12-2013

Identifying Emerging Researchers using Social Network Analysis

Syed Masum Billah
University of Arkansas, Fayetteville

Identifying Emerging Researchers using Social Network Analysis

Identifying Emerging Researchers using Social Network Analysis

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science

by

Syed Masum Billah
Bangladesh University of Engineering and Technology
Bachelor of Science in Computer Science and Engineering, 2008

December 2013
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

_____
Dr. Susan Gauch
Thesis Director

_____
Dr. John Gauch
Committee Member

_____
Dr. Gordon Beavers
Committee Member

ABSTRACT

Finding rising stars in academia early in their careers has many implications when hiring new faculty, applying for promotion, and/or requesting grants. Typically, the impact and productivity of a researcher are assessed by a popular measurement called the h-index that grows linearly with the academic age of a researcher. Therefore, h-indices of researchers in the early stages of their careers are almost uniformly low, making it difficult to identify those who will, in future, emerge as influential leaders in their field. To overcome this problem, we make use of social network analysis to identify young researchers most likely to become successful. We assume that the co-authorship graph reveals a great deal of information about the potential of young researchers. We built a social network of 62,886 researchers using the data available in CiteSeer^{x}. We then designed and trained SVM and Naïve Bayes classifiers to learn how to identify emerging authors based on the personal and social aspects of a set of 3,200 young researchers, who had an h-index of less than or equal to four in 2005. We concluded that the success of young researchers largely depends on the number of their early citations, the number of their collaborators, and the impact and recent research activity of their collaborators.

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my committee chair Professor Susan Gauch; without her generosity of time, keen insights and encouragement, this thesis would not have been possible. I greatly enjoyed the academic freedom she gave me to explore different avenues of research, and the gentle nudges in the right direction when needed.

I would also like to thank my committee members, Professor John Gauch and Dr. Gordon Beavers. It has been an honor to me to have such two wonderful persons as members of my committee.

My ever grateful thanks go to my parents, Syed Ashrafuzzaman and Mohima Khatun who always inspire me to chase my dreams; thanks to my younger sister and all my brothers who tried very hard to wipe out my boredom with playful talks; and of course, my patient, tolerant, and beautiful wife Farhani for not being demanding when there was every reason to be, for being encouraging whenever I felt down, and for taking all the responsibilities by her own to raise our baby daughter Amatullah.

In addition, big thanks to all the departmental stuffs and the tech support teams who are always responsive and eager to help.

Finally, I would also like to thank NSF for their financial support granted through my master's study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Problem

Finding rising stars in academia is an interesting problem. When departments hire new, young faculty, they need a way to assess which of the many candidates show the best potential. When funding agencies or companies want to award funding, they want to send to researchers with the highest potential for having an impact on their field. Typically, the impact and productivity of a researcher are assessed by a popular, widely used metric called the *h-index* that is defined as follows: "a scientist has index h if h of his/her N_p papers have at least h citations each, and the other $(N_p - h)$ papers have no more than h citations each" [1]. Despite many criticisms, this simple measurement is being taken into account when a researcher is applying for promotion, requesting grants, or being interviewed for a new position. Often, new graduate students even choose their professors based on this score.

The h-index grows linearly with the academic age and productivity of researchers [2]. Although it can be reasonably accurate for established researchers, it fails to identify rising stars from among a group of young researchers. In the early stages of their careers, every researcher has an almost identical, low, h-index.

Social network analysis has gained considerable interest in recent years as a way of studying inter-relationships among individuals. In most approaches, the relationships between social actors are modeled as a graph, allowing a variety of new and existing graph algorithms to be applied. Applying social networks to a research community, co-authorship graphs have been widely studied, wherein nodes represent researchers, and edges represent co-authorship between pairs of nodes.

Properties of social graphs are described with respect to two levels: 'global graph metrics' and 'local graph metrics'. Global graph metrics consider the characteristic of the graph as a whole e.g., its diameter, mean node distance, betweenness, size of the giant component, clusters, small-worldness [8], etc., whereas the 'local metrics' relate to the features native to individual nodes such as degree, neighborhood, etc. [9]. Although they are well-defined, little work has been done to study the ability of these metrics to identify an author's impact.

1.2 Objective

We argue that the co-authorship graph reveals a great deal of information about the potential of young researchers. The basic idea is that young researchers with strong social connections to established researchers are more likely to have successful research careers. Our intuition is that these young researchers benefit from superior mentoring, and/or have strong colleagues who will continue to work with them as they establish their own, independent research careers. In this work, we will evaluate the ability of a variety of local graph metrics to identify, from among a set of new researchers, those who have the most potential to have an impact on their field. This addresses a weakness of the existing h-index, its inability to predict future success.

1.3 Approach

In this thesis, we study a social network of authors in Computer Science. To do so, we build a weighted, undirected graph in which authors are nodes, co-authorships, and the weights represent the number of papers on which the authors have collaborated. We focus our study on new authors within the social network, i.e., those with few publications and a low h-index. Our

goal is to identify which of the authors within that set emerge as influential researchers within a few years.

In this work, we define two classes for these new authors, namely 'emerging' and 'non-emerging' in terms of their h-index 6 years later. Then, we study the members of the two groups to identify which features of the authors and their social networks allow us to distinguish between the two classes of authors. With the class definitions and features in hand, we train a Support Vector Machine (SVM) classifier using the historical data available in CiteSeer^x database. Once the SVM is trained, it is used to predict the potential impact of unseen, young researchers.

In a nutshell, our contributions are as follows: (1) we offer a list of individual and social factors that are important for success in an academic position; and (2) we create a classifier to find emerging researchers from among a set of low-impact researchers.

1.4 Organization

The rest of the paper is organized as follows. In Section 2, we present the existing works on h-index and social network analysis in different use cases. Section 3 describes our system. Section 4 contains experimental results, and Section 6 summarizes our findings and offers suggestions for possible future improvements.

2. BACKGROUND

2.1 H-Index

In 2005, Hirsch proposed the h-index measure to characterize the cumulative impact of the research works of individual scientists [1]. Since then, it has been drawing widespread attention of the scientific community, policy makers, and the public media. It has been enthusiastically received by scientific news editors (e.g., Ball [11]), and researchers in various fields of science (e.g., Popov [12], Batista et al [13], etc.). At the same time, the concept of h-index has been criticized as well. Some of the criticisms are as follows: the h-index relies on pure citation counts treating all citations as equal and ignores the context of citations [3, 4]; 40% of citations were found to be irrelevant [5]; it never decreases, and does not account the number of co-authors of a paper [1].

However, in a study on committee peer review, Bornmann & Daniel found that, on average, the h-index for successful applicants for post-doctoral research fellowships was consistently higher than for non-successful applicants [14]. This particular result justifies our assumptions: although h-index does not accurately measure the productivity of young researchers, after a 5- or 6-year window, it is can be considered as an important success indicator.

2.2 Social Network Analysis

Social network analysis (SNA) is not a formal theory, but rather a wide strategy for investigating social structures. Wetherell et al. [24] defined SNA as follows:

> "social network analysis (1) conceptualizes social relationships as a network with ties connecting members and channeling resources, (2) focuses on the characteristics of the ties rather than on the characteristics of the individual members, and (3) views

communities as 'personal communities', that is, as networks of individual relations that people foster, maintain, and use in the course of their daily lives".

As pointed by many researchers such as Watt (2001), Scott (2000), Wasserman and Faust (1994), etc., SNA borrows most of its core concepts from sociometry, group dynamics, and graph theory [8, 9, 6]. Some of those borrowed notions and metrics are discussed in the following sections. Throughout our discussion, we use the terms graph and network interchangeably; same goes for node, actor, and author.

2.2.1 Common Concepts in Network Analysis

A *component* of a graph *G (V, E)* is a sub-graph *G' (V', E')*, where $V' \subseteq V, E' \subseteq E$, and there exists a path between any nodes in *V'*. If the whole graph forms one component, it is said to be fully connected.

The *path length* between two vertices is simply the count of intermediate edges between them.

The *characteristic path length* of a graph *G* is defined as the average shortest path length between every pair of vertices in *G*.

The *clustering coefficient* indicates how well the direct neighbors of a vertex are connected among themselves. For a given node *v*, let *G' (V', E')* be the sub-graph where *V'* is the set of direct neighbors of *v*, and *E'* is the set of edges from *E* between the nodes in V'. Then, the *clustering coefficient* of *v* is defined as $\frac{|E\prime|}{(|V'|*(|V'|-1))/2}$, or in words, it measures the number of edges between the direct neighbors of *v* as a fraction of all edges that could possibly exist between them. The average *clustering coefficient* over all nodes in *G* is the *clustering coefficient* of *G*.

A graph *G (V, E)* is called *random graph* if edges *E* are randomly selected from the set of all possible edges.

A graph *G (V, E)* is said to be a *small-world graph* if it has the following two properties: it has (i) a much higher *clustering coefficient* than similarly sized *random graph*, (ii) only a slightly larger *characteristic path length* than similarly sized *random graphs*.

A graph/network *G (V, E)* is *scale-free* if its degree distribution follows a power law, at least asymptotically. Mathematically, the probability distribution function $P(k)$ of the degree *k* of *scale-free* networks is described by: $P(k) \sim k^{-\gamma}$, where $\gamma > 0$ $(if\ k > 0)$ is called the scale-free exponent.

2.2.2 Centrality Measurement

Centrality measurements are used to describe the cohesion of a network, and the role played by particular nodes in that network. The most important centrality measures are as follows: (i) *degree centrality*, (ii) *closeness centrality*, (iii) *betweenness centrality*, and (iv) e*igenvector/eigenvalue centrality*.

Degree centrality of a node in an undirected graph is simply the number of edges adjacent to this node. For a node *i*, the degree centrality *d(i)* is defined by $d(i) = \sum_j m_{ij}$, where $m_{ij} = 1$ if there is an edge between nodes *i* and *j*, and 0 otherwise. For directed graphs, it becomes *in-degree* and *out-degree centralities* depending on the edge direction. In a co-authorship graph the *degree centrality* of a node is just the number of authors in the graph with whom he or she has co-authored at least one article.

Closeness centrality of a node *i* is equal to the total distance of *i* from all other nodes in the graph. Mathematically, closeness centrality, $c(i)$, of node *i* can be written as, $c(i) = \sum_j d_{ij}$,

where d_{ij} is the number of edges in a shortest path from node *i* to node *j*. It is an inverse measure of centrality since a larger value indicates a less central node while a smaller value indicates a more central. Individual closeness measures can be averaged to define global measure reflecting the cohesion of the entire network.

Betweenness centrality is defined as the number of shortest paths that pass through a given node. The mathematical expression for *betweenness centrality* of node *i*, denoted as *b*(*i*) is $b(i) = \sum_{j,k} g_{jik}/g_{jk}$, where g_{jk} is the number of shortest paths from node *j* to node *k* (*j*,*k* ≠ *i*), and g_{jik} is the number of shortest paths from node *j* to node *k* passing through node *i*. *Betweenness* is an indication to which a node facilitates the flow in the network.

Eigenvector/eigenvalue centrality is a measure of the 'importance' of a node in a network. It simply says if my neighbors are important, then I am important too. In other words, it assigns relative scores to all nodes in the graph based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

2.2.3 Applications of Social Network Analysis

Social Network Analysis (SNA) has a history of at least half a century, and it has produced many results related to disease and epidemic propagation; diffusion and information flow; social influence, inequality, groupings; and 'indeed almost every topic that has interested 20th century sociology' [6, 7, 9, 19]

SNA has been applied in epidemiology to reveal how patterns of human contact aid or inhibit the spread of diseases (e.g., Gonorrhoea) in a population [33]. Similarly, diffusion of innovations theory explores social networks and their role in influencing the spread of new ideas

and practices. By simply changing agents and opinions, leaders often play major roles in spurring the adoption of innovations, although factors inherent to the innovations also play a role [34].

According to Shishkin et al. (2009), human social networks may have a genetic basis [35]. Using a sample of twins from the National Longitudinal Study of Adolescent Health, they found that *in-degree*, *transitivity* (the probability that two friends are friends with one another), and *betweenness centrality* are all significantly heritable. Since existing models of network formation cannot handle this intrinsic node variation, they proposed an alternative "Attract and Introduce" model to explain heritability and many other features of human social networks.

In one study, Mark Granovetter (2007) found 'the strength of weak ties' as they can be important in seeking information and innovation. According to him, since cliques (connected components) exhibit homophilic tendency and share many ideas and many common traits, to find new information or insights, members of the clique should have to look beyond the clique to its other friends and acquaintances [36].

Diverse phenomena can spread within social networks. For example, there exists a number of scientific evidence that suggests that 'influence' can induce behavioral changes among the agents in a network. In 2007, Fowler, and Christakis conducted an intriguing study to determine whether obesity might also spread from person to person [17]. They concluded that a person's chances of becoming obese are increased by 57% if he or she had a friend who became obese in a given interval.

In another study (2008), the same researchers [37] have found that happiness also tends to be correlated in social networks: when an individual is happy, his or her nearby friends have a 25% higher chance of being happy themselves. Moreover, people at the center of a social network are more likely to be happier in the future than those at the periphery. Interestingly

enough, they also found that a person's happiness was associated with the level of happiness of their friends' friends' friends.

In 1967, Stanley Milgram [8] conducted one of the most widely discussed small-world experiments: he selected 296 US individuals as volunteers and asked them to dispatch a message to a specific person, a stockholder living in the Boston suburb of Sharon, Massachusetts. The volunteers were not supposed to send the message directly to the target person, but they should route the message along a chain of acquaintances. Milgram found that the average length of successful chains turned out to be about five intermediaries or six separation steps, which later gave birth to the famous phrase 'six degrees of separation'.

More recently, the emergence of online social networking services such as Facebook[1], LinkedIn[2], Twitter [3]etc. have revolutionized how social scientists study the structure of human relationships. These days, SNA techniques are constantly evolving to measure larger and larger representations of social networks. People do social networking for many reasons, ranging from collaboration between and/or within organizations, pursuit of interests, spending quality times, forming romantic relationships, or finding the right person for the right job, etc.

Currently, academic researchers continue to explore small-world phenomena within large online social networks. Using the entire Facebook network of active users (~721 million users, ~69 billion friendship links), Backstrom et al. (2012) carried out the largest Milgram-like experiment ever performed [38]. By applying HyperANF (an algorithm to study the distance distribution of very large graphs), graph compression, and the idea of diffusive computation, they

[1] www.facebook.com
[2] www.linkedin.com
[3] www.twitter.com

were able to compute the *characteristic path length* of Facebook graph, which was 4.74, corresponding to 3.74 intermediaries or "degrees of separation". Their result clearly indicates that the world is being smaller than before (six degree separation).

Another recent trend in online social networking is to build a social network for professionals (e.g. LinkedIn, ResearchGate[4], etc.) by encouraging users to construct an abbreviated CV and establishing "connections" [39]. These networks enable one to keep a relationship alive by maintaining awareness of others' activities. Among all professional networks, LinkedIn has the edge over others. Employers use LinkedIn for recruiting new employees or finding vendors; to learn more about people they have met or going to meet; or to get quick answers to professional questions from LinkedIn Groups.

2.3 Co-authorship Networks

Co-authorship networks, in which two researchers are considered, connected if they have co-authored one or more scientific papers together, are one of the most extensively studied social networks. In 1979, Garfield conducted early work in this area under the guise of citation network analysis [18]. In comparison to citation, co-authorship implies a much stronger social bond, since it is likely that pair of scientists who have co-authored a paper together are personally known to each other [19]. Currently, the publication record of scientists is well documented by a variety of publicly available electronic databases; and unlike citation data, co-authorship data are available immediately after the publication of a paper. This allows for the construction of large and relatively complete networks via automated means.

[4] www.researchgate.net/

One of the early examples of a co-authorship network is the Erdös Number Project, wherein the smallest number of co-authorship links between any individual mathematician and the Hungarian mathematician Erdös are calculated [25].

Newman (2001) studied co-authorship graph of four major databases (arXiv, Medline, SPIRES, and NCSTRL) and calculated different statistical properties such as the average numbers of papers per author, the average number of authors per paper, and the average number of collaborators per author in the various fields [19]. He found that distributions of these values roughly followed a power-law form, although there were some deviations that may be, according to him, 'due to the finite time window used for the study'. Besides distribution, it was shown that researchers in experimental disciplines were found to have more collaborators on average than those in theoretical disciplines. In second part of his work [20], he showed that those networks form a "small world". Additionally, he proved that for most authors the chunk of the paths between them and other authors in the network go through just one or two of their co-authors -- an effect called ''funneling'' [20].

Co-authorship analysis was further conducted by numerous researchers in different digital libraries, conferences, and journals with different flavors. For example, Smeatonet et al. (2002) constructed a co-authorship graph among authors of the 853 SIGIR conference papers to determine which author is the most 'central': the one who has the shortest average path length (*closeness centrality*) to all other authors in the graph. Their definition of 'central' was equivalent to find the 'Paul Erdös' in SIGIR community; and at that time, it was Chris Buckley (path length 3.65), followed by Gerry Salton (3.76), James Allan (3.791), and Clement Yu (3.862) [21]. An almost similar study was conducted by Nascimento et al. (2003) on SIGMOD community from 1975 to 2002. By computing the *clustering coefficient* and average

characteristic path length, they concluded that SIGMOD's co-authorship graph is just another "small world" [22].

Farkas et al. (2002) also analyzed the co-authorship networks derived from the data in mathematics and neuroscience, and modeled them as deterministic *scale-free* networks. Afterwards, they demonstrated the application of 'spectral graph theory' for the categorization of small measured networks [10].

Luong et al. (2012) suggested using co-authorship networks to recommend publication venues to the unpublished paper's authors based on the 'social similarity' they have with (i) conference Program Committee (PC) members, and/or (ii) with other authors who have publications in the conferences. After analyzing the co-authorship network over the data collected from the ACM digital library and Microsoft Academic Search [28], they showed that the recommendations generated by the second similarity measurement outperformed the baseline content-based recommender by a wide margin [40].

2.4 Influential and Emerging Authors

A large body of work has been dedicated to finding the 'influential' or 'center' nodes in co-authorship networks. From the preceding section, it is evident that the early efforts exploited different relatively simple graph metrics such as *degree centrality* [10], *betweenness centrality* [20], *closeness centrality* [21, 22], etc. to figure out 'social superstars' in the networks. More recently, a series of recursive algorithms that utilize the *eigenvalue centrality* are being used to measure the 'prestige' of the nodes in social network analysis [26]. Algorithms that fall into this category are heavily inspired by either of the two seminal works: (i) PageRank [41] or (ii) HITS [42].

PageRank [41] was originally developed by Page and Brin (1998) to rank web pages by their importance within the Google search engine. Although it was applied to a network in which nodes represented web pages and links hypertext references, one of its variants has been applied by Xiaoming et al. (2005) to a co-authorship network. In their work, called AuthorRank, they converted the binary undirected co-authorship graph into a weighted, directed one by the following means: (i) every undirected edge is replaced by two, symmetrical directed edge, (ii) authors that frequently co-author with each other receive higher edge weigh, and (iii) if an article has many authors, each individual co-author gets less weight. They applied their approach on a variety of conference PC members in the same period and found that AuthorRank outperformed *degree*, *closeness* and *betweenness* centrality metrics in identifying PC members, i.e., influential members of the research community [26].

Independent from PageRank, Kleinberg's (1999) HITS algorithm offers an improved notion of the importance of a web page by assigning two scores: a hub score and an authority score [42]. Adali et al. (2011) extended the original idea to propose another prominence ranking in heterogeneous, tri-partite networks wherein actors (authors) collaborate with each other to create artifacts (e.g., papers) that show up in some groups (e.g., conferences). Furthermore, they utilized the concept that when a social tie between actors is inferred by their participation in some artifact, the properties and relations between those artifacts can significantly improve the ranking (as opposed to only using the co-authorship ties among the actors). When the results were validated against the citation count (collected externally) of individual actors, the algorithm showed off a clear advantage over other well-known ranking methods [16].

More recently, Irfan et al. (2013) took a somewhat different, game theoretic approach to the study the influence in large, finite networks (e.g., the network of the U.S. Supreme Court Justices and the network of U.S. senators) that capture the strategic aspects of complex interactions. While comparing with equivalent random graph, they showed that their 'influence game' algorithm can not only predict stable behavior of the actors, but also compute the most influential actors and its variants (e.g., identify a small coalition of senators that can prevent filibuster) [15].

We have summarized several existing projects that apply social network analysis to co-authorship graphs; they all focus on finding the most influential authors. Although this is an interesting problem, it is also a problem that the existing h-index does reasonably well in academic environment. Our goal is not to find the influential nodes but rather to tackle a problem for which the h-index is poorly suited. We show that social network analysis can be used to

identify ‘rising-stars’ from among a group of new authors. While influence is a global phenomenon in a graph that previous work identifies using the global graph metrics such as *betweenness*, *closeness*, and *eigenvalue* centralities, emergence is purely a local aspect of a node (*degree* centrality). Thus, we focus our approach on calculating, and evaluating, local node metrics.

3. ARCHITECTURE

In this chapter, we present our system for identifying emerging authors. Figure 1 diagrams the main components of system architecture. It consists of an Author Database, a persistent, huge digital library of scientific works; a Social Network Builder; an Author Impact Rater; and an Emerging Author Identifier module. In the following sections, we will discuss each of these modules in more detail.

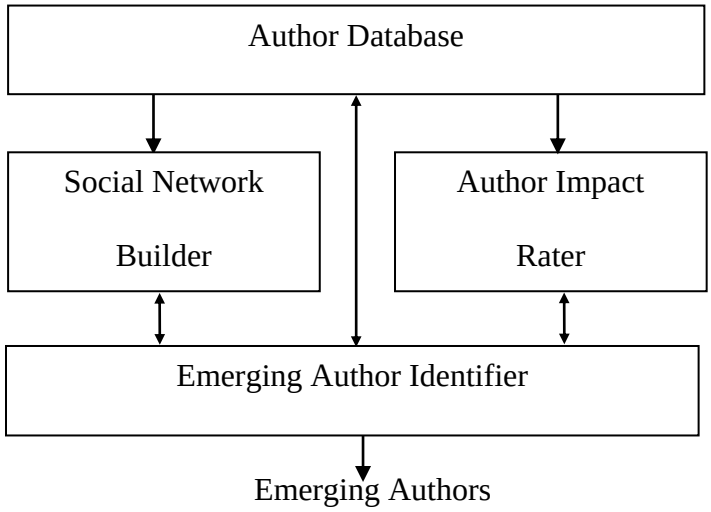

Figure 1: High level Block Diagram

3.1 Author Database

One difficulty in building a social network of authors is to accurately identify all of their papers. Author names may appear in many different formats, so we need to normalize the names and collect information on a per-author basis rather than a per-name basis. The main purpose of this module is to provide fully qualified name of the researchers together with their publications and citations record. It also contains a rich set of metadata associated with each scientific paper

such as publication year, venue, bibliography, citations by year, etc. Figure 2 depicts the subcomponents of Author Database module.

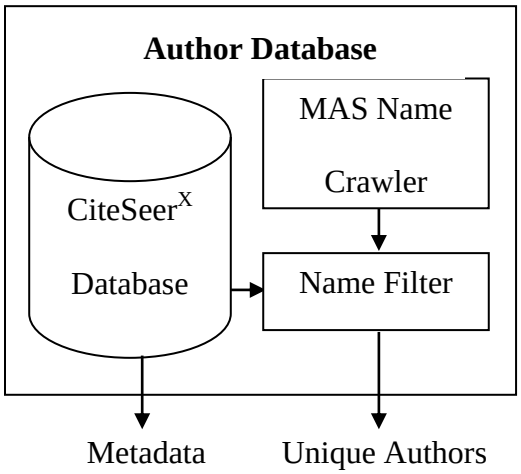

Figure 2: Author Database Module

3.1.1 CiteSeer^x Database

Our primary source of data is CiteSeer^x, a well-known scientific document digital library. It is an automatic citation indexing system that indexes academic literature in electronic format (e.g. Postscript files on the Web) [27]. After locating and downloading Postscript files that are available on the internet, CiteSeer^x analyzes and extracts bibliographical information from the downloaded files. As of now (2013), it contains 308,116 authors from different academic disciplines; 2,190,179 entries for papers; and 25,982,373 citation records. Since the whole library is built in automated manners, there are many identity (e.g., name, paper) duplications, ambiguities, and noise. Thus, we need to disambiguate the names using another source of information.

3.1.2 MAS Name Crawler

Microsoft Academic Search (MAS) [28] provides services almost similar to CiteSeer^x, and it is less noisy. Papers are associated with authors, regardless of the format in which the name appears in the paper. MAS also provides a list of authors sorted by 'Field Rating' which is similar to h-index but limited to within a specific field of study. Although we use CiteSeer^x as the basis of information for our social network, we make use of the disambiguated author names available in MAS, using a crawler to collect the 99,982 canonical names of researchers in the field of Computer Science.

3.1.3 Name Filtering

Our next goal is to identify unique authors from ambiguous names in the CiteSeer^x database. We have two sets of names: 99,982 canonical names ('first name', 'middle name/initial',' last name') from MAS and 308,116 noisy names from CiteSeer^x. To identify unique authors in CiteSeer^x, we take the intersection of these two sets, ending up with 62,884 names (exact matches). We expect each of these names represent unique authors, although there might be some homonymous authors.

3.2 Social Network Builder

This module (Figure 3) takes input from Author Database module and builds co-authorship multigraph. Afterwards, this multigraph representation allows us to generate any instance of co-authorship graph at any specific time/year, t.

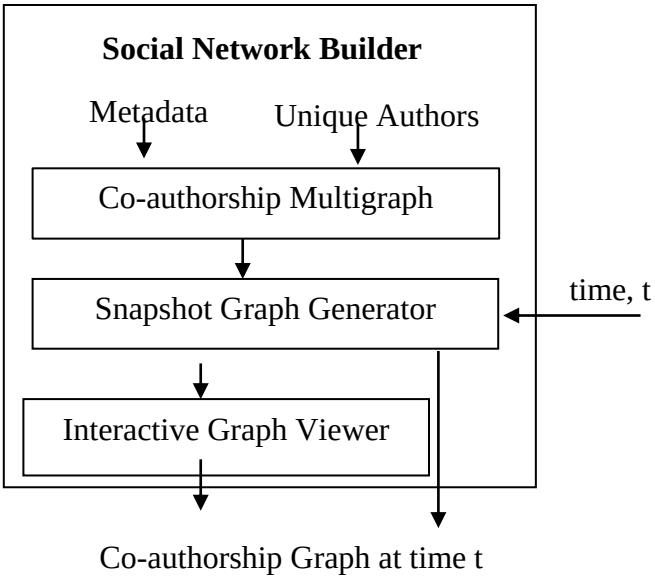

Figure 3: Social Network Builder Module

3.2.1 Co-authorship Multigraph Builder

Our co-authorship network is basically an undirected, multigraph G (V, E) where each edge represents a temporal co-authorship relationship. Therefore, it is obvious that a node-pair in G can have multiple temporal edges depending on the number of papers they've co-authored with. Keeping the magnitude and the scalability of this graph in mind, we choose to use Neo4j, an open source graph database that can handle number of nodes as many as 32 billion [29]. The multigraph generation steps are given in Table 1.

Table 1: Co-authorship Multigraph Generation Algorithm

1:	Input: ***A*** = Set of Authors
2:	Set of nodes, ***N*** = Φ; Set of edges, ***E*** = Φ
3:	for each author *a* in ***A***:
4:	*(i)* create *a*'s representative node *n* in *Neo4j DB* (ii) ***N*** = ***N*** ∪ {n}
5:	for each *n* in ***N:***
	(i) grab the list of papers, ***P*** written by *n* with metadata (e.g., publication year) from the CiteSeer[x] (ii) for each paper *p* in ***P:***
	a. extract the list of co-authors, ***C*** of *p* b. for each pair of nodes (n_1, n_2) in ***C*** such that $n_1 \in \boldsymbol{C}$, $n_2 \in \boldsymbol{C}$, $n_1 \in \boldsymbol{N}$, *and* $n_2 \in \boldsymbol{N}$:
	create an edge $e(n_1, n_2)$ with the attribute *'publication year'* of *p* in *Neo4j* $\boldsymbol{E} = \boldsymbol{E} \cup \{e\}$
6:	return multigraph ***G (V, E)***

Using the above algorithm on the 278,904 papers authored by our disambiguated authors, we build a social network that contains 62,886 nodes (authors) and 795,594 links (co-authorship relationships). As we mentioned earlier, this social network is stored in a NoSQL graph database called Neo4j.

3.2.2 Snapshot Graph Generator

To generate a snapshot of multigraph G at a particular time requires only the merging of multiple edges between each pair of nodes under certain condition(s). For example, to get a co-authorship graph up to the year 2005, we simply (i) count the number of edges between each pair of nodes in G with property 'publication year' $\leq$ 2005, and (ii) replace those edges with a single one having weigh equal to the count. Therefore, the snapshot graph is an undirected weighted graph. Figure 4 shows a 2-level deep co-authorship graph as of 2005 for an arbitrary author, *Konstantina Papagiannaki*. The graph is rendered by our graph viewer, described in the next section.

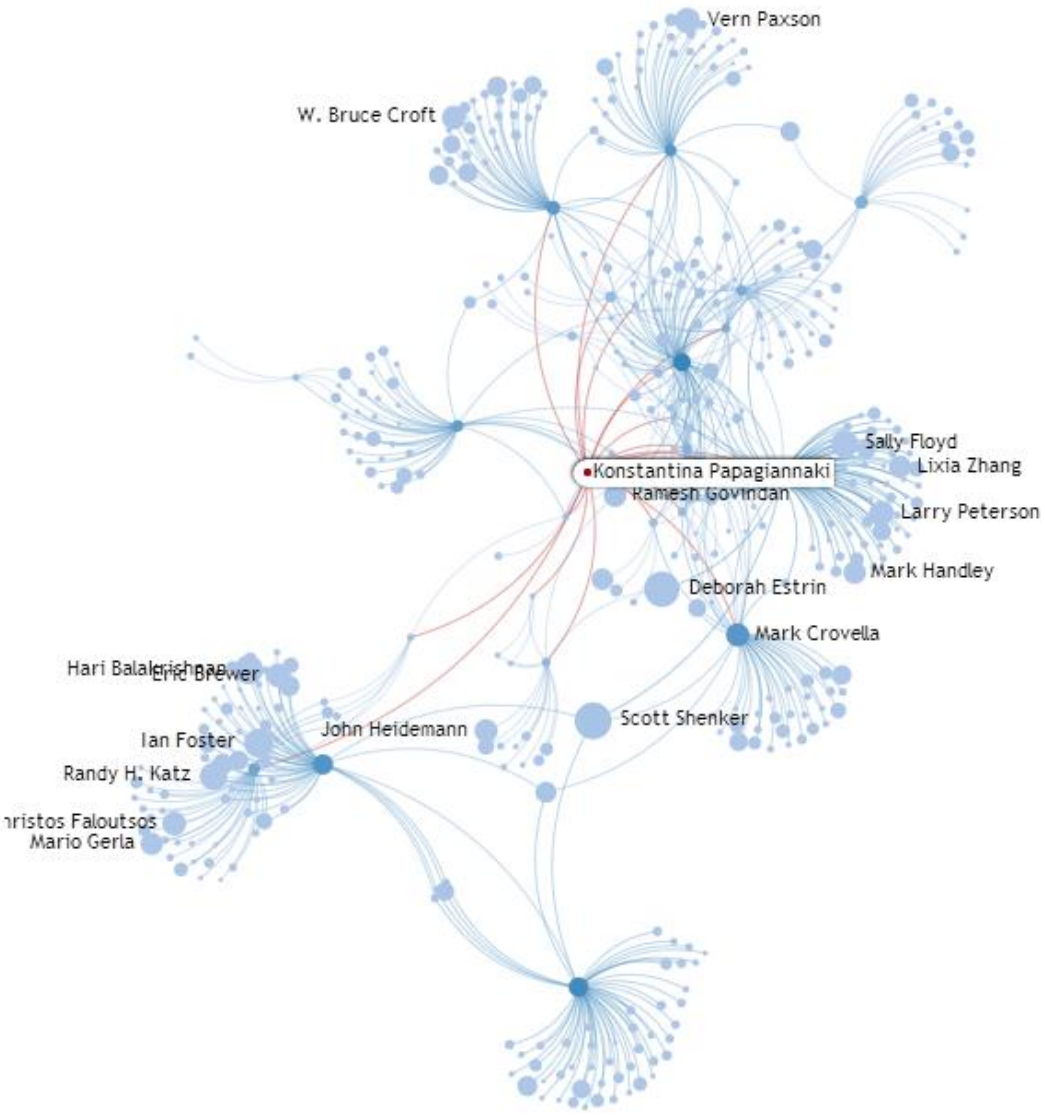

Figure 4: Co-authorship Graph for an Arbitrary Author (*Konstantina Papagiannaki)*

3.2.3 Interactive Graph Viewer

We also developed an online, interactive co-authorship graph viewer[5] with many useful features. Figure 5 shows the internal components of this sub-module. Briefly, it is comprised of two entities: server and client, each of which has several logical components.

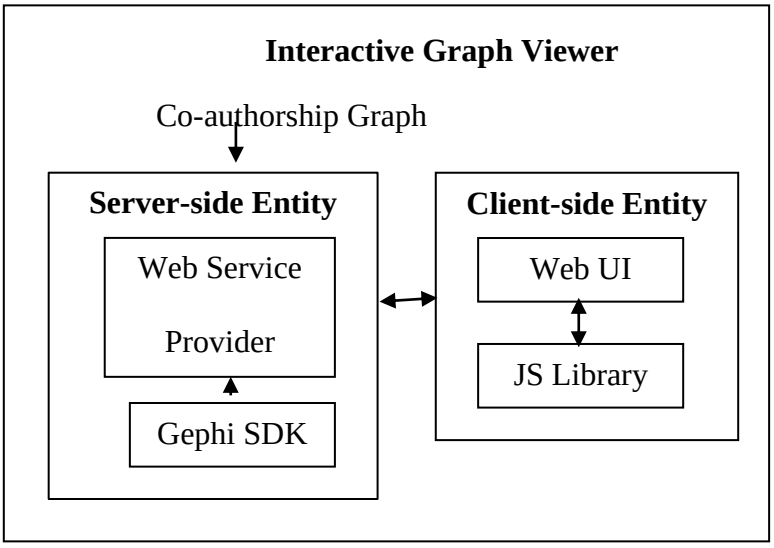

Figure 5: Interactive Graph Viewer Sub-module

3.2.4 Server-side Entity

We use open source Apache Tomcat[6] as web server and servlet container developed by the Apache Software Foundation (ASF). It provides a "pure Java" HTTP web server environment for Java code to run in.

Web Service Provider:

We offer RESTful web services by implementing JAX-RS API introduced in Java SE 5. Some of our web services are given below:

---

[5] http://citeseer.uark.edu:8480/graphs/pages/graph.htm
[6] http://tomcat.apache.org/

1. Search authors by name (either first name or last name of both),
2. Get an author's profile (e.g., name, number of publications, h-index, number of collaborators, etc.) either by his *citeseer_id* (primary key in CiteSeer[x] database) or *node_id* [7](primary key in Neo4j graph database),
3. Get an author's co-authorship graph by her *citeseer_id* or *node_id*, together with depth [8](how many hops to fetch from that author) and year (which snapshot?) parameters.

Our web services are public and any web client can make requests and consume one or more of them. To carry out any service request, the provider contacts the Social Network Builder module to get the appropriate graph data in JSON format.

Gephi SDK[9]:

Gephi is an open-source network analysis and visualization software package written in Java [43]. It supports multiple graph layout algorithms such as Force Atlas, Yifan Hu [44], etc.; it calculates graph centralities such as degree, betweenness, closeness, etc., and allows node/edge's size and color to be proportionate to a measurement. For our interface, we use the Gephi API prior to sending graph data to the clients. Currently, we applied Yifan Hu layout on our graphs, and the nodes' colors and sizes are proportional to their degrees and h-indices, respectively.

[7] citeseer.uark.edu:8480/graphs/rest/graphs/node/{node_id}
[8] citeseer.uark.edu:8480/graphs/rest/graphs/path/coauthor/{node_id}/{depth}
[9] https://gephi.org/

All of our Java classes, servlets, 3rd party libraries (e.g., Gephi API, Neo4j API etc.), Web pages (HTML and related files), and configuration files are bundled into a single .WAR (Web Application Archive) file and deployed into the Tomcat's *webapps* directory.

3.2.5 Client-side Entity

We used HTML and Java Script libraries to develop our web interface. User requests are translated into Ajax (Asynchronous JavaScript or XML) calls, minimizing the data exchange between server and client.

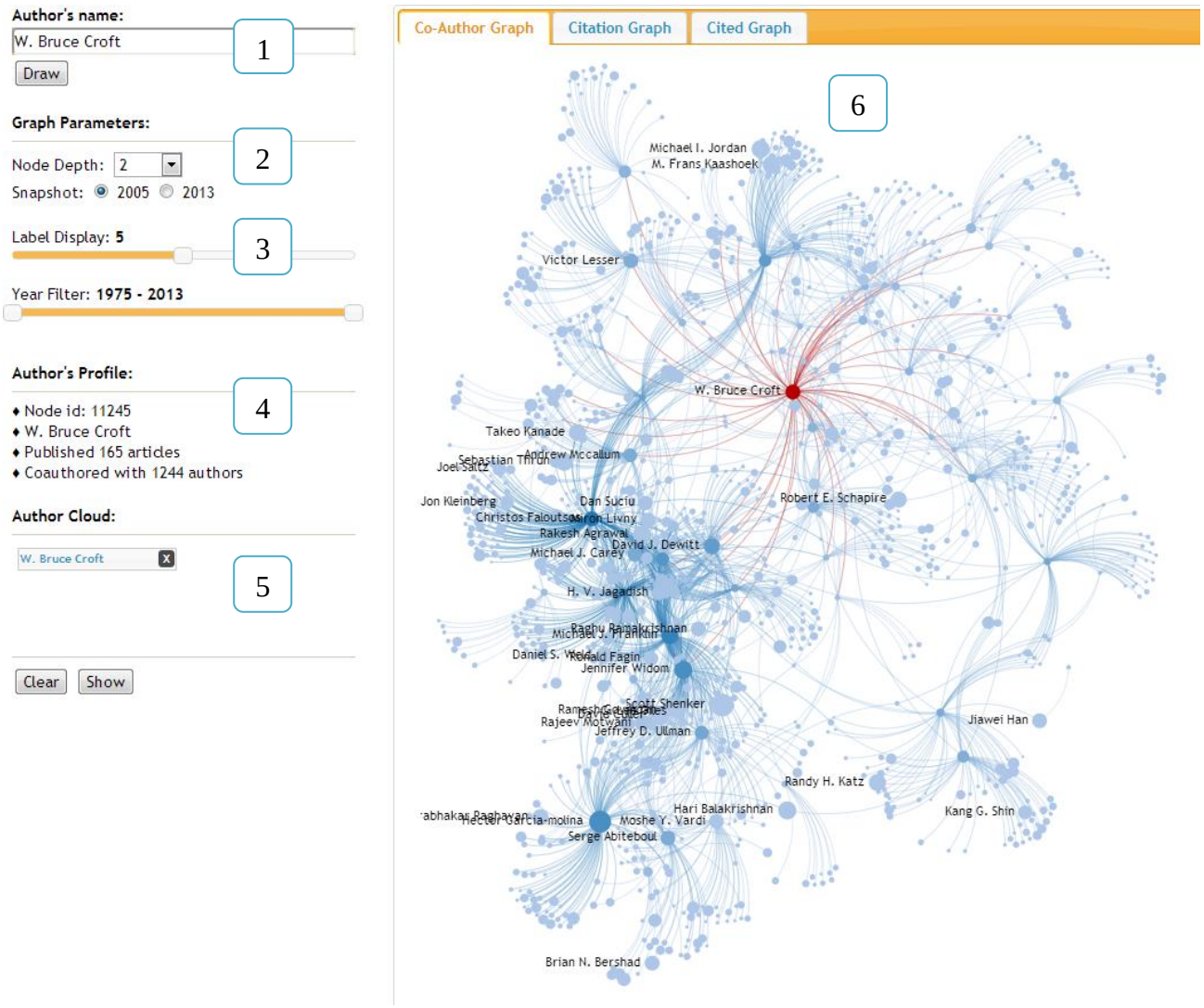

Figure 6: Screenshot of our Interactive Graph Viewer

Web UI:

A screenshot of our interactive User Interface (UI) is displayed in Figure 6. The functions supported by the UI (highlighted in Figure 6) are given below:

1. *Autocomplete Search Box:* Users can search for an author by starting to type his or her first name or last name, or both. The autocomplete box suggests a list of possible names from Graph database that match fully or partially to the typed text.
2. *Graph Parameters:* Currently there are only two parameters available for the users: node depth (1 to 3) and snapshot year (2005 or 2013)
3. *Display Parameters:* The visibility of the labels of the nodes is tunable. Similarly, the Year filter controls the visibility of nodes and edges by time period.
4. *Author's Profile:* This displays quick information about an author.
5. *Author Cloud:* This show the most recently viewed authors for quick re-selection and display.
6. *Visual Panel:* This panel displays the preprocessed co-authorship graph (received from the server). We incorporate the following user interactions: (i) zoom in/out (by mouse wheel), (ii) graph scrolling i.e., left-right and up-down (either by keyboard arrows or mouse drag), and (iii) popup menu (by left-click on a node), and (iv) the popup menu contains several useful actions and links.

JS (Java Script) Library:

We also use two Java Script (JS) libraries. Descriptions of those are given below.

1. *jQuery*[10]: it is an open source cross-browser JS library designed to simplify the client-side scripting of HTML. Our UI segments 1, 2, 3, and 5 are geared by *jQuery*.

[10] http://jquery.com/

2. *sigma.js*[11]: it also an open-source lightweight JS library to draw graphs on HTML canvas element. Our Visual Panel (segment 6) is mechanized by this library.

3.3 Author Impact Rater

The primary purpose of this module (Figure 7) is to compute the impact factors (h-index) of the authors in the 'Author Database' module, as of a given year. Then, based on the impact scores, it generates a list of low-impact authors at time *t* for the next module.

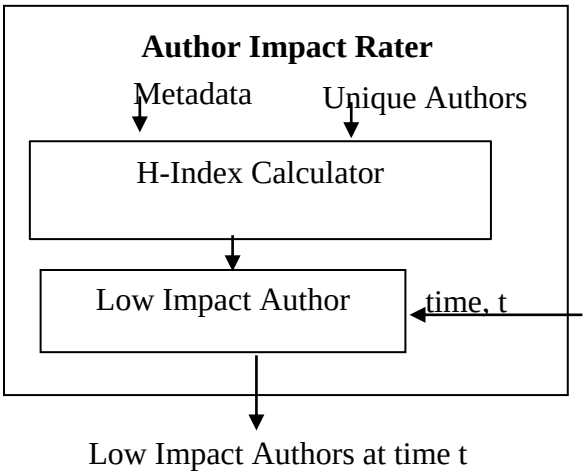

Figure 7: Author Impact Rater Module

3.3.1 H-Index Calculator

We calculate the h-index of individual author using the metadata available in CiteSeer[x]. For a particular author, we grab all the papers he or she has, and sort those papers by their citations. Publications and citation data are collected from CiteSeer[x]. The detailed algorithm is given in Table 2.

[11] http://sigmajs.org/

Table 2: H-index Calculator

1:	Input: ***a*** = author id, ***t*** = time/year
2:	*h-index* = 0; hash-table *ht* = Φ;
3:	for each paper *p* written/co-authored by ***a*** if *p*'s publication year <= ***t:*** *ht[p]* = number of citations of *p*
4:	*sort ht by value*
5:	*for p in sorted ht:* *if ht*[*p*] >=*h-index* *h-index++*
6:	return *h-index*

3.3.2 Low Impact Author Selector

According to Bornmann et al. [14], h-index of 5.15 is an indication of a successful researcher. Based on their work, we define 'low-impact' authors as authors having h-index <= 4. Therefore, this sub-module outputs a collection of authors having h-index <= 4 as of year *t*.

3.4 Emerging Author Identifier

From the feeds of 'Social Network Builder', 'Author impact rater', and 'Author Database' modules, this module performs all the tasks necessary to predict emerging authors, i.e., those whose research impact is likely to increase substantially in the years to come. It consists of a class labeler, feature extractor, dataset builder, and classifier (Figure 8).

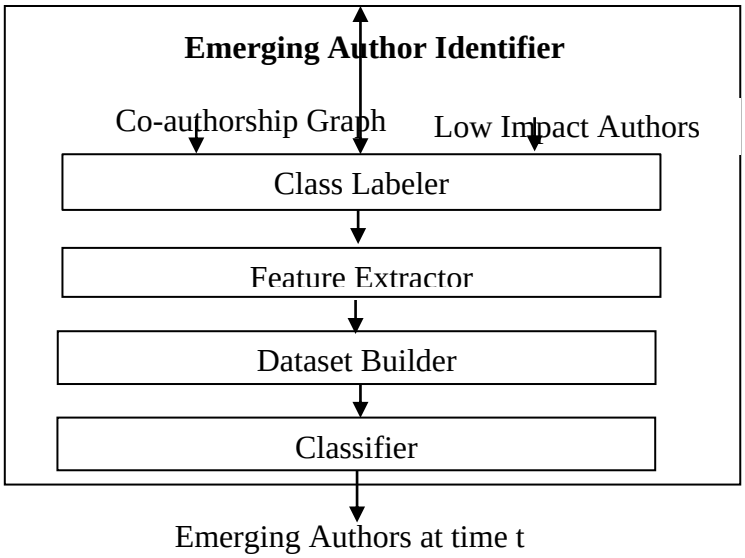

Figure 8: Emerging Author Identifier Module

3.4.1 Class Labeler

We try to identify whether or not a low-impact author is likely to emerge as a successful researcher based on his or her historical data available in CiteSeer[x]. For a researcher *r*, we define $h\text{-}index_t(r)$ as *r*'s h-index at time *t*. Then, in a 6-year window, we define 'emerging' and 'non-emerging' authors as follows:

Table 3: Class Labels for Low-impact Authors

Class	Label	Definition
E	*Emerging*	$h\text{-}index_t(n) \leq 4$, and $h\text{-}index_{t+\Delta t}(n) - h\text{-}index_t(n) \geq 4$, where $\Delta t = 6\ years$
NE	*Non-emerging*	$h\text{-}index_t(n) \leq 4$, and $h\text{-}index_{t+\Delta t}(n) - h\text{-}index_t(n) < 4$, where $\Delta t = 6\ years$

3.4.2 Feature Extractor

After defining the classes, the next step is to represent each class member as vector of features. In order to understand the nature of emerging authors, we generate a snapshot of the co-authorship graph at *t = 2005*; compute the author impact at *t = 2000, 2005, and 2011*. Thus, for the authors whose social networks are known as of 2005, we can look at publication productivity for the 4years prior and the 6 subsequent years.

To build our intuition about the relationship between a low-impact author's social network and their future research success, we randomly selected 15 low-impact authors at *t = 2005* and extracted their 1-level deep neighborhood graphs (see Figure 9 to Figure 25). In each of these graphs, the center node is the author being studied, i.e., Engin Kirda (Figure 9), Konstantina Papagiannaki (Figure 10), Byron Cook ((Figure 11), Marco F. Duarte (Figure 12), Sven Apel (Figure 14), etc. The size of each node represents the change in h-value from 2005 to 2011 (Δh_{2011}), and the color represents the h-index value as of 2005. Thus, a large, dark circle indicates a researcher who had high h-index as of 2005 and whose h-index (or number of citations and publications) grew from 2005 to 2011.

In the next 3 tables, Tables 4 through 6, we present the co-authorship graphs of 15 authors in 2005 who, by 2011, either fit our definition of emergence (Table 4) or not (Table 5 and Table 6). Each of the nodes is labeled by the following order: author's name -- number of publications at 2005 -- h-index at 2000 – increase of h-index from 2000 to 2005 -- increase of h-index from 2005 to 2011.

Table 4: Social Networks of Emerging Nodes (highly active authors)

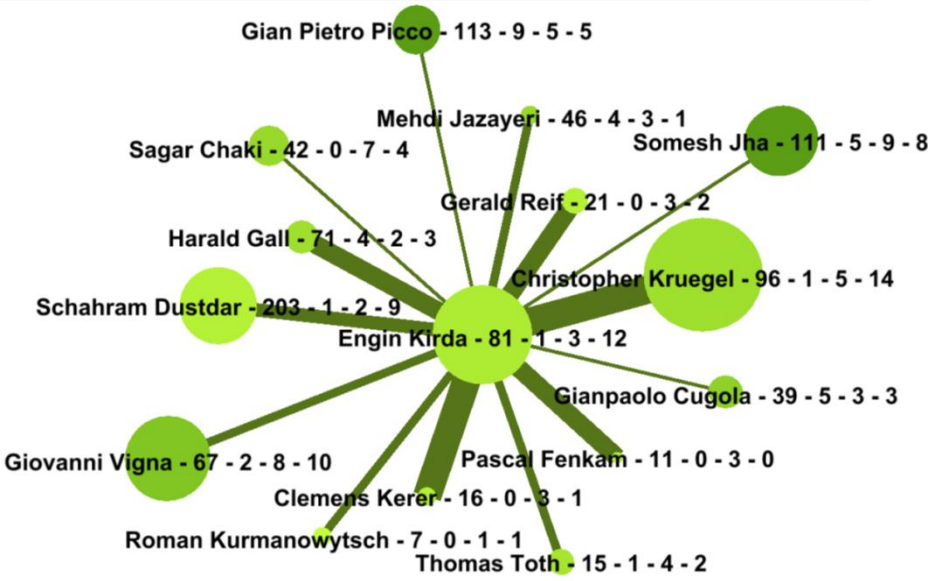

Figure 9: Engin Kirda

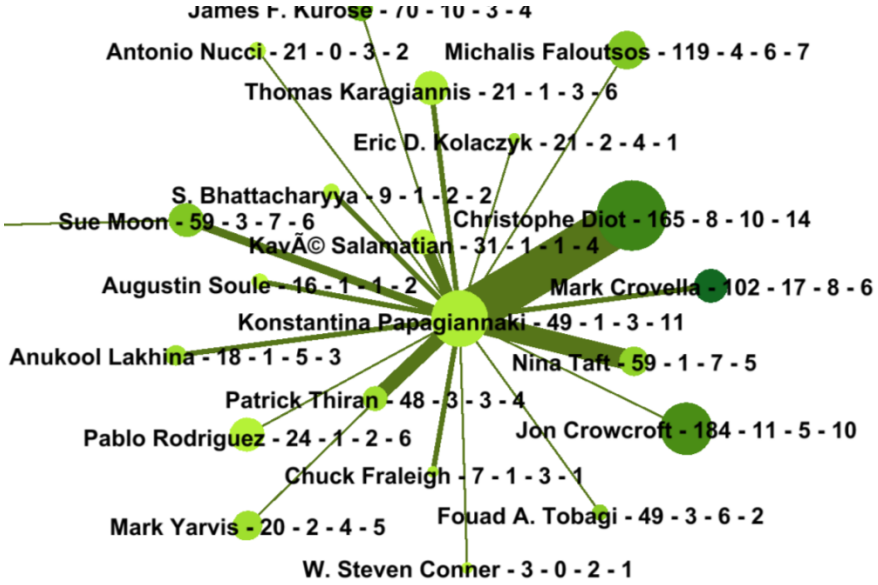

Figure 10: Konstantina Papagiannaki

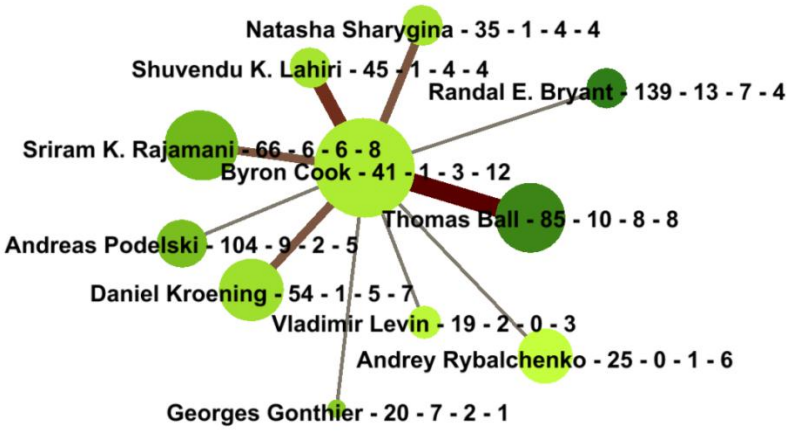

Figure 11: Byron Cook

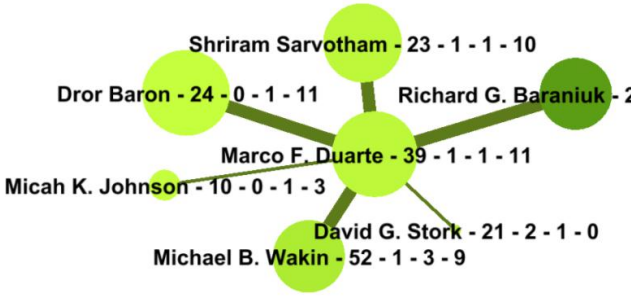

Figure 12: Marco F. Duarte

Figure 13: Marco F. Duarte (MAS)

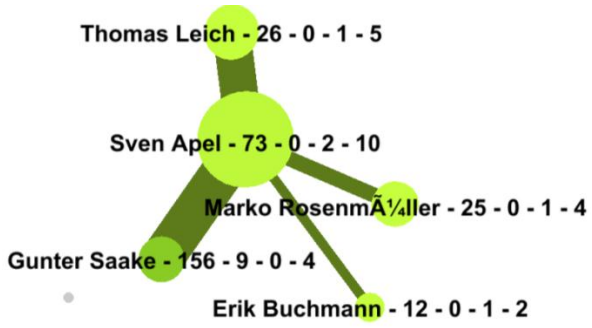

Figure 14: Sven Apel

Figure 15: Sven Apel (MAS)

In the above table (Table 4), first 3 authors (Figure 9, 10, and 11) have a decent number of collaborators that justify their high productivity. But the authors in the Figure 12 and Figure 14 look different: they have only a few collaborators (degree). So, we grab their co-authorship graphs from MAS (Figure 13 and Figure 15) which says Marco F. Duarte has 88 co-authors and Sven Apen has 141, as opposed to 6 and 4 from our graphs. We understand these are the noises due to the insufficient date in CiteSeer[x] that would affect our algorithm later.

On the other hand, first 3 authors in Table 5 (Aseem Agarwala, Anne Adams, and Alice M. Agogino) have a decent number of collaborators and rich neighborhoods, but fail to overcome our definition of emergence. Again, we believe their limitations come from the insufficient data in CiteSeer[x].

Identifying authors in Table 6 are fairly straightforward: they do not have many collaborators and their colors are faded as well.

Table 5: Social Networks of Non-Emerging Nodes (moderately active authors)

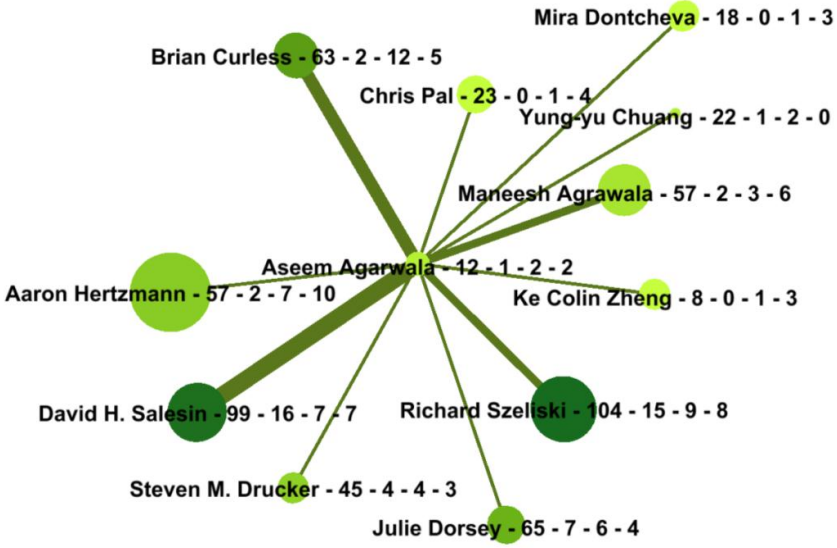

Figure 16: Aseem Agarwala

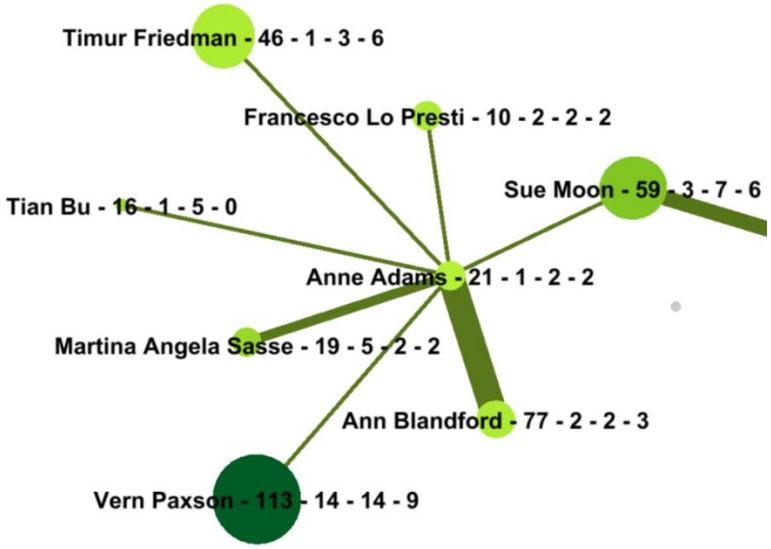

Figure 17: Anne Adams

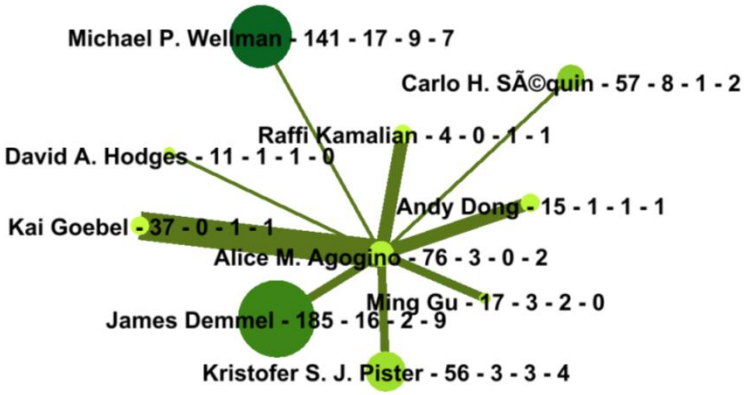

Figure 18: Alice M. Agogino

Figure 19: Andre Adelsbach

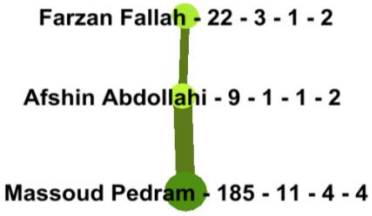

Figure 20: Afshin Abdollahi

Table 6: Social Networks of Non-Emerging Nodes (inactive authors)

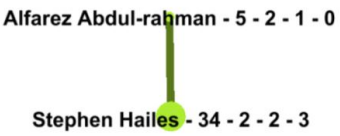

Figure 21: Alfarez Abdul-Rahman

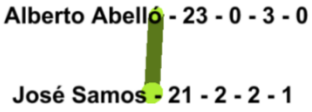

Figure 22: Alberto Abello

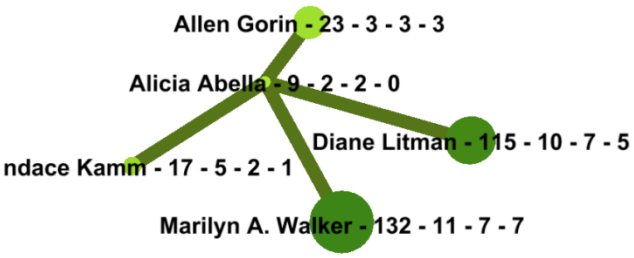

Figure 23: Alicia Abella

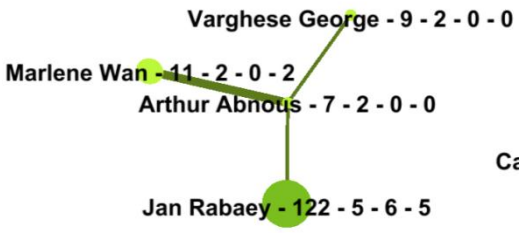

Figure 24: Arthur Abnous

Figure 25: Amund Aarsten

Examining the graphs of the emerging versus non-emerging authors, we can identify social network characteristics associated with the emerging nodes/authors:

(i) They have higher degrees than non-emerging authors (more co-authors).

(ii) Their neighbors are dynamic too (large circles)

(iii) Their neighbors have higher h-indices (dark color).

Besides these social network characteristics, we also assume emerging authors have the following personal features:

(i) Their publications/future publications are going to be 'important' (i.e., have relatively high citations).

(ii) They tend to publish more papers than their non-emerging peers.

Our hypothesis is that, although all low-impact authors may have the same or similar h-index at time *t*, there might be some differences in the number of papers and/or citations that are being overlooked by the h-index now, but may be an important predictor of future success. Therefore, for a researcher/node *n*, we categorize his or her features into two groups: personal features and social features, which are listed in the following Table 7.

Table 7: List of Features

Type	**Features**	**Feature Definition**
Personal features	f_0: $hindex_t(n)$	*the largest **x** for which **n** has **x** papers with at least **x** citations each until time **t (inclusive)***
	f_1: $\Delta hindex_t(n)$	$hindex_t(n) - hindex_{t-\Delta t}(n)$
	f_2: $num\_pubs_t(n)$	*total number of publications of **n** at time t*
	f_3: $num\_citations_t(n)$	*total number of citations of **n** at time t*
Social features	f_4: $degree_t(n)$	$\|Adj_t(n)\|$, where $Adj_t(n)$ is the set of adjacent nodes of *n* in the co-authorship graph at time *t*.
	f_5: $sum\_hindex_t(n)$	$\sum_{m \in Adj_t(n)-n} h\text{-}index_t(m)$
	f_6: $sum\_\Delta hindex_t(n)$	$\sum_{m \in Adj_t(n)-n,\ \Delta t=5} (h\text{-}index_t(m) - h\text{-}index_{t-\Delta t}(m))$

In section 4, we investigate the relationship of each feature to future success and then look at the effectiveness of promising features on classification accuracy.

3.4.3 Dataset Builder

In order to build a reliable test dataset, we needed to create a set of low-impact authors whose future success is known. Thus, we select our low-impact authors for *t = 2005* and we can

generate the truth values of their future success based on their h-index in 2011. On the other hand, to extract the features for these authors as of 2005, we need to gather their data for the years from 2000 to 2005. Therefore, we start with generating a snapshot of the co-authorship graph at *t = 2005*, and calculating the author impacts at *t= 2000, 2005, and 2011*. Then, following the definition of our classes and features, we end up building the training dataset for the classifiers. A fragment of our training set is given in the Table 8. The truth values for each author are shown in the final column in where *E* stands for *Emerging* and *NE* for *Non-Emerging*.

Table 8: Fragment of Training Dataset

Name	Num_pubs (2005)	Num_citatio	h-index (2005)	h-index (2011)	Δh (05)	degree (2005)	sum_Δh	sum_hindex (2005)	Class
Byron Cook	9	430	4	16	3	10	39	89	*E*
Engin Kirda	18	426	4	16	3	14	58	90	*E*
Marco F. Duarte	7	268	2	13	1	6	12	25	*E*
Konstantina Papagiannaki	19	632	4	15	3	20	85	156	*E*
Sven Apel	12	276	2	12	2	4	3	12	*E*
Alice M. Agogino	26	124	3	5	0	9	21	70	*NE*
Afshin Abdollahi	7	51	2	4	1	2	5	19	*NE*
Aseem Agarwala	5	334	3	5	2	11	53	102	*NE*
Andre Adelsbach	12	60	3	5	3	4	6	8	*NE*
Anne Adams	7	126	3	5	2	7	35	63	*NE*
Arthur Abnous	2	39	2	2	0	3	6	15	*NE*
Alberto Abello	8	47	3	3	3	1	2	4	*NE*
Alicia Abella	6	163	4	4	2	4	19	48	*NE*
Alfarez Abdul Rahman	3	414	3	3	1	1	2	4	*NE*
Amund Aarsten	7	27	3	3	2	3	5	9	*NE*

3.5 Classifier Design

We apply two supervised learning algorithms, namely Gaussian Naive Bayes (GNB) and Support Vector Machine (SVM) to see which provides the more accurate emerging author identification results.

3.5.1 Gaussian Naive Bayes (GNB)

Given a class variable y and a feature vector $\vec{x} < x_0, x_1, \ldots, x_n >$, Bayes' theorem assumes features are independent of one another within each class, and provides the following classification rule:

$$\hat{y} = arg\ \max_{y} P(y) \prod_{i=0}^{n} P(x_i|y)$$

where $\hat{y}$ is the predicted class. If we use Maximum A Posteriori (MAP) estimation to estimate class prior $P(y)$, and posterior probabilities $P(x_i|y)$; the former is then the relative frequency of class *y* in the training set. GNB also assumes the likelihood of the features to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\pi\sigma_y^2}\right)$$

where μ_y and σ_y are estimated using maximum likelihood. Rather than implement a Naïve Bayes classifier, we installed and used the python-based machine learning library, scikit-learn [31].

3.5.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) [30] is famous for its good generalization performance and the ability in handling high dimensional data. The SVM tries to find an optimal separating hyperplane to maximally separate two classes of training data. Suppose, $\{(\vec{x}_0, y_0), (\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$ be a two-class linearly separable training dataset, where $\vec{x}_i$ stands for

individual training feature vector and $y_i \in \{-1, +1\}$ for label. Then, computing an SVM corresponds to minimizing $\|\vec{\boldsymbol{w}}\|$ such that

$$y_i(\vec{\boldsymbol{w}}.\vec{\boldsymbol{x}}_i + w_0) - 1 \geq 0, \forall i$$

and the decision function is simply the sign of $\{y_i(\vec{\boldsymbol{w}}.\vec{\boldsymbol{x}}_i + w_0)\}$. Although SVM predicts only class label without probability information, Chang et al [32] shows how to transform SVM decision values into probability values. Again, instead of implementing an SVM classifier, we used the python-based machine learning library, scikit-learn [31].

3.5.3 k-Fold Cross Validation

In *k*-fold cross-validation, the original sample is randomly divided into *k* equal size subsamples. Of the *k* subsamples, a single subsample is set aside as the test data to evaluate the model, and the remaining $k - 1$ subsamples are used as training data. The whole process (training + testing) is then repeated *k* times (the *folds*), with each of the *k* subsamples used exactly once as the test data. The *k* results from the folds are then combined by taking the average to produce a single estimation. The benefit of this method over repeated random sub-sampling is that all observations are used for both training and testing, and each observation is used for testing exactly once. For *k* value, *10* is the most popular choice in the machine learning community. Therefore, we also choose to use 10-fold cross validation in our experiment.

4. EXPERIMENTS

4.1 Test Sets

In the CiteSeer^x database, the numbers of emerging (E) and non-emerging (NE) authors are 1,612 and 50,551 respectively within the time frame from 2000 to 2005. Since the number of emerging authors or the size of *E* is only 3.18% of the size of *NE*, this could skew our classifier accuracy. However, it is clear from this data that the vast majority of low-impact researchers do not, ultimately, go on to make sustained contributions to their field. Thus, we randomly select 1,600 authors from each class (3,200 in totals) to make a balanced test dataset, DS_ALL. Since it takes long time to train the SVM for large dataset, we do not work with DS_ALL; rather we divide the DS_ALL dataset into 8 smaller datasets (DS1 to DS8) each of which contains 400 randomly selected instances of *E* and *NE* (200 from each class). Then, for each of the smaller datasets (DS1 to DS8), we apply 10-fold cross validation to train and evaluate both of our classifiers (SVM and GNB). Finally, these 8 results from the 8 smaller datasets are combined by taking the average. All the accuracies in Table 9 and Table 10 are calculated in this manner. Another intuition of doing this is since we already know that our dataset is noisy, we try to minimize its effect by dividing the whole dataset into smaller chunks, work with them separately, and combine them by taking the average.

4.2 Feature Evaluations

4.2.1 Relative importance of Features

First, we examined the relative importance of individual features as predictors of future success. Therefore, we conducted the classification experiments using only one feature at a time. Table 9 shows the average classification accuracy of SVM and GNB with their statistical significance. The h-index alone (55%) is as good as random guess (50%) at predicting future success. Among all other the features, the 'individual citations count' (f_3) produces the best accuracy (74.3% and 70.2%). With that exception, the social network features such as sum_degree_t, sum_hindex_t, $sum\_\Delta h\text{-}index_t$ are more accurate than the personal features. We also observe that, with a single feature, the Support Vector Machine classifier and the Naïve Bayes classifier perform comparably.

Table 9: Relative Importance of Individual Feature.

Feature	Feature Name	Support Vector Machine (SVM)		Gaussian Naïve Bayes (GNB)		P-value (2-tailed T-test)
		Accuracy	StdDev	Accuracy	StdDev	
f_0	$hindex_t$	0.557	0.026	0.556	0.031	0.966
f_1	$\Delta hindex_t$	0.611	0.02	0.612	0.021	0.928
f_2	num_pubs_t	0.619	0.025	0.604	0.022	0.231
f_3	$num\_citations_t$	**0.743**	0.03	**0.702**	0.042	0.041
f_4	$degree_t$	0.641	0.03	0.613	0.021	0.049

f_5	sum_hindex_t	0.638	0.025	0.608	0.02	0.018
f_6	$sum\_\Delta hindex_t$	0.663	0.023	0.627	0.021	0.005

4.2.2 Combinations of Features

Since more than one feature produces accurate classifications we expect that the combinations of two or more features might work even better. From Table 9, it is obvious that the 55% accuracy of f_0 (h-index) is essentially a random guess (50%). Since it does not contribute anything to the classifier, we omit this feature in our next experiments.

In this set of experiments, we train our classifiers with all possible combinations of 6 features (f_1 to f_6). In Table 10, we display the top performing combinations, grouped by feature size, and highlight the 'local best' within each group in boldface. Table 10 also reveals several interesting findings:

(i) f_3 (citation count) appears most frequently. This is not surprising because it was the most accurate single feature.

(ii) Although f_1 and f_2 provided similar accuracy (61%), f_2 (number of publications) appears less often than f_1 (change of h-index) in the 'local best' combinations.

(iii) As the number of features used by the classifier increases, the f_2 (number of publications) is superseded by the social network feature f_4 (degree centrality), and f_1 is further backed up by f_6 (change of h-index in the neighborhood).

(iv) The Support Vector Machine classifier consistently outperforms the Naïve Bayes classifier in terms of accuracy, by approximately 8.6% on average.

Table 10: Performance Comparison of Different Feature Combinations

Feature Size	Combination of Feature Indices	Support Vector Machine (SVM)		Gaussian Naïve Bayes (GNB)		P-value (Two tailed T-test)
		Accuracy	StdDev	Accuracy	StdDev	
2	**2+3**	**0.745**	0.027	0.676	0.038	0.001
	1+3	0.741	0.033	0.685	0.036	0.006
	3+5	0.74	0.031	**0.688**	0.034	0.006
	3+6	0.739	0.026	0.687	0.037	0.005
	3+4	0.738	0.024	0.687	0.034	0.003
3	**3+5+6**	**0.748**	0.033	0.675	0.034	0.001
	2+3+4	0.743	0.031	0.68	0.033	0.002
	3+4+5	0.741	0.029	0.68	0.038	0.003
	1+3+4	0.74	0.034	0.691	0.033	0.011
	1+3+6	0.738	0.029	**0.694**	0.033	0.014
4	**3+4+5+6**	**0.75**	**0.032**	0.675	0.034	0
	2+3+5+6	0.747	0.032	0.682	0.035	0.002
	1+3+5+6	0.746	0.026	0.686	0.031	0.001
	1+3+4+5	0.741	0.035	0.69	0.028	0.006
	1+3+4+6	0.74	0.023	**0.691**	0.029	0.002
5	**1+3+4+5+6**	**0.75**	**0.028**	0.679	0.031	0
	1+2+3+5+6	0.748	0.031	**0.686**	**0.033**	0.002

	1+2+3+4+6	0.741	0.026	0.685	0.034	0.003
6	1+2+3+4+5+6	0.748	0.032	0.681	0.035	0.001

4.2.3 Accuracy vs. Number of Features

Table 11 summarizes the most accurate results for each feature set size. We observe that as the number of features used to train the classifier increases, the accuracy continues to increase until feature size 5. These gains are surprisingly modest though. Table 11 also reveals that the highest accuracy (75%) is achieved by SVM with both of the combinations $f_3+f_4+f_5+f_6$ and $f_1+f_3+f_4+f_5+f_6$. We consider $f_1+f_3+f_4+f_5+f_6$ the best performer since it has a smaller standard deviation, 0.028 versus 0.032. However, the scenarios are quite different for Naïve Bayes: some of the best combinations in GNB such as $f_1+f_3+f_6$ (accuracy 69.4%) and $f_1+f_3+f_4+f_6$ (accuracy 69.1%) still perform worse than the single feature f_3 (70.2%). We will discuss this issue in section 4.2.4.

Table 11: Best Performing Combinations in each Feature-size Group

Feature size	**Support Vector Machine (SVM)**			**Gaussian Naïve Bayes (GNB)**		
	Accuracy	**StdDev**	**Features**	**Accuracy**	**StdDev**	**Features**
1	0.743	0.03	3	0.702	0.042	3
2	0.745	0.027	2+3	0.688	0.034	3+5
3	0.748	0.033	3+5+6	**0.694**	**0.033**	1+3+6
4	*0.75*	*0.032*	*3+4+5+6*	**0.691**	**0.029**	**1+3+4+6**
5	**0.75**	***0.028***	**1+3+4+5+6**	0.686	0.033	1+2+3+5+6

6	0.748	0.032	1+2+3+4+5+6	0.681	0.035	1+2+3+4+5+6

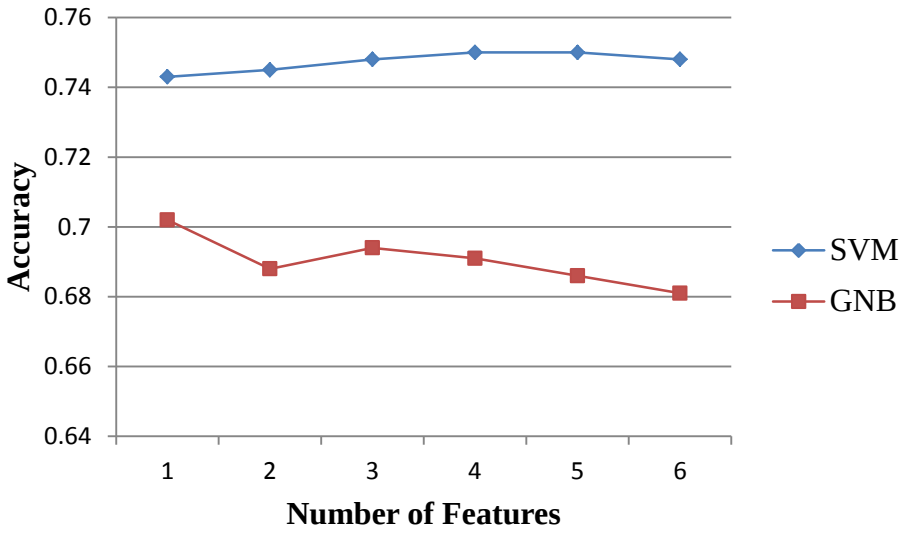

Figure 26: Accuracy vs. Number of Features

4.2.4 Accuracy vs. Training Set Size

Up to this point, all of our experiments have been carried out over the smaller datasets DS1 to DS8. Since we have gained sufficient insights regarding the best performing set of features, we are now interested in measuring the performance of our classifiers as we increase training dataset size. Therefore, we vary the percentage of the gold dataset (DS_ALL) for training from 12.5% to 100% with a step size of 12.5%. Figure 27 indicates the following facts:

(i) A training dataset of size 37.5% (or 1200 authors) of DS_ALL provides the highest accuracy for the SVM classifier.

(ii) Between two competitive feature-sets $f_3+f_4+f_5+f_6$ and $f_1+f_3+f_4+f_5+f_6$, the latter wins in the long run for SVM.

(iii) For GNB, among three competitive features $f_1+f_3+f_6$, $f_1+f_3+f_4+f_6$, and f_3, neither of them is a clear winner.

(iv) The performance of h-index remains unchanged, and it is slightly better than the random guess (the 0.5 line).

(v) The performance of co-authorship graph's degree centrality, SVM (degree) nearly catches the GNB at 37.50% of DS_All dataset point; however, it is 11.69% worse than the SVM (1+3+4+5+6) (the topmost accuracy) at that point.

Finally, we can conclude that the SVM classifier with feature set $f_1+f_3+f_4+f_5+f_6$ and training dataset of size 37.5% of DS_ALL are the optimal settings of our emerging author detection algorithm.

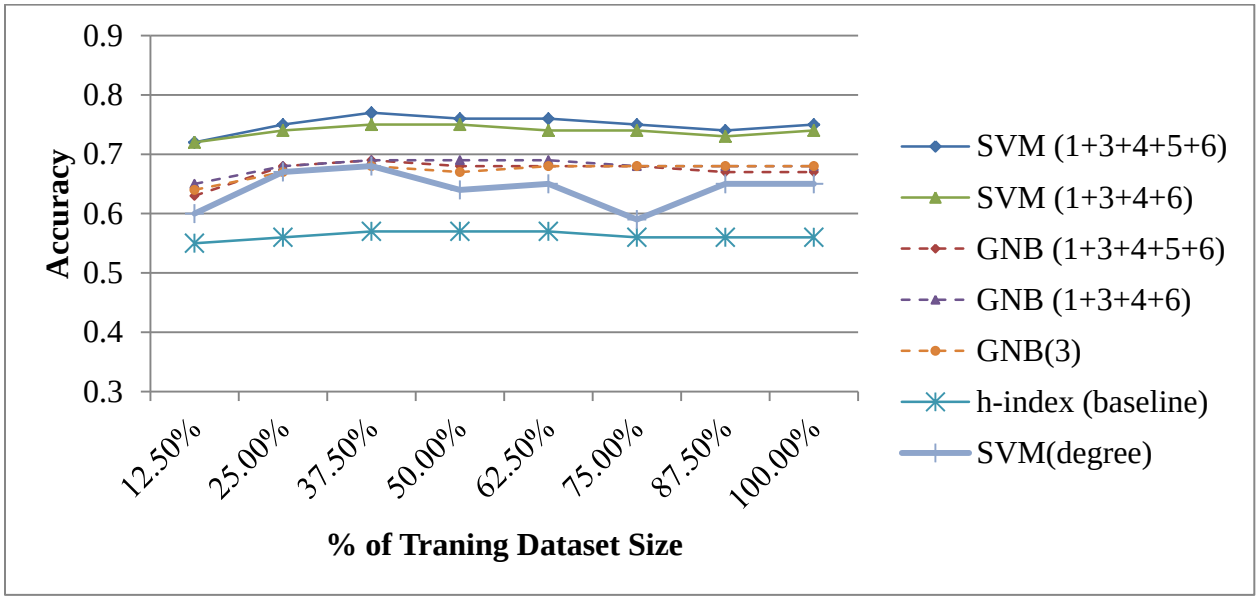

Figure 27: Prediction

4.3 Predicting Emerging Authors

To validate our approach, we end our work by using our algorithm to find emerging authors in the year 2011. We go through the similar process of selecting young researchers described in Section 3.4 where *t = 2011*, and classify them into either of the two classes, namely emerging (E) and non-emerging (NE). Besides class labels, our classifier provides prediction probability. In the following table (Table 12), we enlist a number of authors that the classifier believes to have high ($probability \geq 0.90$), moderate ($0.7 \leq probability < 0.9$), and low ($0.5 \leq probability < 0.7$) chances of being rising stars. We also include their most recent h-indices (@2013) from CiteSeer[x] as an indicator of their true achievements.

Table 12: Predicting Emerging Authors

Author's Name	**Affiliation**	**h-index (2011)**	**h-index (2013)**	**Prediction probability**
L. Grimson	MIT	4	8	0.997732
Stefan Naher	University of Trier	4	7	0.999406
Martn Abadi	Microsoft	4	5	0.998755
Robert Bridson	University of British Columbia	4	5	0.958197
Val Breazu-Tannen	University of Pennsylvania	4	5	0.945338
David L. Applegate	AT&T Labs Research	3	6	0.727797
Bruce L. Worthington	Microsoft	4	6	0.839929
Markus M. Breunig	German Aerospace Center	3	5	0.815587
Paul I. Dantzig	IBM	4	5	0.749348

Frederico Torres Fonseca	Universidad Nacional de Colombia	4	5	0.748446
Martin R. Andersson	Chalmers University of Technology	4	7	0.698364
Thomas Gartner	University of Bonn	4	6	0.683219
Valentin Jijkoun	University of Amsterdam	4	6	0.618103
Saul London		3	6	0.646772
Antonina Starita	University of Pisa	4	6	0.50737

The following table, Table 13, contains 15 randomly selected authors who are identified as non-emerging according to our algorithm.

Table 13: Prediction of Non-Emerging Authors

Author's Name	**Affiliation**	**h-index (2011)**	**h-index (2013)**	**Prediction probability**
Brendan Mccane	University of Otago	3	3	0.357155
Youssef Iraqi	Khalifa University of Science	3	3	0.345736
Brian K. Grant	University of Washington	3	3	0.468394
Pascal Gautron	Institut de Recherche en Informatique	3	3	0.323327
Torsten Schlieder	Free University of Berlin	3	5	0.327689
Lee W. Campbell	Massachusetts Institute of Technology	2	4	0.343336
Uffe Kjaerulff	Aalborg University	2	4	0.332213
Hugues Marchand	Universit	2	4	0.314621

Philippe Morignot	Atomic Energy Commission	2	4	0.302967
Jean-Baptiste Pomet	The French National Institute	2	4	0.304718
Caixue Lin	University of Tennessee Knoxville	1	4	0.410267
Edward Bortnikov	Yahoo Research Labs	1	2	0.41243
Michael Baentsch	IBM	1	2	0.430239
Osman Balci	Selcuk University	1	2	0.347947
Richard Cavanaugh	University of Florida	1	2	0.319026

When we compare the two groups after just two years, the emerging authors have increased their average h-index from 3.8 to 5.87 whereas the predicted non-emerging authors have only increased their average h-index from 2 to 3.26.

4.4 Discussion

4.4.1 Why Citation Count Works so Well

Historically, citation count plays a big role in measuring the total impact of a researcher. However, one of its disadvantages is that it may be inflated by a small number of ’big hits’ that may not be the actual contribution of the individual if he or she is co-author with many others on a one or a few highly cited papers. To overcome this drawback, many researchers suggested using a variant of citation count: the number of citations to each of the q most cited papers, (for example, q = 5) [1]. Since the young researchers do not have that many publications, their total citation count is equivalent to this measurement. This might be a reason why citation count works so well.

Another explanation is citation count provides a wider margin to the classifier. As Hirsch mentioned, if an author's h-index is h, then her total citation counts would be $a * h^2$, where $a = 3 \sim 5$ [1]. Therefore, unlike any other features, citation count is directly proportional to the square of individual's h-index which makes it less vulnerable to noise.

4.4.2 Dataset in Retrospect

In Section 3.2.1, we mentioned that we built the graph of 62,886 nodes from the set of 278,904 papers. However, there were 199,628 authors in the paper-set, and our algorithm (in Table 1, Line 5.ii. b.) discarded almost 68.5% of the total authors because they did not belong to our disambiguated author-set of size 62,886. Moreover, by taking random samples and inspecting them visually, we have found that an individual author has 2.26 duplicate entries on average in CiteSeer^x. Therefore, the numbers of publications and the number of citations of an author are distributed among his or her duplicate entities. Among these multiple entities of an author, we choose to use the entity that has the highest number of publications. As a result, it is obvious that we have lost a lot of useful information that might have affected the accuracy of our results.

4.4.3 Usefulness of pure Co-authorship Graph

Although we have shown that the best performance comes from the combination of personal and social features, some of our social and personal features are difficult to calculate. For example, collecting citation data chronologically is very difficult as these are often copyrighted by digital libraries. Without proper citation data, the calculation of the h-index would not be possible. Therefore, some of our features such as f_1, f_5, and f_6 would be unavailable

as well. On the other hand, co-authorship data are freely available online[12]. So, one could easily build fairly complete and large-scale co-authorship network. Our experimental data shows that we can achieve accuracy as high as 68% using just degree centrality (Figure 27). Although it is 11.69% worse than the best performing counterpart, we expect to have better accuracy with less noisy data. Therefore, we can consider degree centrality as a cheap, alternative single-valued feature for the classifier.

[12] http://dblp.uni-trier.de/xml/

5. CONCLUSIONS

5.1 Summary

In this study, we empirically classify young researchers into two classes, namely emerging and non-emerging, depending on their h-indices. Then, we investigate which are the key characteristics of emerging authors based on personal and social features. We concluded that the success of a young individual researcher largely depends on his or her early citations, number of collaborators, and the impact and recent research activity of the collaborators.

We built a social network of 62,886 authors using the data available in CiteSeer^{x}. To view these social networks online, we developed an interactive, web-based user interface. Moreover, we also offer web services so that anyone can work with these graphs by their own way.

We then designed and trained SVN and Naïve Bayes classifiers to learn how to identify emerging authors based on the personal and social aspects of a set of 3,200 young researchers who had an h-index of less than or equal to four in 2005. We represented each of these researchers as a six-dimensional vector of features. Since we already knew that there was noise in our data, we divided our original dataset into 8 smaller datasets averaged the results. It is noteworthy to mention to that we trained both classifiers on all possible combinations of features (a total of 63 sets of features) to determine which combination(s) worked best. We found that SVM classifier with the feature set <individual's change of h-index, citation count, degree centrality, total h-indices of the neighbors, and total change of h-indices of the neighbors> worked best, providing an accuracy of 75% when predicting emerging authors as of 2011, 5 years later.

After we completed our experiments with our test and training data set, the best performing classifier was used to make the prediction of producing research impacts in the coming years of a set of 50,551 researchers who had an h-index of less than or equal four in 2011. Finally, when we examined the results, we found that after just two years (in 2013), the predicted emerging researchers had increased their average h-index from 3.8 to 5.87 whereas the predicted non-emerging ones had only increased their average h-index from 2 to 3.26 (from the data available in CiteSeer[x]).

5.2 Contributions

Throughout this study, we made the following contributions:

- We find a combination of personal and social features that allows us to predict future success for young researchers.
- We offer a new visual browsing interface for CiteSeer[x].
- We propose that with the lack of citation data, degree centrality could be an alternative single-feature for training classifiers.
- We found that, although the h-index is a poor estimator of the potentials of young researchers, citation count is a strong candidate.

5.3 Future Work

While this work provides the basic framework for finding emerging authors, there is still plenty of room for improvement. For example, we extract social features of a node from its immediate neighbors (1-level deep) only. It would be an interesting study to see the effect of extracting features from nodes at distance two or more, making use of more of an author's academic social network. Moreover, our co-authorship graph is weighted, but we do not

incorporate edge-weights in this study. Furthermore, we can vary the threshold value of h-index in the definition of our Emerging/Non-Emerging classes and re-do the experiment. Finally, we are excited to see the results of our algorithm on a different, clean dataset.

6. REFERENCE

1. J. E. Hirsch, "An Index to Quantify an Individual's Scientific Research Output", *PNAS*, 102 (46), 2005, 16569–16572.

2. R. Guns, R. Rousseau, "Simulating Growth of the H-index", *JASIST*, 60 (2), 2009, pp. 410-417.

3. D. Lindsey, "Using Citation Counts as a Measure of Quality in Science Measuring What's Measurable Rather Than What's Valid", *Scientometrics*, 15 (3), 1989, pp. 189–203.

4. P. Lawrence, "The Mismeasurement of Science", *Current Biology*, 17 (15), 2007, R583.

5. M. Moravcsik and P. Murugesan, "Some Results on the Function and Quality of Citations", *Social Studies of Science*, 5 (1), 1975, pp. 86.

6. S. Wasserman, and K. Faust, "Social Network Analysis: Methods and Applications", Cambridge University Press, 1994.

7. E. Otte, and R. Rousseau, "Social Network Analysis: a Powerful Strategy, also for the Information Sciences", *Journal of Information Science*, 28 (6), 2002, pp. 441–453.

8. D. Watts, "Small Worlds: The Dynamics of Networks between Order and Randomness", Princeton University Press, 2001.

9. J. Scott, "Social Network Analysis: A Handbook", 2nd ed., Sage Publications, London, 2000.

10. I. Farkas, I. Derenyi, H. Jeong, Z. Neda, Z. N. Oltvai, E. Ravasz, A. Schubert, A.-L. Barabasi, and T. Vicsek, "Networks in life: Scaling Properties and Eigenvalue Spectra", *Physica A*, 314 (1-4), 2002, pp. 25-34.

11. P. Ball, "Index Aims for Fair Ranking of Scientists", *Nature*, 436 (7053), 2005, pp. 900.

12. S. B. Popov, "A Parameter to Quantify Dynamics of a Researcher's Scientific Activity", ArXiv:physics/0508113, 2005, accessible via http://arxiv.org/abs/physics/0508113.

13. P.D. Batista, M.G. Campiteli, O. Kinouchi, and A.S. Martinez, "Is it Possible to Compare Researchers with Different Scientific Interests?" ArXiv:physics/0509048, 2005, accessible via http://arxiv.org/abs/physics/0509048.

14. L. Bornmann, and H.-D. Daniel, "Does the H-index for Ranking of Scientists Really Work?" *Scientometrics*, 65 (3), 2005, pp. 391-392.

15. M. T. Irfan, L. E. Ortiz, "On Influence, Stable Behavior, and the Most Influential Individuals in Networks: A Game-Theoretic Approach", CoRR, 2013, accessible via http://arxiv.org/abs/1303.2147.

16. S. Adali, X. Lu, M. Ismail, and J. T. Purnell, "Prominence Ranking in Graphs with Community Structure", *ICWSM*, 2011.

17. N. A. Christakis, J. H. Fowler, "The Spread of Obesity in a Large Social Network Over 32 Years", *N. Engl. J. Med*., 357, 2007, pp. 370–379.

18. E. Garfield, "Citation Indexing-Its Theory and Application in Science, Technology, and Humanities", John Wiley & Sons, New York, NY, 1979.

19. M. E. J. Newman, "Scientific Collaboration Networks: I. Network Construction and Fundamental Results", *Physical Review E.*, 64:016131, 2001.

20. M. E. J. Newman, "Scientific Collaboration Networks: II. Shortest Paths, Weighted Networks, and Centrality", *Physical Review E.*, 64:016132, 2001.

21. A. F. Smeaton, G. Keogh, C. Gurrin, K. McDonald, and T. Sodring, "Analysis of Papers from Twenty-Five Years of SIGIR conferences: What have we been Doing for the Last Quarter of a Century", *SIGIR*, 36 (2), 2002.

22. M. A. Nascimento, J. Sander, and J. Pound, "Analysis of SIGMOD's Co-authorship Graph", *SIGMOD*, 32 (3), 2003.

23. S. He, and A. Spink, "A comparison of Foreign Authorship Distribution in JASIST and the Journal of Documentation", *Journal of the American Society for Information Science and Technology*, 53 (11), 2002, pp. 953–959.

24. C. Wetherell, A. Plakans, and B. Wellman, "Social Networks, Kinship, and Community in Eastern Europe", *Journal of Interdisciplinary History*, 24 (4), 1994, pp. 639–663.

25. R. D. Castro, and J. Grossman, "Famous trails to Paul Erdös. MATHINT: The Mathematical Intelligencer", 21, 1999, pp. 51–63.

26. X. Liu, J. Bollen, M. L. Nelson, and H. V. Sompel "Co-Authorship Networks in the Digital Library Research Community", Information Processing and Management, 41 (6), 2005, pp. 1462-1480.

27. C. L. Giles, K. D. Bollacker, and S. Lawrence, "CiteSeer^{x}: An Automatic Citation Indexing System", *Proceedings of the 3rd ACM conference on Digital libraries*, New York, NY, 1998, pp. 89–98.

28. Microsoft Academic Search, Retrieved on July 21, 2013, from http://academic.research.microsoft.com/.

29. Neo4j - The World's Leading Graph Database, Retrieved July 21, 2013, from http://neo4j.com.

30. V. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, 1995.

31. Pedregosa et al., "Scikit-learn: Machine Learning in Python", *JMLR*, 12, 2011, pp. 2825-2830.

32. C. Chang, and C Lin, "LIBSVM: A Library for Support Vector Machines", ACM Transactions on Intelligent Systems and Technology, 2 (3), 2011, pp. 27:1-27:27.

33. P. De, A. E. Singh, T. Wong, W. Yacoub, A. M. Jolly, "Sexual Network Analysis of a Gonorrhoea Outbreak", *Sex Transm Infect*, 80(4), 2004, pp. 280-285.

34. Valente, W. Thomas, "Network Models of the Diffusion of Innovations", Cresskill, NJ: Hampton Press, 1995.

35. Shishkin, Philip, "Genes and the Friends You Make", Retrieved on January 27, 2009, from http://online.wsj.com/article/SB123302040874118079.html.

36. M. Granovetter, "Introduction for the French Reader", *Sociologica*, 2, 2007, pp. 1–8.

37. J. H. Fowler, N. A. Christakis, "Dynamic Spread of Happiness in a Large Social Network: Longitudinal Analysis over 20 years in the Framingham Heart Study", *BMJ*, 337, 2008, a2338.

38. L. Backstrom, P. Boldi, M. Rosa, J. Ugander, S. Vigna, "Four degree of separation", arXiv:1111.4570, 2011, accessible via http://arxiv.org/abs/1111.4570.

39. M. M. Skeels, and J. Grudin, "When Social Networks Cross Boundaries: A Case Study of Workplace Use of Facebook and LinkedIn", *Proceedings of the ACM 2009 international conference on Supporting group work*, 2009, pp. 95-104.

40. H. Luong, T. Huynh, S. Gauch, and K. Hoang, "Exploiting Social Networks for Publication Venue Recommendations", *ACIIDS*, 3, 2012, pp. 426-435.

41. S. Brin, and L. Page, "The Anatomy of a Large-scale Hypertextual Web Search Engine", *Proceedings of the 7th International World Wide web Conference*, 1998.

42. J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment", *Journal of ACM (JASM)*, 46 (5), 1999, pp. 604-632.

43. M. Bastian et al., "Gephi: An Open Source Software for Exploring and Manipulating Networks", *International AAAI Conference on Weblogs and Social Media*, USA, 2009.

44. Y. F. Hu, "Efficient and High Quality Force-Directed Graph Drawing", *The Mathematica Journal*, 10, 2005, pp. 37-71.