A Recommendation System Based on Hierarchical Clustering of an Article-Level Citation Network

Jevin D. West, Ian Wesley-Smith, and Carl T. Bergstrom

Abstract—The scholarly literature is expanding at a rate that necessitates intelligent algorithms for search and navigation.For the most part, the problem of delivering scholarly articles has been solved. If one knows the title of an article, locating it requires little effort and, paywalls permitting, acquiring a digital copy has become trivial. However, the navigational aspect of scientific search – finding relevant, influential articles that one does not know exist – is in its early development. In this paper, we introduce *Eigenfactor Recommends* – a citation-based method for improving scholarly navigation. The algorithm uses the hierarchical structure of scientific knowledge, making possible *multiple scales of relevance* for different users. We implement the method and generate more than 300 million recommendations from more than 35 million articles from various bibliographic databases including the AMiner dataset. We find little overlap with co-citation, another well-known citation recommender, which indicates potential complementarity. In an online A-B comparison using SSRN, we find that our approach performs as well as co-citation, but this new approach offers much larger recommendation coverage. We make the code and recommendations freely available at babel.eigenfactor.organd provide an API for others to use for implementing and comparing the recommendations on their own platforms.

Index Terms—scholarly recommendation, citation networks, hierarchical clustering, community detection, big scholarly data

1 Introduction

Back in the 20th century, when libraries were predominantly physical institutions, the process of document acquisition facilitated serendipitous discoveries of related work and the formation of unexpected intellectual connections. To read a journal article, a scholar had to physically acquire the issue in which it was printed; in the process she would often stumble fortuitously across other relevant articles in the same issue. To read a book, a researcher would first have to *navigate* the library stacks and scan across dozens of titles-often finding even more relevant volumes in the process.

No longer. Today, digital document delivery approaches a single-click level of ease. A scholar types a few words of a title, or the surnames of a few authors into a search engine, portal, or document repository, and can then proceed immediately to the required document without any of the search and browsing time that we routinely had to invest only ten or fifteen years ago. For the most part, this represents an enormous increase in efficiency, but something has been lost as well: readers are no longer exposed to related material as part of the document acquisition process. Fortunately, this loss is not a necessary consequence of the digital transition in scholarship. We can deliberately engineer tools to replicate the benefits of these previous serendipitous processes, but with hugely greater time efficiency and perhaps with better-than-chance ability to direct researchers to relevant studies.

Balancing serendipity with relevancy, however, is challenging given the exponential growth of scientific knowledge. The volume of scholarly literature roughly doubles every decade [1], [2]. More than a million articles are added to this corpus on an annual basis [3]. We are long past the days in which a scholar could keep up with the literature simply by reading through each issue of each journal in her field as it was published. Exhaustive search no longer scales, and we face a desperate need for informetric tools that facilitate document discovery. Of course, tools for targeted search-designed to help researchers find needed papers when they know approximately what they are looking for-will play an important role, as will systems for more efficiently navigating the topography of academic scholarship fro papers the researcher does not know exist. But there is a role of chance discovery as well, the sort of chance discovery that used to be a regular miracle among the stacks of our university libraries. This is our aim in the present paper: to develop a system that serves both purposes, leading readers to the key literature in the areas of their interest, while also providing suggestions that may be as valuable as they are non-obvious.

We propose a recommender with the following set of objectives. The recommendation system (based on a seed paper) needs to determine (1) what papers are relevant to the seed paper and (2) of these, which are the most important (3) for different user types (e.g., novice versus experienced). We propose a new citation-based approach to this problem called Eigenfactor Recommends (EFrec) that meets these three objectives. We couple the hierarchical structure of the citation network – which reflects the natural hierarchical structure of scientific domains, fields, subfields, and so forth – with

- J.D. West and I. Wesley-Smith are with the Information School, University of Washington, Seattle, WA 98195. E-mail: {jevinw, iwsmith}@uw.edu.
- C.T. Bergstrom is with the Department of Biology, University of Washington, Seattle, WA 98195. E-mail: cbergst@uw.edu.

Manuscript received 1 Aug. 2015; accepted 12 Feb. 2016; revised 5 Jan. 2016. Date of publication 24 Mar. 2016; date of current version 29 July 2016. Recommended for acceptance by Y.-R. Lin, H. Tong, J. Tang, and K. S. Candan. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TBDATA.2016.2541167

TABLE 1
Babel Datasets

Dataset	Papers	Citations	Recommendations
AMiner	2,092,356	8,024,869	22,112,496
JSTOR	1,787,351	8,227,537	14,813,224
PLOS	1,599,712	3,232,766	8,647,037
PubMed	5,538,322	16,004,596	34,026,854
arXiv	626,441	781,108	5,624,262
DBLP	781,108	4,191,677	2,163,313
MAS	27,352,532	262,554,975	245,796,494

Recommendations were generated for the following datasets and available at babel.eigenfactor.org.

importance scoring based upon a network centrality measure. In this way we use hierarchical clustering to determine relevance and then recommend papers based upon their importance within these clusters. Thus, we are able to generate a spectrum (or scale) of recommendations for any given topic, paper, or set of key words. We can find papers that are very closely related but perhaps not yet very influential (*Expert Recommendations*). Alternatively, we can find papers that may be more distantly related but represent foundational contributions to the broader area of research (*Classic Recommendations*) for researchers new to a field. We find that this approach provides recommendations for a much larger set of papers than one can provide using a co-citation approach, and performs at least as well in A-B testing. It also, distinctly, provides recommendations at different scales for different user types.

We apply EFrec to the AMiner data set [4] included in this special issue on 'Big Scholar Data Discovery'. To supplement this analysis, we also generate more than 300 million recommendations for various other bibliographic datasets including the arXiv, JSTOR, Microsoft Academic Search (MAS), PLoS, Social Science Research Network (SSRN) and Pubmed Central (Table 1) and make the code and data available through a public API.

1.1 Previous Work

The Eigenfactor Recommends algorithm is not the first scholarly recommender. A number of document collections and portals are already deploying basic recommendation designs that aim toward this goal. For example, when one views an abstract on Pubmed, the system suggests five articles as "related citations" in the sidebar. Elsevier's ScienceDirect system, Highwire press, and many other content systems present a similar sidebar highlighting related articles. In this section, we explain how EFrec fits within the broad category of scholarly recommender systems.

First, there are collaborative filtering approaches, which rely on finding similar users and using their ratings to provide recommendations. These techniques are extremely powerful and widely explored in the literature [5]; they require no knowledge about the items being recommended, and with sufficient user ratings can infer properties about the items in question [6]. They do, however, have several limitations that make them difficult to use for scholarly article recommendation [7]. First and foremost is the "cold-start" problem – items that don't have user ratings cannot be recommended [8], which is a substantial portion of the literature. Another limitation is that to be effective you need good rating coverage; the number of ratings should dominate the number of items being rated to perform well. In domains with many items and few user ratings, such as scholarly literature, this condition can be difficult to satisfy. Perhaps the largest problem, though, is the difficulty in acquiring user ratings. To collect a sufficient amount of data you need substantial traffic and the infrastructure to serve and gather ratings from a large number of users. Unless one is a big publisher, these resources are difficult to obtain and since publishers rarely share data, it is difficult to develop methods using usage data.

At the other end of the recommender spectrum are content based methods, which instead match items to similar items based on features [7]. Content based methods are ideal for datasets where there are few users, however there is substantial difficulty in automated feature extraction. Term frequency-inverse document frequency (TF-IDF) is a commonly used technique [7]. However, this approach has significant downsides, especially in scientific literature where synonymy is an issue [9], [10]. And, like usage data, full text is difficult to obtain from publishers.

When domain-specific features are available, such as links in web pages or citations in academic literature, they can be used to great effect. The approach of using links between documents is often called link analysis. The first instance of it was in the early 1960s, when Kessler invented bibliographic coupling [11]. Bibliographic coupling measured document similarity by the number of shared citations. In his paper, Kessler describes some of the strengths of citation-based approaches over textual analysis, noting that they are language-independent and do not demand the recommender to have any expertise in the subject matter.

The next stage in the evolution of citation analysis came in the early 1970s, when Small introduced co-citation analysis [12], a method that uses the frequency with which papers are cited together as a measure of similarity. Small address several of the shortcomings of bibliographic coupling, including the permanence of bibliographic coupling, and how co-citation is able to change as intellectual patterns change over time. Co-citation analysis was later applied to authors by White and Griffith [13] and has been used extensively as a standard citation-based method for recommending related papers.

The 1990s brought a flurry of papers on link analysis and represents the next epoch of techniques. This advancement was not motivated by the aim of providing better tools for scholars, but rather by the need to navigate the new, massive corpus of linked documents: the world wide web. The two most notably entries are HITS [14] by Kleinberg and PageRank [15] by Brin and Page. These methods sought to exploit the links present in web pages to provide a notion of authority or importance.[1] Both also utilized similar techniques of a random walker (or surfer), following links at random around the network. By tracking where this walker goes we can, based on the frequency of node visits, determine which nodes are the most important.

1. There is a common misconception that PageRank is a search algorithm. In actuality, the function of PageRank is to rank the nodes in a network by importance. These rankings can then be used to determine the order in which search results are presented. Variants of PageRank have been applied to scholarly networks to determine the impact of journals, authors and articles [16], [17], [18], [19], [20].

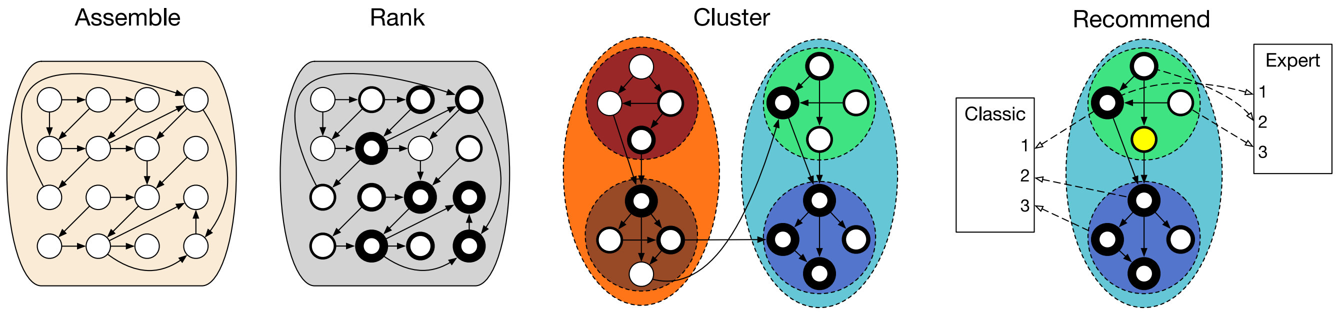

Fig. 1. *Eigenfactor Recommends*. The process for generating paper recommendations includes the following steps. (1) In the first step, the paper citation graph is assembled into an adjacency matrix using the citations (links) between papers (nodes). (2) The second step ranks the nodes using the article-level Eigenfactor algorithm (ALEF). This is a modified, time-directed version of the PageRank algorithm [15]. (3) The third step imports the citation graph and node rankings and then clusters the citation graph using the hierarchical MapEquation [23]. (4) The last step generates recommendations given a seed paper (highlighted in yellow). *Expert* recommendations are drawn from the lowest level of the hierarchical tree, while *Classic* recommendations include papers from one level up the hierarchy. A paper can be both an expert recommendation and a classic recommendation if it is the highest rated paper at the lowest and upper levels.

To take PageRank (or any impact measure) and turn it into a search algorithm you need to partition the network intelligently. Brin et al. did this by using standard information retrieval (IR) techniques, augmenting them slightly for hyperlinks. They then take this IR score and combine it with the PageRank to generate a final score for a given document. Kleinberg did this by collecting a "root set" of the top results from text-based search engines (specifically AltaVista and HotBot). HITS then matches a query to one of these partitions, and then generates impact scores.[2] Haveliwala[21] modified PageRank and created a "topic-sensitive" variant, where, much like HITS, an external source is used to generate a partition and then an impact score is generated for that network, resulting in topic-specific PageRank scores. A variant of this idea was used by PaperRank [22], a recommender for scholarly literature. In PaperRank's case they used the top-n results from the ACM digital library, and crawled all citations (up to 2,000 papers) to form a topic specific network.

What makes EFrec unique among these link analysis techniques is that it can both determine an impact score for papers in a network and partition the network using only the citation graph. We do this by exploiting domain-specific properties of scholarly literature: that scientific literature is hierarchically structured into domains, fields, subfields, sub-subfields, etc. If we can partition the network according to this hierarchy, we can then determine the impact within a specific area and provide more accurate recommendations. And, most importantly, EFrec can provide 'levels of relevance', which can be important when no information is provided about a user needs. We explain this further in the following Methods section.

2 Methods

The objective of the EFrec algorithm is to find relevant papers, given a seed paper.[3] The data required to generate these recommendations is a citation graph. The output consists of a list of paper IDs and a set of recommended papers associated with each paper ID. The number of recommendations to be obtained can be set by the user, and range from one to hundreds of recommendations. Fig. 1 illustrates the four-step process for generating paper recommendations.

2.1 Assemble Citation Network

The first step requires assembling the citation graph of a relatively large corpus, where "relavitely large" means at least a few hundred thousands papers and a few million citations – bigger the better. This method does not perform as well for small graphs. The graph is represented as a link list file where column 1 is the citing paper ID and column 2 is the cited paper ID. For the AMiner dataset, the network consisted of 2,092,356 nodes (papers) and 8,024,869 links (citations). Because each paper cites only a small number of other references, the networks underlying large corpora will be highly sparse, as described in the section Sparseness of Citation Graphs [3.3].

2.2 Rank Nodes

The second step consists of ranking the nodes. We rank the nodes according to the article-level Eigenfactor (ALEF[4]) [25]. The original Eigenfactor algorithm, which is closely related to original PageRank algorithm [15], performs well on journal-level citation graphs because these are cyclic, in that one can follow directed links (citations) from one journal out to other journals and then back to the originating journal. At the article level, however, citation graphs are acyclic or nearly so: a paper only cite those articles that preceded it in time. When standard PageRank approaches are applied to acycling citation graphs, older papers are excessively weighted [26], [27].

ALEF is a modified version of PageRank (PR), tailored specifically for the time-directed acyclic networks associated with article-level citations. We show in a recent study that ALEF performs better than PageRank and degree centrality [28] and have also demonstrated that ALEF better separates papers that contribute to a given theory [29].

2. This is a bit of a simplification of both steps. HITS grows out from the initial root set, and does not provide a single score but rather a hub score and an authority score.

3. Instead of a paper, the input could be an author's name. The recommendations would then be related authors. We have developed ranking methods at the author level to the author-disambiguated SSRN dataset [17]. Because of the author disambiguation issues with the other datasets we have not deployed the author-level version of EFrec as of yet.

4. These scores are used to order the top articles in ref. [24]. A manuscript describing the method in detail is in preparation and will be posted on the arXiv. This paper and details of the method can be provided upon request.

As mentioned above, the problem with using a PageRank approach on article-level citation graphis is that the "random walker" on the graph will move inexorably backwards in time, and as a result will over-weight older papers. ALEF addresses these issues with the following two modifications: (1) it shortens the number of contiguous steps of the random walker before she teleports to another part of the network and (2) it teleports to links rather than nodes [30]. These modifications help adjust for the over-weighting without losing the ability to exploit network structure.

The mechanics of ALEF are relatively simple and proceed in five steps. First, the teleportation weight, w_i for each node i is calculated by summing the in- and out-citations. Second, the random walker teleports to a random node based on the teleportation weights. Third, the random walker takes a step along the citation graph. Fourth, we compute the asymptotic rate at which each node is visited under this process. And fourth, these rates—which provide our rankings—are normalized so that the average score for all papers equals 1.

Specifically, the article citation network can be represented as an $n \times n$ adjacency matrix, $\mathbf{Z}$, where the Z_{ij} entry equals 1 if article i cites article j and 0 otherwise. The matrix is highly sparse since an individual article cites a tiny portion of all articles in the corpus. The teleportation vector, indicating the rate of teleportion to each node i, is calculated in the following way:

$$w_i = \sum_j^n (Z_{ij} + Z_{ij}^T).$$

The matrix $\mathbf{Z}_{ij}$ is then row normalized so that the sum of each row i equals 1. We call this row stochastic matrix, $\mathbf{H}_{ij}$

$$\mathbf{H}_{ij} = \frac{\mathbf{Z}_{ij}}{\mathbf{Z_i}}.$$

The ALEF scores are then calculated by multiplying w_i by $\mathbf{H}_{ij}$ and normalizing the scores by the number of papers, n, in the corpus

$$\mathrm{ALEF} = n\frac{\mathbf{H}_{ij}^T . w_i}{\sum_i [\mathbf{H}_{ij}^T . w_i]_i}.$$

We have chosen the current ranking strategy as a balance between a fully degree centrality ranking (i.e., counting links) and the original PageRank algorithm.[5]

2.3 Hierarchically Cluster Nodes

The third step is to cluster the network, using the citation graph from step 1 and node rankings from step 2. Based on these inputs, the hierarchical MapEquation [23] uncovers the boundaries between domains, fields, subfields, etc.

For simplicity, we begin with a description of the non-hierarchical map equation [31], [32], which uncovers basic modular structure within networks, returning a hard partition in which each node is assigned to a single module. The map equation exploits the duality between compressing data and revealing patterns within the data. The core idea is to compress a description of a random walk on the network. If the network has localized regions such that a random walker has a long persistence time in a small group of nodes, a random walk can be concisely encoded when this structure is exploited in the coding scheme.

Optimally compressed using a two-level description (nodes and modules), the per-step description length M of a random walk on a network is given by the map equation:

$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p_{\circlearrowright}^{i} H(\mathcal{P}^i).$$

The term $H(\mathcal{Q})$ represents the description length necessary to transmit the name of the module in which the random walker resides, and this is weighted by $q_{\curvearrowright}$, the frequency of movement between modules. The term $H(\mathcal{P}^i)$ represents the description length necessary to transmit the node within module i to which the walker has moved. This term is weighted by the frequency $p_{\circlearrowright}^{i}$ with which the random walker moves within module i. Shannon's source coding theorem [33] states that the minimum code length necessary to describe a random variable X is given by its entropy: $H(X) = -\sum_1^n p_i \log(p_i)$. Using this fact, for any given partition of the network each of the terms can be calculated in straightforward fashion from the citation matrix. Details are provided in refs. [31], [32]. By searching numerically across the space of possible network partitions for the one that allows the shortest description length, we can find an optimal partition of the network. Benchmark studies have revealed that this approach performs extremely well relative to other network clustering algorithms [34], [35].

The hierarchical map equation extends this basic approach to reveal modular structure on multiple levels. This is done by extending the coding scheme of the basic map equation to a hierarchical one. The reader is referred to ref. [23] for details. Performance of this algorithm is extremely good [36] and it readily scales to networks with tens of millions of links and hundreds of millions of nodes [37]. The software for running this part of the algorithm[6] can be downloaded at mapequation.org.

2.4 Recommendation Selection

The last step involves selecting nodes for three types of recommendations: (1) *Expert*, (2) *Classic* and (3) *Serendipity*. Each of these types use no specific information about the user.[7] However, the different recommendation types offer results that will be most useful to different kinds of users. The *Expert* recommendations are aimed at researchers familiar with a community of papers and authors. The algorithm aims to select papers that are highly specific to a particular sub-discipline. The *Classic* recommendations, on the

5. It should be noted that the basic logic of the Eigenfactor Recommends algorithm could have used a different ranking method, but recent data challenges have shown that ALEF performs better for identifying important papers [28].

6. When using InfoMap, select the "-t" option for undirdir, or undirected, directed. Undirdir is the approach underlying the ALEF ranking.

7. Another version of the algorithm could utilize usage characteristics like a reader's bibtex file, reading list or viewing behavior. We are working with researchers at SSRN to build a personalized version of EFrec.

TABLE 2
The Top 10 *Expert* Recommendations for the Paper, "Authoritative Sources in a Hyperlinked Environment" [14]

ALEF	Title	Year	Venue
790	The anatomy of a large-scale hypertextual Web search engine	1998	WWW
255	Authoritative sources in a hyperlinked environment	1998	ACM-SIAM symposium
125	Topic-sensitive PageRank	2002	WWW
121	Improved algorithms for topic distillation in a hyperlinked environ.	1998	ACM SIGIR
111	Trawling the Web for emerging cyber-communities	1999	WWW
110	Automatic resource compilation by analyzing hyperlink structure...	1998	WWW
101	Rank aggregation methods for the Web	2001	WWW
86	Inferring Web communities from link topology	1998	ACM conference
79	The World-Wide Web	1994	Communications of the ACM
77	Silk from a sow's ear: extracting usable structures from the Web	1996	SIGCHI

The ALEF for this paper is 455, which would put it second in this list.

other hand, are geared towards a graduate student or someone new to a specific field of science. The *Classic* papers are foundational articles in a field that every first year graduate student in the field should read. Expert and classic recommendation examples for the famous Kleinberg paper on "Authoritative sources in a hyperlinked environment" can be found in Tables 2 and 3. The *Serendipity* recommendations are papers that are randomly chosen for every new user session. However, they are not randomly chosen from any part of the corpus. Rather, they are chosen from within the relevant community of papers as defined by the hierarchical network structure. The size distribution of communities for the different recommendations can be found in Fig. 3.

Fig. 2 shows an example of how Eigenfactor Recommends generates *Classic* and *Expert* recommendations. In this example the seed paper, or paper we are generating recommendations for, is highlighted in yellow and has an ALEF score of 12. First, we locate the leaf node the seed paper is in—colored sea foam green in Fig. 2. Next we generate a list of candidate papers, excluding the seed paper. When generating *Expert* recommendations we use all of the papers in seed paper's node (sea foam green), while *Classic* recommendations also use papers from sibling nodes (aqua). Finally, we order the papers according to their Eigenfactor scores (descending) and select the top-N papers.

In some unusual circumstances the *Classic* and *Expert* recommendations can be the same. If, for example, the top-3 papers in the seed node have a larger Eigenfactor score than all the papers in sibling nodes, the *Classic* algorithm would only select those papers, resulting in the same recommendations as the *Expert* algorithm provides. Conversely, it is possible that the *Classic* algorithm would not recommend any papers from the seed node if all papers in the seed node had unusually low Eigenfactor scores.

There are also situations where Eigenfactor Recommends cannot generate recommendations. Occasionally, InfoMap will place a paper into a node by itself, creating a singleton. For the AMiner set, there are approximately 80,000 singletons. Singletons normally arise when a paper has too few citations to be placed in a cluster, either due to errors in the dataset, lack of sufficient coverage (cited papers don't exist in this graph), or because the paper made very few citations. Since one cannot generate recommendations for singletons, we currently do not provide recommendations for these papers. However, there are several ways to deal with these papers. One way is to use cosine similarity [5] between citations from these papers and the citations from clusters in other parts of the tree. Papers could then be placed in clusters where they can receive recommendations.

3 Results

3.1 Recommendation Generation

We applied the EFrec algorithms to the AMiner dataset, generating recommendations for 1,218,504 papers, 58.2 percent of the dataset. The recommendations are available at http://babel.eigenfactor.org. We also provide the code[8] for calculating these recommendations, which can be used on any citation network. In addition to AMiner, we generated recommendations for several other datasets including papers on the arXiv, Microsoft Academic Search, the Social Science Research Network (SSRN), JSTOR, PLoS, and PubMed Central. This includes recommendations for more than 35 million papers from more than 300 million citations. Table 1 shows the number of recommendations for each dataset.[9]

Table 2 provides example *Expert* recommendations for the well-known *Hubs and Authority* paper by Jon Kleinberg [14]. "If you are reading the *Hubs and Authority* paper, you should also consider reading the Google-creating PageRank paper by Larry Page and Sergey Brin." The *ALEF* score for this paper is 455, which would put it second in this list. The recommendation list also includes the *Hubs and Authority* version that was presented at an ACM-SIAM symposium on discrete algorithms. Table 2 lists the top 10 *Expert* papers, but one could list the top N papers.

Table 3 provides the *Classic* Recommendations for the "Hubs and Authority" paper. The recommendation has zoomed out to the broader topic of information retrieval. This includes books and seminal papers in information retrieval and textual analysis.

3.2 Properties of Hierarchical Tree

The right-skewed distribution that we see at the paper level (Fig. 4) is also found at the cluster level. Fig. 3 shows the number of papers found in the clusters for the Expert and Classic recommendations. For the Expert recommendations, there were 202,437 clusters with at least one paper. The

8. https://github.com/jevinw/ALEF/
9. There is overlap in these datasets. For example, certain papers are listed both in MAS and in Pubmed Central. However, we have not collated the datasets. This is something we plan to do in the future.

TABLE 3
The Top 10 *Classic* Recommendations for the Paper, "Authoritative Sources in a Hyperlinked Environment" [14]

ALEF	Title	Year	Venue
948	Introduction to Modern Information Retrieval	1986	Introduction to Modern Information Retrieval
789	The anatomy of a large-scale hypertextual Web search engine	1998	WWW
725	Modern Information Retrieval	1999	Modern Information Retrieval
717	Compilers: principles, techniques, and tools	1986	Compilers: principles, techniques, and tools
713	Information Retrieval	1979	Information Retrieval
539	Latent dirichlet allocation	1999	The Journal of Machine Learning Research
477	GroupLens: an open architecture for collaborative filtering of netnews	1994	CSCW
468	Foundations of statistical natural language processing	1999	Foundations of statistical natural language proc.
450	Building a large annotated corpus of English: the penn treebank	1993	Computational Linguistics
445	The mathematics of statistical machine translation: parameter estimation	1993	Computational Linguistics

The ALEF for this paper is 455, which would put it 9th in this list.

average cluster size is **6.4** papers per cluster with a right-skewed distribution and a variance of 739 papers. The largest cluster includes 2,688 papers, with top 10 papers in this group shown in Table 4. Appropriately for the subject matter of this article, this cluster is about recommendation algorithms, with the top papers being about collaborative filtering. Because this is the largest cluster, the most recommended papers would be those listed in Table 4, if a recommendation was requested for every paper in the corpus. Since the *Expert* recommendation generates recommendations from the lowest part of the tree, it contains more, smaller clusters than the *Classic* recommendation, which produces recommendations by zooming out 1 level on the citation tree. There were 23,831 classic clusters with at least 1 paper. There were 174 singletons (0.7 percent of all clusters) not included in Fig. 3.

Singletons are clusters with only one paper. For Fig. 3 we did not include singletons. There were 79,242 clusters (39 percent of all clusters) that were removed. The number of singletons generally decreases with increased citation density. Like most bibliographic datasets, the AMiner network is sparse, which partly explains why 39 percent of the expert clusters are singletons. The papers in these singleton clusters tend to be sparsely connected to the larger groups within their discipline. If you only include clusters with more than 1

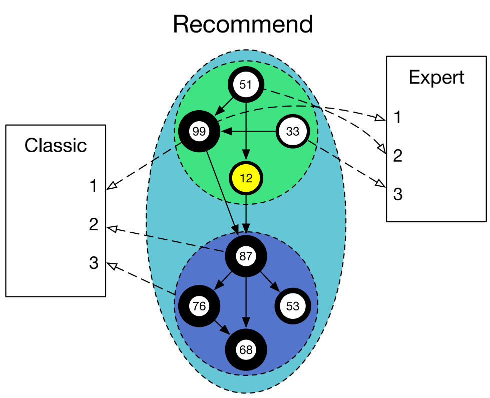

Fig. 2. *Recommendation closeup.* A closeup of Fig. 1.4 shows how *Classic* and *Expert* recommendations are generated for a given seed paper (highlighted, score of 12). *Expert* recommendations are drawn from the same leaf node (sea foam green) as the seed paper, while *Classic* recommendations include sibling nodes as well (aqua). Papers are ordered by their Eigenfactor score, descending. The *Classic* recommendations provide a more diverse set of papers, while *Expert* recommendations are more specific.

paper, the average cluster size increases to 9.8 papers per cluster. There also exists clusters at the highest level of the tree with singletons. Typically, the highest level clusters as you move up the tree include tens of thousands of papers. There were 122 clusters in the tree that fit this criteria. These are clusters that include papers highly disconnected from the rest of the corpus. These papers receive no recommendations with our algorithm. Other techniques (e.g., semantic similarity) must be used for these kinds of papers.

3.3 Sparseness of Citation Graphs

The AMiner network, just like most citation databases, is sparse. The average out-degree (i.e., the number of citations from an AMiner paper to other AMiner papers) is 3.8 with a standard deviation of 7.4. The average in-degree distribution (e.g., the number citations to AMiner papers from AMiner papers) is right skewed with a mean of 8.5 and a standard deviation of 34.9.

Although most readers will be familiar with the AMiner dataset provided for this issue, we want to take a moment to describe some attributes pertinent to citations based recommenders. The most important factor for citations based recommenders is the density of connections between nodes, that is the number of citations a paper makes. For this dataset there are a large number of papers that make no citations, 1,233,562 of the total 2,092,356 papers (59 percent). Part of this is incompleteness in the data set. For example, the paper *Rough computational methods for information systems* by Guan and Bell is listed as having no citations, but the original paper actually has 14 papers in its bibliography.

This incompleteness in data is very common for datasets of scholarly articles, and it represents one of the greatest challenges to large scale deployment of citation-based recommenders. It is interesting to note that this problem is worse in scholarly literature than on the web. The web benefits substantially from a standardized link format, and easy disambiguation between sites. Scholarly literature is inconsistent in its use of bibliographies, many of which are encoded in textual formats and have no semantic markup to indicate what is a citation and what isn't. Even worse the textual formats available vary widely, from convoluted PDFs and TEX files to proprietary formats that cannot be reliably parsed. Figs. 4 and 5 shows the citation counts for the entire dataset. Recommender systems can also be evaluated by the coverage[38] they provide. This metric can be measured in various ways, however it is commonly described as the percentage of items

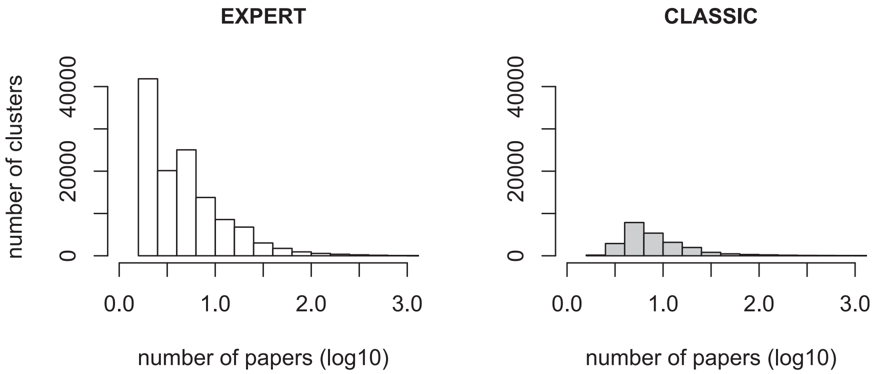

Fig. 3. *Cluster size distribution.* The bar plots show the number of clusters with a given number of papers. For the expert recommendation, clusters are extracted from the lowest part of the tree; therefore, there are more clusters with smaller number of papers. For the classic recommendation, the algorithm zooms out one level of the tree. This produces fewer clusters with more papers in each cluster. We did not include singletons – clusters with only one paper – in this figure. There were approximately 80k singletons in the AMiner tree.

for which recommendations can be formed. By this metric EFrec performs fairly well, with recommendations available for 58.2 percent of the AMiner dataset (1,218,504/2,092,356). Co-citations, by contrast, is only able to generate recommendations for 7.79 percent (163,051/2,092,356) of the corpus.

3.4 Recommendation Overlap

Another metric that can be used to compare various recommendation algorithms is how much overlap there is in recommendations they generate. There are several reasons this metric is useful. First, if you are claiming to produce better recommendations than the competition they would have to also be different recommendations. If Eigenfactor and co-citation have substantial overlap it would be unlikely that they generated materially different recommendations. Second, there is a lot of discussion around hybrid or ensemble recommenders. The intuition behind this idea is that recommenders all have strengths and weaknesses, but if we combine many different recommenders we can generate better overall recommendations in more situations. Therefore, one would want recommenders that are "orthogonal" to each other; they should produce substantially different recommendations for the same paper.

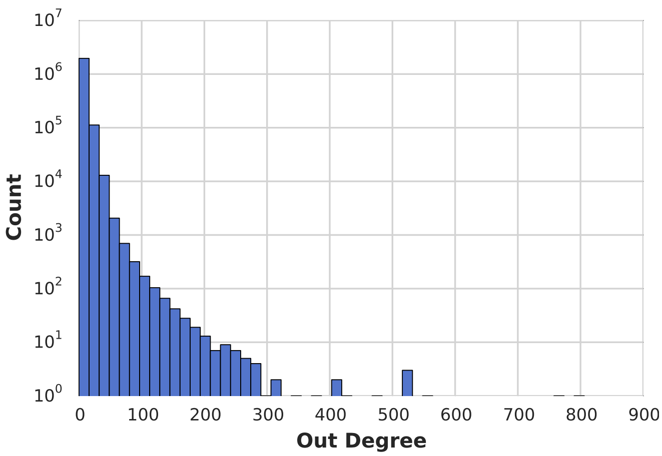

Fig. 4. *AMiner paper's out-degree distribution.* The bar plot shows the frequency of outbound citations made by papers in the AMiner dataset. The largest value, 0, accounts for 1,233,562 papers or 59 percent of the network.

We compared the overlap of the Eigenfactor Recommends (EFrec) results to the well-known co-citation recommendation method. To do this we generated recommendations for a set of papers, took the top-n from each algorithm, calculated the intersection and divided by n. The below equation describes this process where N is the set of papers to generate recommendations for, O_i is the overlap score for two recommendations, $C_{i,n}$ is the top-n recommendations generated by the co-citation method for paper i, $R_{i,n}$ is the top-n recommendations generated by the co-citation method for paper i, and n is the maximum number of recommendations. If both $C_{i,n}$ and $R_{i,n}$ were empty the paper was thrown out,

$$\forall i \in N, O_i = \frac{|\, C_{i,n} \cap R_{i,n} \,|}{n}. \tag{1}$$

When running over the entire miner data set ($N = aminer$) we compared the recommendations for 1,218,504 papers, with an average overlap of 0.011845 and a standard deviation of 0.064401. One potential issue here is the limited number of recommendations the co-citation method can generate for this dataset; our implementation only generated recommendations for 163,051 papers, 7.79 percent of the entire corpus. To further validate our overlap findings we ran the experiment again, this time only using papers that the co-citation method had generated a recommendation for as the input set (N). With this smaller dataset we compared recommendations for 163,051 papers, finding an average overlap of 0.088522 and a standard deviation of 0.155588.

4 Discussion

In this paper, we describe a simple method for recommending scholarly papers at different scales. To do this, we (1) assemble a citation graph, (2) rank the nodes according to a modified form of PageRank, (3) cluster the network hierarchically using the MapEquation framework and (4) then select recommendations for a seed paper given its location in the hierarchical tree. For the last step in the algorithm (step 4), we describe three approaches for selecting articles in this tree at different scales of relevance, which we dub *Expert*, *Classic*, and *Serendipity*. However, one could deploy many other variants. For example, one may select articles within a

TABLE 4
The Top 10 Papers for the Largest *Expert* Cluster

EF	Title	Year	Venue
477	GroupLens: an open architecture for collaborative filtering of netnews	1994	CSCW
404	Using collaborative filtering to weave an information tapestry	1992	Communications of the ACM
358	Social information filtering: algorithms for automating word of mouth	1995	SIGCHI
346	Empirical analysis of predictive algorithms for collaborative filtering	1998	Uncertainty in AI
259	Item-based collaborative filtering recommendation algorithms	2001	WWW
259	Item-based collaborative filtering recommendation algorithms	2005	WWW
229	GroupLens: applying collaborative filtering to Usenet news	1997	Communications of the ACM
215	Fab: content-based, collaborative recommendation	1997	Communications of the ACM
204	An algorithmic framework for performing collaborative filtering	1999	ACM SIGIR
204	Recommender Systems	1997	Communications of the ACM

Germane to this paper, the articles in this group are mostly about recommendation research.

cluster based on semantic similarity or co-authorship. One could also zoom out multiple levels depending on thedepth of a discipline. The authors of this paper have developed similar network-based ranking methods at the author-level that could be applied to this ranking system [17].

We found that EFrec performed as well as the widely-used co-citation method on papers that both methods can rank—but EFrec provides far more recommendations (more than twice as many for both expert and classic recommendations), and it provides recommendations at different levels of granularity (expert and classic). More experiments are needed to further analyze and compare this approach to other citation-based approaches.

We performed some preliminary analysis on live users on the Social Science Research Network to evaluate EFrec's performance against other recommendation algorithms. Using an A-B testing environment developed by SSRN [37], we compared EFrec classic and expert to co-citations and co-downloads. Users were randomly assigned one of the four recommenders with equal probability whenever they viewed an article page. As Table 5 shows, Eigenfactor expert and co-citation had very similar click-through rates (CTR): 0.24 and 0.26 percent respectively. Eigenfactor classic's CTR was half co-citations at 0.13 percent. Though Eigenfactor did not outperform co-citation in this experiment, it does provide substantially greater coverage with comparable recommendation performance. Co-citation could only generate recommendations for 94,043 of the 426,412 papers (22 percent), while EFrec expert and classic could provide recommendations for 215,627 (51 percent) and 227,049 (53 percent) of papers respectively.

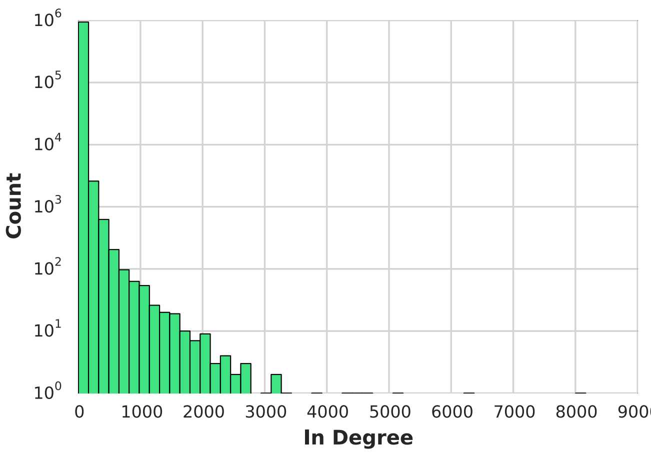

Fig. 5. *AMiner paper's in-degree distribution*. The bar plot shows the frequency of inbound citations of papers in the AMiner dataset. Most papers are cited few times, while one paper is cited 8,166 times.

We also had usage data for the current production SSRN recommender: co-downloads. Co-download is a collaborative filtering method that tracks user downloads to generate recommendations. It is important to keep in mind that co-downloads and citation based methods are not measuring the same thing—co-downloads is a measure of popularity, citation based methods of importance/impact. The difference between citation based methods and collaborative filtering is illustrated by visiting a product page on amazon.com. For most products Amazon provides at least two sets of recommendations: "Customers Who Bought This Item Also Bought" and "Compare to Similar Items". Collaborative filtering methods (like co-download) are the "Customers Who Bought This Item Also Bought" style of recommendations, though for co-download it would more accurately be titled "Users Who Read This Paper Also Read". This is also a measure of popularity, recommending the most downloaded items. EFrec is akin to the "Compare to Similar Items"—we find similar papers based on the latent hierarchical structure encoded in citation networks. In addition ,we note that co-download approaches require that we have detailed information about the browsing and searching activities of each and every individual user. The SSRN platform, with whom we conducted these tests, have this information because user login is required to use their system—but this is the exception rather than the rule among platforms that would benefit from deploying a recommendation system. As a result, a citation-based method may be of great value even if it performs modestly in head-to-head comparisons with co-download approaches.

Although we do not have the exact coverage numbers for co-download, the lower appearance count implies that it is not able to generate recommendations for as many papers as EFrec. However, those recommendations co-download does generate are quite effective, with a 0.69 percent CTR compared to EFrec expert's 0.24 percent. One possible explanation for this impressive performance is that co-download can quickly begin recommending new papers.

There are advantages and disadvantages in using citations as the primary substrate for recommending papers. One clear advantage is the insularity of this method to textual noise; it simply looks at the connections among papers and ignores

TABLE 5
SSRN Usability Experiment

Algorithm	Coverage		Appearances		Clicks		Downloads		CTR	D/C	D/A
	Papers	%	Count	%	Count	%	Count	%			
Co-Download†	N/A	N/A	272,760	22.45%	1,882	51.63%	556	56.22%	0.69%	29.54%	0.20%
Eigenfactor Classic	227,049	53%	462,653	38.08%	595	16.32%	167	16.89%	0.13%	28.07%	0.04%
Eigenfactor Expert	215,627	51%	410,622	33.80%	988	27.11%	209	21.13%	0.24%	21.15%	0.05%
Co-Citation	94,043	22%	68,939	5.67%	180	4.94%	57	5.76%	0.26%	31.67%	0.08%

Coverage is the number of paper recommendations generated. This information was not provided for the co-downloads. The coverage is likely smaller than the citation-based methods. Appearance is the number of times recommendations for the given algorithm were shown to users. Clicks is the number of times one of those recommendations was clicked on. Downloads is how often a clicked paper was downloaded. CTR is click-through-rate, a measure of the frequency that users click on a recommendation, and is calculate as $clicks/appearances$. *D/C is similar to CTR, but measures progress from clicks to actual downloads and is calculated as* $downloads/clicks$. *While CTR indicates how interesting the title of a recommended paper is, D/C is a measure of how useful the paper actually is, at it tracks the final outcome: downloads. D/A shows the probability that a recommendation will finally be downloaded, and is calculated as* $downloads/appearances$. *Data collected from SSRN from July 13th - July 22nd, 2015.*
†Co-download is a collaborative filtering method, and direct comparison to content based methods (e.g., Eigenfactor, co-citation) can be misleading.

semantic information. The major disadvantage is due to the permanence of citations (and the lack of semantic information). It takes considerable time to accumulate enough citations to obtain signals of influence and community structure. For some fields such as economics, ecology and mathematics, the citation lags are much longer [16]. When compared to usage data, citations tend to better locate field-starting, classic papers, while usage-based methods locate recent, popular papers. Given the somewhat complementary nature of co-downloads and EFrec, we plan to integrate both into a hybrid recommender. Ideally, this will combine a strong impact measure and greater coverage of EFrec, while adapting quickly to changing fields and new papers.

One of the strengths of the method described in this paper is the simplicity of the algorithm conceptually and computationally. It is easy to describe, build upon and compute. In this paper, we simply sort the articles within the clusters using ALEF, based on a modified version of PageRank, and aggregate the clusters at different levels (Expert and Classic). Computationally, the method is relatively fast. We can run the entire AMiner dataset of 2.1 million papers and 8 million citations on a standard desktop machine (2.6 GHz Intel) on one core in less than 30 minutes. The ranking step (2 from Fig. 1) and the recommendation step (4 from Fig. 1) are especially fast. During the recent 2016 WSDM Cup Challenge we ran the ranking step (Fig. 1.2) on the Microsoft Academic Search citation graph, which contained over 49 million papers and 949 million citations, in approximately 30 minutes [28]. The bottleneck is the clustering step (3), but we are continuing to improve the speed of this step by parallelizing the code for multi-core machines and distributed systems [39]. Current details and improvements about the computational time can be found at babel.eigenfactor.org.

We would like to close the discussion with a short note about this "Special Issue on Big Scholar Data Discovery and Collaboration." Given the growth of science there is a strong need for better tools for navigating the literature, but most of the recommendation research has been developed for commercial uses (Netflix, Amazon, etc). We hope that scientists spend more time helping one another discover important, relevant papers and authors. A major reason for this lack of recommender development hasn't been lack of interest in developing better algorithms and tools, but rather the difficulty of accessing content. There has been some progress in this area, such as the recent release of the Microsoft Academic Search citation graph [40] and the AMiner graph for this special issue. Our hope is that publishers will continue to work more with researchers in this area so that development continues beyond this special issue.

5 Conclusion

In this paper, we describe a simple citation-based method for recommending articles. The method is based on the hierarchical structure of scientific knowledge, allowing for different scales of influence.

The method scales well for large citation networks and is made freely available at babel.eigenfactor.org. We have generated hundreds of millions of recommendations for tens of millions of scholarly articles, and we plan to continually generate more as bibliographic data becomes available.

In this paper, we compare EFrec to other recommendation methods, including citation based and usage-based ones, on live users of the Social Science Research Network. When compared to other citation based methods (co-citation), EFrec produced comparable recommendations as measured by click-through rate, but with substantially higher recommendation coverage: 52 versus 22 percent of the corpus. However, when EFrec is compared to collaborative filtering based methods (co-downloads), we found that EFrec has a substantially lower a click-through rate: 0.24 versus 0.69 percent. However, since these methods employ different data sources we see them as being more complimentary than in competition.

Finally, we see this as only one way of using the ranking and clustering aspects of the EFrec method. There are many variations and extensions that can be built on the general framework. We hope the method, code, and recommendations will provide fodder for further study in scholarly recommendation.

Acknowledgments

This work was supported the Metaknowledge Network funded by the John Templeton Foundation. The authors would like to thank Martin Rosvall and Daril Vilhena for early discussions around the topic of recommendation. They would also like to thank Ralph Dandrea, Anatoly Chukhin and Fernando Agostino at SSRN for helpful discussions and for providing data for the preliminary usage experiments that are being developed for a future paper.

REFERENCES

[1] D. J. De Solla Price, "Networks of scientific papers," *Science*, vol. 169, pp. 510–515, 1965.

[2] L. Bornmann and R. Mutz, "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references," *J. Assoc. Inform. Sci. Technol.*, vol. 66, pp. 2215–2222, 2015.

[3] A. E. Jinha, "Article 50 million: An estimate of the number of scholarly articles in existence," *Learned Publishing*, vol. 23, no. 3, pp. 258–263, 2010.

[4] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 990–998.

[5] P. B. Kantor, L. Rokach, F. Ricci, and B. Shapira, *Recommender Systems Handbook*. New York, NY, USA: Springer, 2011.

[6] Y. Koren, R. Bell, and C. Volinsky. (2009). Matrix factorization techniques for recommender systems. *Computer* [Online]. pp. 42–49. Available: https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

[7] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou. (2012, Feb.). Recommender Systems, p. 97 [Online]. Available: http://arxiv.org/abs/1202.1112

[8] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2002, pp. 253–260.

[9] X. Su and T. M. Khoshgoftaar. (2009). A survey of collaborative filtering techniques. *Adv. Artif. Intell.* [Online]. *2009(Section 3)*, pp. 1–19. Available: http://www.hindawi.com/journals/aai/2009/421425/

[10] O. Küçüktunç, E. Saule, K. Kaya, and U. Çatalyürek. (2012). Direction awareness in citation recommendation, vol. i, pp. 3–8 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.301.1927

[11] M. M. Kessler, "Bibliographic coupling between scientific papers," *Am. Documentation (pre-1986)*, vol. 14, no. 1, p. 10, 1963.

[12] H. Small, "Co-citation in scientific literature: A new measure of the relationship between two documents," *J. Am. Soc. Inform. Sci.*, vol. 24, no. 4, pp. 265–269, 1973.

[13] H. White and B. Griffith, "Author cocitation: A literature measure of intellectual structure," *J. Am. Soc. Inform. Sci.*, vol. 32, pp. 163–171, 1981.

[14] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[15] S. Brin and L. Page. (1998). The anatomy of a large-scale hypertextual web search engine, in *Proc. 17th Int. World-Wide Web Conf.* [Online]. Available: http://ilpubs.stanford.edu:8090/361/

[16] J. West, T. Bergstrom, and C. Bergstrom, "The eigenfactor metrics: A network approach to assessing scholarly journals," *College Res. Libraries*, vol. 71, no. 3, pp. 236–244, 2010.

[17] J. D. West, M. C. Jensen, R. J. Dandrea, G. J. Gordon, and C. T. Bergstrom. (2013, Apr.). Author-level Eigenfactor metrics: Evaluating the influence of authors, institutions, and countries within the social science research network community. *J. Am. Soc. Inform. Sci. Technol.* [Online]. vol. 64, no. 4, pp. 787–801. Available: http://doi.wiley.com/10.1002/asi.22790

[18] J. Bollen, M. A. Rodriquez, and H. Van de Sompel, "Journal status," *Scientometrics*, vol. 69, no. 3, pp. 669–687, 2006.

[19] S. J. Liebowitz, and J. P. Palmer, "Assessing the relative impacts of economics journals," *J. Econ. Literature*, vol. 22, no. 1, pp. 77–88, 1984.

[20] G. Pinski and F. Narin, "Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics," *Inform. Process. Manage.*, vol. 12, no. 5, pp. 297–312, 1976.

[21] T. H. Haveliwala, "Topic-sensitive pagerank," in *Proc. 11th Int. Conf. World Wide Web*, 2002, pp. 517–526.

[22] M. Gori and A. Pucci, "Research paper recommender systems: A random-walk based approach," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2007, pp. 778–781.

[23] M. Rosvall and C. T. Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems," *PloS One*, vol. 6, no. 4, p. e18209, 2011.

[24] J. D. West, J. Jacquet, M. M. King, S. J. Correll, and C. T. Bergstrom, "The role of gender in scholarly authorship," *PloS One*, vol. 8, no. 7, p. e66212, 2013.

[25] J. West, M. Rosvall, and C. Bergstrom, "Ranking and mapping article-level citation networks," *arXiv.org*, 2016.

[26] D. Walker, H. Xie, K.-K. Yan, and S. Maslov, "Ranking scientific publications using a model of network traffic," *J. Statist. Mech.: Theory Exp.*, vol. 2007, no. 06, p. P06010, 2007.

[27] S. Maslov and S. Redner, "Promise and pitfalls of extending Google's pagerank algorithm to citation networks," *J. Neurosci.*, vol. 28, no. 44, pp. 11 103–11 105, 2008.

[28] I. Wesley-Smith, C. T. Bergstrom, and J. D. West, "Static ranking of scholarly papers using article-level eigenfactor (ALEF)," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*.

[29] K. R. Larsen, D. Hovorka, J. West, J. Birt, J. R. Pfaff, T. W. Chambers, Z. R. Sampedro, N. Zager, and B. Vanstone, "Theory identity: A machine-learning approach," in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, 2014, pp. 4639–4648.

[30] R. Lambiotte and M. Rosvall, "Ranking and clustering of nodes in networks with smart teleportation," *Phys. Rev. E*, vol. 85, no. 5, p. 056107, 2012.

[31] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci.*, vol. 105, no. 4, pp. 1118–1123, 2008.

[32] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *Eur. Phys. J.: Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[33] C. E. Shannon, "A mathematical theory of communication," *Bell Labs Tech. J.*, vol. 27, pp. 379–423, 1948.

[34] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E*, vol. 80, no. 5, p. 056117, 2009.

[35] G. K. Orman, V. Labatut, and H. Cherifi, "On accuracy of community structure discovery algorithms," *arXiv preprint arXiv:1112.4134*, 2011.

[36] J. I. Perotti, C. J. Tessone, and G. Caldarelli, "Hierarchical mutual information for the comparison of hierarchical community structures in complex networks," *Phys. Rev. E*, vol. 92, no. 6, p. 062825, 2015.

[37] I. Wesley-Smith, R. Dandrea, and J. West, "An experimental platform for scholarly article recommendation," in *Proc. 2nd Workshop Bibliometric-Enhanced Inform. Retrieval*, 2015, pp. 30–39.

[38] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inform. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[39] S.-H. Bae, D. Halperin, J. West, M. Rosvall, and B. Howe, "Scalable flow-based community detection for large-scale network analysis," in *Proc. IEEE 13th Int. Conf. Data Mining Workshops*, 2013, pp. 303–310.

[40] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. P. Hsu, and K. Wang. (2015). An overview of Microsoft academic service (MAS) and applications," [online]. Available: http://dx.doi.org/10.1145/2740908.2742839

Jevin D. West received the PhD degree in biology from the University of Washington. He is an assistant professor at the University of Washington Information School and a cofounder of the Datalab. He is a data science fellow at the eScience Institute and Affiliate with the Center for Statistics and Social Sciences at UW. His research lies at the cross section of network science, scholarly communication, knowledge organization, and information visualization. He cofounded Eigenfactor.org—a free website that ranks and maps the scholarly literature in order to better navigate and understand scientific knowledge. He has been invited to give talks at more than 50 academic and industry conferences around the globe including Harvard, Stanford, and the National Academy of Sciences. Prior to joining the faculty at UW, he was a post-doc in the Department of Physics at Umea University in Sweden. [more details here: jevinwest.org]

Ian Wesley-Smith received the BS degree in computer science from Louisiana State University. He is currently working toward the PhD degree at the University of Washington Information School and is a member of the Datalab. His research interests include machine learning, network analysis, recommender systems, distributed computing, and cryptography. He is currently working on Babel, a platform to develop, distribute and measure scholarly article recommendations, all as a web service, freely available to everyone. Prior joining the DataLab, he was a developer on Amazon Web Services and spent several years as an engineer on Windows Security.

Carl T. Bergstrom is a professor in the Department of Biology at the University of Washington. He is a pioneer in applying the methods of network science to citation networks. Together with author Jevin West, he developed and launched the Eigenfactor Metrics, a system for ranking scholarly journals the full network of citations. Eigenfactor has since become widely adopted as an industry standard. He is also the coinventer of the InfoMap algorithm, a leading method for mapping the community structure of large networked systems. In addition to his work in scholarly communication, he is the coauthor with Lee Alan Dugatkin of a leading evolutionary biology textbook, *Evolution.* [more details here: octavia.zoology.washington.edu]