balancing algorithms in p2p streaming. In *Proc. of the 46th IEEE Annual Allerton Conference on Communication, Control, and Computing*, pages 960–967, 2008.

[17] T. Z. J. Fu, D. M. Chiu, and J. C. S. Lui. Performance metrics and configuration strategies for group network communication. In *Proc. of the 15th IEEE/ACM International Workshop on Quality of Service (IWQoS)*, pages 173–181, 2007.

Appendix B
Submitted Paper List

[1] T. Z. J. Fu, Q. Song, and D. M. Chiu. The academic social network. *Submitted to Journal of Scientometrics*.

[2] T. Z. J. Fu, Q. Ying, and D. M. Chiu. MYE: Missing year estimation in academic social networks. *Submitted to Journal of Scientometrics*.

[3] Y. Wu, T. Z. J. Fu, and D. M. Chiu. Generalized preferential attachment considering aging. *Submitted to the Journal of Informetrics*.

Appendix C
Patent List

[1] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. "Replication Decision in P2P VoD Systems". U.S. Patent (SN.: US 13/111,786), Filing Date: May 19, 2011.

[2] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. "Methods for Replicating Media Contents and P2P VoD Systems". U.S. Patent (SN.: US 12/892,806), Filing Date: Sep 28, 2010.

[3] Z. Lei, T. Z. J. Fu, and D. M. Chiu. "System and Method for Evaluating Network Transport Effects on Delivery of Media Content". U.S. Patent (SN.: US 12/844,849), Filing Date: Jul 28, 2010.

[4] T. Z. J. Fu, and D. M. Chiu. "Systems and Processes of Identifying P2P Applications Based on Behavioral Signatures". U.S. Patent (SN.: US 12/018,676), PCT No: PCT/CN2009/070200, 2008. [**Exclusive Licensed**].