Ranking of Academic Papers

CS 519: Data Science
December, 2018

Ankit Aggarwal
CS, Stony Brook University

Keshav Gupta
CS, Stony Brook University

Soumyadeep Chakroborty
CS, Stony Brook University

1 Abstract

While the h-index [1] has long served as a touchstone for measuring and ranking the research output of a scholar, it is deficient in many ways. It has fallen behind in an age where more and more bibliographic data is available beyond mere citation count. We aim to form the $j - index$ of a publication (paper, book, lecture notes, thesis etc.) and by extension, the $s - index$ of a researcher. These metrics will aim to remove biases such as those derived from the area of research - field bias (papers in Neuroscience are cited significantly more than papers in Computer Science) or those derived from year bias (research output spikes due to the invention of critical technology). These metrics would also aim towards a more holistic appreciation of the research and researcher, incorporating aspects such as the intra-disciplinary and inter-disciplinary reach of research, awards, quality of institution and the rank of the journal/conference of publication.

2 Background

To rank researchers, the h-index has been widely used in the research community. It is calculated as follows: If a researcher's papers are ranked in descending order by number of times cited, C_i, where i = 1, 2,..., N, then the h-index is the maximum i such that $C_i \geq i$.

This work [2] discusses several shortcomings of the h-index - mainly being that it favors researchers who have produced a large quantity of modestly cited material and the fact that it completely ignores the top most cited publications of the data. As the study highlights, such wrongly favored researchers often end up having a better h-index than the most eminent scholars in the field. The h-index does not value authors who have a small number of highly cited works. Google introduced the i-10 index, which alleviates this problem to an extent as it is measured by the number of publications that have more than 10 citations.

Even the i10 index does not account for biases such as that of the field of research. The number 10 seems to be too general and should be specific to each field. Biases borne from the age of a paper (papers which are old have room to accumulate citations over time) are not addressed by these metrics.

A variety of metrics come under the umbrella term Altmetrics [5] (Alternative metrics) have been proposed and evaluated in the community. These involve looking at factors such as the number of downloads of a publication, social media data such as the number of views and data that evaluates the impact of the publication on the news. Attaining a dataset that unifies these metrics for across all research fields is a challenging task as often such data is proprietary.

The journal/conference ranking in which researchers publish can also be taken into account. JIF (Journal Impact Factor)s measure the average number of citations a journal published article has received over the course of two years since it was published. Not all journals publish their impact factor as many believe the JIF to be a skewed metric. This can be attributed to the fact that journals with high impact factors often arrive at that figure with a few highly-cited articles. The EigenFactor score [3] for a journal is thought to be more robust than the traditional impact factor. The Scimago journal factor[4] takes into account the previous three years.

PageRank[14] is a method that has been often adopted to find nodes with high centrality (importance) in graphs. It is based on the probability of a random walk through a graph/network ending at a specific node. PageRank, thus naturally lends itself to a citation network - graph of "cited-by" relationships between publications.

3 Experimental Setup

We used a Google Cloud Platform instance with 4 CPUs and 52GB RAM from the us-east1-b zone with a hard disk size of 100GB to perform our experiments. As explained in the following sections, even such scale did not eliminate the need for careful management of resources such as the adoption of chunking strategies for memory-intensive tasks.

4 Datasets

- We used the DBLP v10 dataset [6] which consists of over 3 million publications and 25 million cited-by relationships. This is our primary dataset and we have used it directly to construct the citation graph to calculate the reach function.

  From this dataset alone, we used for publications and researchers fields like: citation count, authors of a publication, the publication date etc.

- We used the Scimagojr journal and conference ranking dataset [7] in order to find the journal/conference rank (Scimago Journal Rank - SJR[4]) for a given publication. It contains data on almost 34k journals/conferences, almost 7k of them being in Computer Science. This dataset also contains the sub-fields within Computer Science that a particular journal can be ascribed to (such as AI, ML etc.). As explained in the following section, these factors are directly fed into the $s - index$ and the $j - index$ computation.

- Joining the aforementioned datasets was non-trivial as the join was to be performed on the venue column of the DBLP dataset and the Title column of the Journal Ranking dataset. Both of these columns are strings and after a vanilla equi-join, only 63k of out about 307k rows were matched. After employing a fuzzy-join using the fuzzymatcher[8], based upon the principle of Probabilistic Record Linkage [9], about 180k rows were reasonably matched (with a match score of 0.25 or above). An example of a match of score 0.25 is depicted in Figure. Fuzzy joining the two datasets was proved to be hard on memory, forcing us to join the two datasets in 11 chunks of size 300k rows each.

- The Open Academic Graph project [10], which is a project that links the Microsoft Academic Graph (MAG) and the AMiner datasets, proffers a much richer dataset, having over 166 million publications from MAG and 154 million from Aminer. These publications are not limited to Computer Science and are spread across a wide variety of disciplines. The dataset is feature-rich, having fields such as keywords, language, article length and the document type. Using this dataset instead of the DBLP v10 dataset proffers obvious advantages. However, the infrastructure required to carry out this task is considerable and well beyond the non-premium-plan limitations of the Google Cloud Platform. The dataset has an overall compressed size of 143.6GB. The Aminer data by itself takes up 39GB. We had considered using a subset of the Aminer data, to make the problem tractable in terms of memory and computational overhead. However, this approach has serious implications on the calculation of the reach function. This is attributable to the fact that for any given publication, the publication may be cited by another publication that does not reside in the subset that we have considered. This is why we decided to move ahead with the DBLP dataset, coupled with the Scimago Journal Rank dataset.

- We use the top-1000 ranked researchers in Computer Science by $h - index$ from guide2research.com[11] for validation. We extracted the data from the webpage by a simple drag-and-select operation for each page of results (10 pages) and then processed the resultant plain-text file programatically.

- We use a Kaggle dataset[12] having 24,000+ academic arXiv papers in Computer Science to run our predictions. This dataset has only a limited set of columns - authors, title, abstract, field of study (available as arXiv subject tags - e.g. cs.AI). Since the dataset had only a relatively small number of such tags, manual reconciliation of the field(s) of study between our master dataset and this dataset was sufficient. Expansions of tag abbreviations were available here:[13].

5 Time-series analysis

We analyzed the time-series data inherent to our dataset, which helped us root out two biases intrinsic to academia. Field bias arises from some fields getting more citations than other fields. This could be attributed to

a variety of factors - such as the accessibility of a certain subject, its relative appeal or even its hype. Year bias arises from some years being witness to more citations than other years - breakthroughs in fields, introduction of buzzwords all contribute heavily to this bias. Figure 1 leaves no doubt to just how prevalent these biases are. We can see spikes in many of the fields in specific years apart from the general increase over the years. The increase in number of papers is certainly much more pronounced for certain fields such as "Computer Networks and Communications" and "Artificial Intelligence".

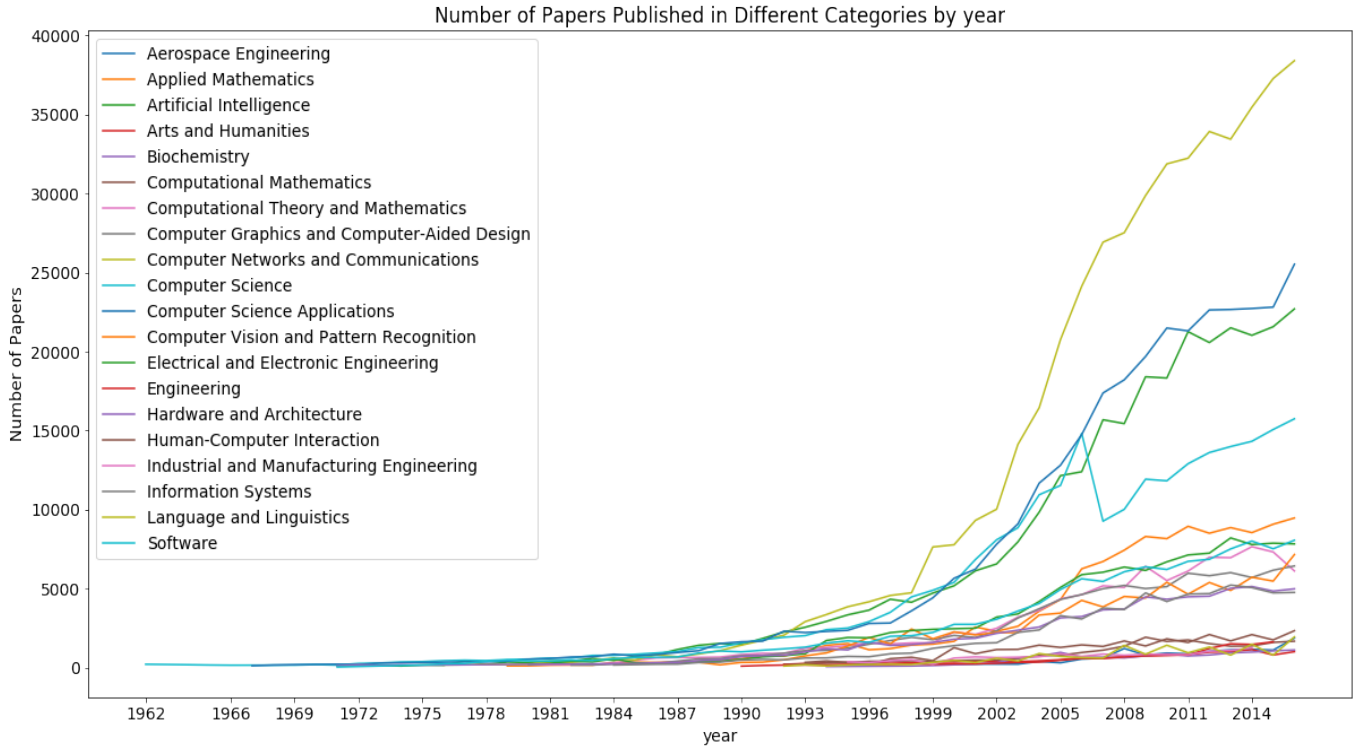


Figure 1: Time series analysis of DBLP publications

6 Ranking Method

First we calculate for every publication p, a citation score, which corrects for both field and year bias. The citation score (C(p, f, y)) for a publication p published in year y in field of study f corrects for these biases and is given by the formula:

$$C(p, f, y) = \frac{c(p)}{(C^m(f, y) \times C^t(f, y))}$$

where c(p) is the number of citations received since date of publication and $C^m(f, y)$ is the maximum number of citations any paper published in year y for field f has achieved and $C^t(f, y)$ is the total number of citations for all papers published in year y for field f.

For every publication, we obtain its SJR value SJR(p) as described in the Dataset section.

Finally, we also obtain the reach score(R) for every publication. This is explained in detail in the following section.

The j-index J(p) of the publication p is given by the formula:

$$J = (w_1 \times SJR^o(p)) + (w_2 \times C^o(p, f, y)) + (w_3 \times R^o(p))$$

where $SJR^o(p)$, $C^o(p, y)$), $R^o(p)$ are the scaled SJR score, citation score and reach score respectively. Scaling of all these fields follow the same strategy: divide each score by the maximum value of that score in the dataset. w_1, w_2 and w_3 represent weights for each factor. We empirically refined these weights to be finally: w1 = 0.01, w2 = 0.24 and w3 = 0.75. Then $s - index$ of a researcher r is computed as follows:

$$S(r) = \sum_{p \in p(r)} J^o(p)$$

where p(r) is the set of publications in which r is an author and $J^o(p)$ is the scaled j-index (scaled with the same strategy as above).

7 Reach

The "reach" of a work recognizes how fundamental it is. We consider a graph of citations, with the nodes being publications and directed edges representing the "cited-by" relationship. Figure 1 shows two examples of such a graph.

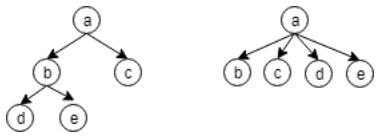


Figure 2: Two possible citation networks for publications a,b,c,d,e

We define the reach of a node p as follows:

$$reach(p, par, h) = k(par \rightarrow p) \times h \times |adj(p)| + \sum_{q \in adj(p)} reach(q, p, h + 1)$$

The reach algorithm resembles the modified version of PageRank[14] algorithm, however, it only accounts for outgoing edges rather than a combination of incoming and outgoing edges for each node as seen in traditional PageRank. This reach score is in turn fed back into the j-index ranking metric calculation as discussed previously.

The above function calculates the reach of node p with parent par (i.e there exists an edge $par \rightarrow p$) and adjacency list adj(p) on the h^{th} hop of a traversal of the citation graph starting from some starting node. This structure awards more weight/importance to a publication which is cited indirectly at a larger distance in the network graph. k is a multiplicative factor that influences this. k is defined as follows:

$$k(par \rightarrow p, h) = \begin{cases} |p.subf \setminus par.subf| * h, & \text{if } |p.subf| \neq 0 \\ h, & \text{otherwise} \end{cases}$$

where subf is the set of sub-fields, $|subf|$ is the cardinality of the set.

We collect the values of k in a dictionary by traversing each edge in the citation graph up front. Then we memoize the values of reach in a dictionary as well, as there are overlapping subproblems of the form: reach(p, h).

For memory tractability, we maintain a recursion depth cutoff d, till which we explore citation relationships in the citation graph. We executed the reach computation for d = 5, 6 and 10. We observed that a cutoff greater than 10 resulted in convergence of the reach score and impacted the usefulness of the reach feature.

8 Results and Validation

8.1 Initial Iteration - Accounting for year bias

1) **j-index**: In the first iteration, we only considered year bias when we calculated the citation score. Effectively, in reference to the formula for citation-score in Section 5, we get rid of the factor f to arrive at :

$$C(p, y) = \frac{c(p)}{(C^m(y) \times C^t(y))}$$

Figure 3 depicts the top 10 publications by j-index. We can see that our ranking has captured some prominent publications in the field. Since we assign a larger weight towards reach score (w3 = 1.0) as compared to citation score (w2 = 0.5) and SJR score (w1 = 0.01), our results favor publications with a high reach score. This is especially evident in the difference in j-indices of the 0th and 1st ranked papers "The Design and Analysis of Computer Algorithms" and the "Introduction to Decision Trees". The 0th paper has a j-index of 1.000412 and the 1st paper has 0.6666723. This is directly attributable to the huge difference in their reach scores. Also, the 2nd ranked paper has a much lower citation count but still has a significant reach - This seminal paper, Visual Learning and recognition of 3-D objects has "reached" across multiple sub-fields. These aspects show that the reach score is a clear differentiator.

We also see that the 8th ranked publication: "On Quine's Axioms of Quantification" by George Berry has a significantly lower citation count (50) and reach score (1.68e-13). However our model rewards the paper as in 1941, when the work was published, the total number of citations racked up by other works in Computer Science in 1941 amounted to 64. This work contributes to 50 of the 64. Clearly this is the highest cited paper from that year. This is why this paper has the highest citation score of 1.0. The bias against papers recently published also seems to be somewhat accounted for due to the presence of papers from the 90s in the top 10. This depicts that the citation score is also a differentiator.

Finally, the SJR score perhaps has a very small weight (w1 = 0.01), rendering it relatively insignificant. This is done on purpose. We wanted to see if one of the factors would be dominated as a result of our weighting scheme. Notice how the 1st and 2nd ranked publications have a very small difference between their reach scores and citation scores but a significant difference between their SJR scores (0.05 and 0.16 respectively). Had there been no dominance, these two publications would have had their ranks flipped.

	title	authors	venue	year	n_citation	sjr_score	reach_score	citation_score	j-index
0	The Design and Analysis of Computer Algorithms	[Alfred V. Aho, John E. Hopcroft]		1974	13227	0.018838	1.000000e+00	0.000447	1.000412
1	Induction of Decision Trees	[John Ross Quinlan]	Machine Learning	1986	19320	0.050355	6.661679e-01	0.000103	0.666723
2	Visual learning and recognition of 3-D objects...	[Hiroshi Murase, Shree K. Nayar]	International Journal of Computer Vision	1995	2736	0.166715	6.568129e-01	0.000003	0.658482
3	Example-based learning for view-based human fa...	[Kah Kay Sung, Tomaso Poggio]	IEEE Transactions on Pattern Analysis and Mach...	1998	2234	0.171497	5.570867e-01	0.000003	0.558803
4	Notes on Data Base Operating Systems	[Jim Gray]	Advances in Computers	1978	2727	0.014853	5.345290e-01	0.000048	0.534702
5	Support-Vector Networks	[Corinna Cortes, Vladimir Vapnik]	Machine Learning	1995	26114	0.050355	5.326541e-01	0.000032	0.533174
6	Implementing remote procedure calls	[Andrew Birrell, Bruce Jay Nelson]	ACM Transactions on Computer Systems	1984	2838	0.104188	5.258613e-01	0.000033	0.526920
7	Probabilistic Reasoning in Intelligent Systems...	[Judea Pearl]		1988	6589	0.018838	5.204852e-01	0.000024	0.520686
8	On Quine's Axioms of Quantification	[George D. W. Berry]	Journal of Symbolic Logic	1941	50	0.027605	1.686521e-13	1.000000	0.500276
9	The design and implementation of INGRES	[Michael Stonebraker, Gerald Held, Eugene Wong...	ACM Transactions on Database Systems	1976	539	0.053760	4.983320e-01	0.000011	0.498875

Figure 3: Top 10 papers by $j - index$

2) **s-index:** Figure 4 depicts the top-10 authors by s-index and a comparison of the h-index and the s-index.

We see that Takeo Kanade and Andrew Zisserman feature in both top-10 lists. A few others have s-index rank close to the top 10.

Herbert Simon and Terrence Sejnowski, who have very high h-indices have low s-indices. While Herbert Simon[15] is a very well known economist and political scientist, Terrence Sejnowski[16] is a leader in the field of Computational Neurobiology. Their ample citations and reach outside CS is not captured by our dataset.

Jiawei Han[17] is a stalwart in the field of Data Mining, with many books and stellar publications. However, his research is narrower and more focused, as depicted in his reach score of 0.176.

Azriel Rosenfeld [18] was a leading researcher can be described as the father Computer Image Analysis, with pioneering contributions across the field over 40 years. He wrote the first textbook in the field, was founding editor of its first journal and was co-chairman of its first international conference. He published over 30 books and over 600 book chapters and journal articles, and directed nearly 60 Ph.D. dissertations. This suggests a high reach score and citation score for a large number of his publications, which explains his rank.

To mathematically compare the s-index and the h-index, we define the following metric - Rank Divergence, which takes the absolute difference between the s-index and h-index ranks of a researcher ($R_{S(r)}$ and $R_{H(r)}$ respectively):

$$\delta_R = |R_{S(r)} - R_{H(r)}|$$

To summarize δ_R, we can compute $\widetilde{\delta_R}|_n$, the median of the top-n rankings for both ranking metrics. We pick the median in order to remove the sensitivity to large difference outliers. For n = 10, we achieve: $\widetilde{\delta_R}|_n = 142.5$.

	authors	academic_age	s-index
0	Azriel Rosenfeld	42	4.014671
1	Michael Stonebraker	44	3.354656
2	Takeo Kanade	45	3.073051
3	Jitendra Malik	35	3.003419
4	Alex Pentland	36	2.953394
5	Andrew Zisserman	32	2.947706
6	Cordelia Schmid	25	2.597863
7	Pietro Perona	29	2.563672
8	Tomaso Poggio	32	2.539557
9	Robert E. Schapire	30	2.505502

(A)

	Name of Researcher	h-index	h-index rank	s-index rank	s-index
0	Anil K. Jain	179	0	27	1.712872
1	Herbert Simon	175	1	1230	0.227589
2	Jiawei Han	162	2	328	0.516504
3	Terrence Sejnowski	151	3	256	0.595427
4	David Haussler	151	4	13	2.265852
5	Takeo Kanade	151	5	2	3.073051
6	Philip S. Yu	148	6	198	0.680757
7	Michael I. Jordan	148	7	56	1.284312
8	Scott Shenker	146	8	102	0.995337
9	Andrew Zisserman	144	9	5	2.947706

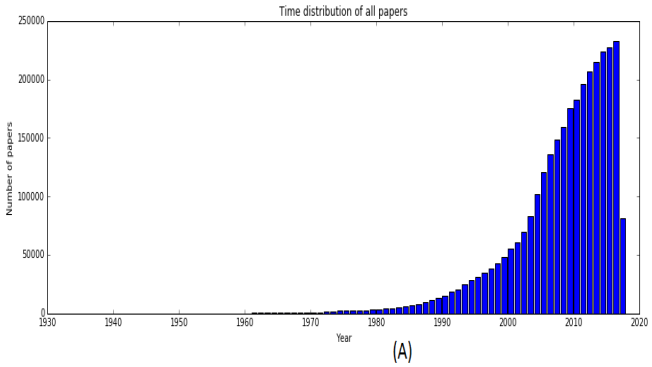
(B)

Figure 4: (A) Ranking of top-10 researchers by $s-index$. (B) Comparison of h-index and $s-index$

3) Our rankings and these time distribution plots (Figure 5A, 5B) show an important characteristic. Older papers usually have a lower citation count given their age and changes in their respective fields. To rise in rank, they have to be extremely fundamental, i.e. they have to have a very high reach. New papers have to be have a very high citation score on the other hand to deal with their inherently low reach (as there has not been enough passage of time for reach to develop). As the figure 5(B) depicts, research published in the 90s have the best ranks. They are at the sweet spot between reach and citation score.

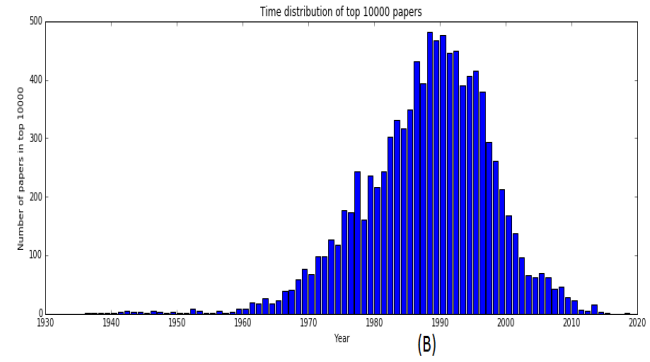
4) From the correlation analysis (Fig 5(C)), we can see that the sjr_score, $reach\_score$ and the $citation\_score$ are not correlated at all, showing that they are not derivative - i.e. the j-index is approached from completely different angles.

5) We can see that the academic age distribution of the researcher is normal-like (Fig 5(D)), emphasizing that an academic age of about 30 is the most common among top researchers.



	sjr_score	reach_score	citation_score	j-index
sjr_score	1.000000	0.022276	0.000571	0.166167
reach_score	0.022276	1.000000	0.001642	0.975747
citation_score	0.000571	0.001642	1.000000	0.166137
j-index	0.166167	0.975747	0.166137	1.000000

(C)



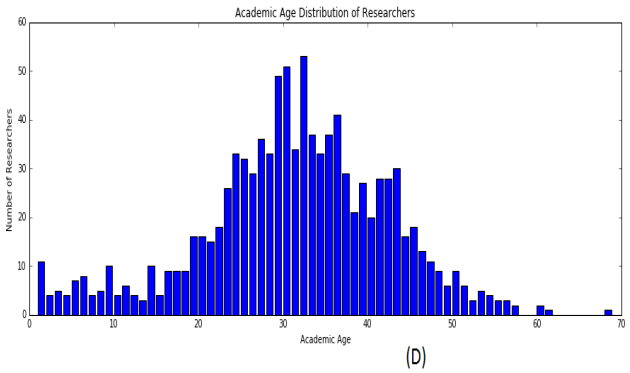


Figure 5: (A) Distribution of papers by publication year. (B) Distribution of top 10,000 papers by publication year. (C) Correlation matrix of j-index factors. (D) Distribution of academic age of **top 1,000** researchers.

8.2 Final Iteration: Accounting for field and year bias

1) **j-index:** We corrected for field bias in addition to year bias to calculate the citation score, as explained in Section 5. We used:

$$C(p, f, y) = \frac{c(p)}{(C^m(f, y) \times C^t(f, y))}$$

We also tuned the weights with reach score (w3 = 0.74), citation score (w2 = 0.25) and SJR score (w1 = 0.01). From the top-10 papers sorted by j-index as depicted in Figure 6, we can observe much of the same desirable characteristics as in our earlier iteration. The ranking has again picked seminal contributions to the field - be it John Ross Quinlan's "Induction of Decision Trees", Codd's invention of the Relational Model and Leslie Lamport's work on Vector Clocks. From Figure 7, we observe that again between citation score, reach and SJR score, there is little correlation - an indicator that these factors are non-derivative.

Looking at the differences in j-indices between publications in the top 10, we see that the deviation is approximately 0.03 and the maximum deviation is 0.09 (as compared to 0.34 in our first iteration). This points towards a more balanced weight assignment.

	title	authors	venue	year	n_citation	fos	citation_score	SJR	reach	j-index
0	Induction of Decision Trees	[John Ross Quinlan]	Machine Learning	1986	19320	Software	0.124708	0.050355	1.000000	0.771681
1	A relational model of data for large shared da...	[E. F. Codd]	Communications of The ACM	1970	7295	Computer Science	0.516114	0.051369	0.744526	0.680492
2	Visual learning and recognition of 3-D objects...	[Hiroshi Murase, Shree K. Nayar]	International Journal of Computer Vision	1995	2736	Artificial Intelligence	0.013811	0.166715	0.894354	0.666942
3	Distinctive Image Features from Scale-Invarian...	[David G. Lowe]	International Journal of Computer Vision	2004	42508	Artificial Intelligence	0.017499	0.166715	0.792169	0.592247
4	Snakes: Active Contour Models	[Michael Kass, Andrew P. Witkin, Demetri Terzo...	International Journal of Computer Vision	1988	21256	Artificial Intelligence	0.082338	0.166715	0.753153	0.579585
5	Object recognition from local scale-invariant ...	[David G. Lowe]	international conference on computer vision	1999	15203	Computer Vision and Pattern Recognition	0.065958	0.128605	0.741470	0.566463
6	Support-Vector Networks	[Corinna Cortes, Vladimir Vapnik]	Machine Learning	1995	26114	Software	0.063352	0.050355	0.710757	0.542301
7	Feature extraction from faces using deformable...	[Alan L. Yuille, D. S. Cohen, Peter W. Hallinan]	computer vision and pattern recognition	1989	2624	Computer Vision and Pattern Recognition	0.646857	0.013259	0.435431	0.484066
8	Scale-space filtering	[Andrew P. Witkin]	international joint conference on artificial i...	1983	3926	Artificial Intelligence	0.135708	0.036299	0.587448	0.469001
9	Time, clocks, and the ordering of events in a ...	[Leslie Lamport]	Communications of The ACM	1978	9521	Computer Science	0.078235	0.051369	0.583986	0.452222

Figure 6: Top 10 publications by j-index.

	j-index	n_citation	reach	SJR	citation_score
j-index	1.000000	0.304625	0.509079	0.115284	0.900653
n_citation	0.304625	1.000000	0.448502	0.074681	0.124436
reach	0.509079	0.448502	1.000000	0.023788	0.090550
SJR	0.115284	0.074681	0.023788	1.000000	0.043827
citation_score	0.900653	0.124436	0.090550	0.043827	1.000000

Figure 7: Correlation matrix of j-index factors

We can see the field bias correction at work when we see Codd's seminal work is ranked 2nd, in spite of its relatively lower number of citations - 7295 as compared to the third ranked publication that has 42508 citations. Databases is certainly a field which has a significantly lower number of citations as compared to Artificial Intelligence or Machine Learning. The same applies to Leslie Lamport's paper on vector clocks. This is further substantiated by Figure 8. For example: If we refer back to Figure 1, we can see that the field that sees the most papers published is Computer Networks by some distance. Here, in Figure 8, we see that Networks has modest representation in our Top 100 papers. On the other hand, we see that Applied Mathematics sees much more representation in spite of having relatively lesser papers published.

We also see observe (in Figure 9) a more uniform distribution in terms of the year of publication in our top-10 and top-100 rankings. This is in stark contrast to the initial iteration (Figure 5B), which followed a normal-like distribution, where most of the papers were concentrated in the 90s. Here we have a much wider range starting from the 60s well into the 2000s. This shows that our year bias correction is robust.

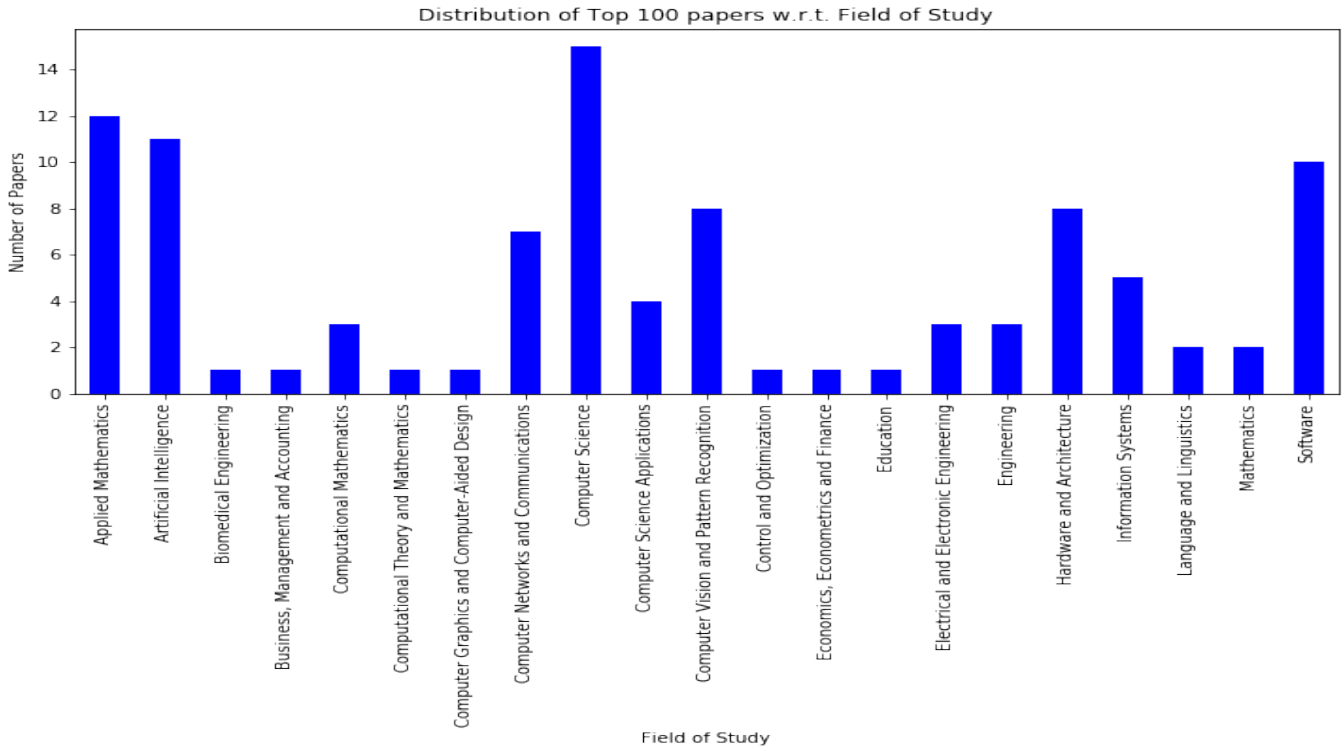


Figure 8: Distribution of Top 100 papers by j-index across fields of study

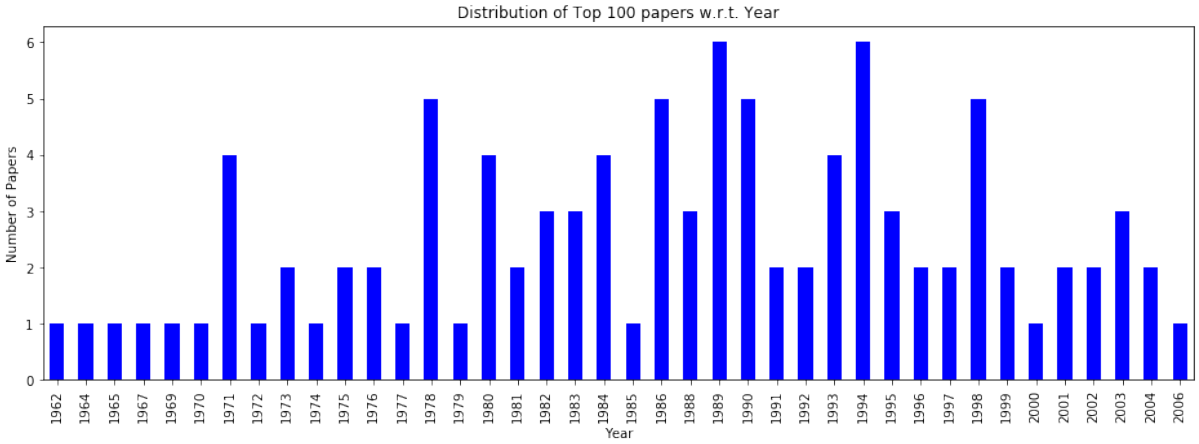


Figure 9: Distribution of Top 100 papers by j-index by the year

2) **s-index:** In Figure 10A, we again see eminent personalities in the ranking. When we look at Figure 10B, we see that our rankings have improved substantially with respect to the rank divergence metric (as outlined in the previous iteration): δ_R.

For n = 10, we achieve: $\widetilde{\delta_R}|_n = 14.5$

For n = 100, we achieve: $\widetilde{\delta_R}|_n = 176.5$

Figure 11 depicts the rank divergence distribution for the top 10,000. We see that it obeys a power law, with the researchers having their two metrics close together, being exponentially more numerous than ones with higher rank divergence. This is a very strong result which clearly highlights the performance of the s-index.

	author	s-index
0	Takeo Kanade	3.995822
1	Christos Faloutsos	3.750321
2	Jitendra Malik	3.730585
3	Wei Wang	3.609265
4	Andrew Zisserman	3.608690
5	Alex Pentland	3.572317
6	Cordelia Schmid	3.161691
7	Robert E. Schapire	3.136381
8	Pietro Perona	2.944071
9	Thomas S. Huang	2.902495

(A)

	Name of Researchers	h-index	h-index rank	s-index rank	s-index
0	Anil K. Jain	179	0	14	2.612697
1	Herbert Simon	175	1	2344	0.390766
2	Jiawei Han	162	2	90	1.540170
3	David Haussler	151	3	18	2.495822
4	Takeo Kanade	151	4	0	3.995822
5	Terrence Sejnowski	151	5	375	0.896534
6	Michael I. Jordan	148	6	19	2.480081
7	Philip S. Yu	148	7	53	1.842231
8	Scott Shenker	146	8	21	2.386200
9	Andrew Zisserman	144	9	4	3.608690

(B)

Figure 10: (A) Ranking of top-10 researchers by s-index. (B) Comparison of h-index and s-index for the top 10 authors by h-index

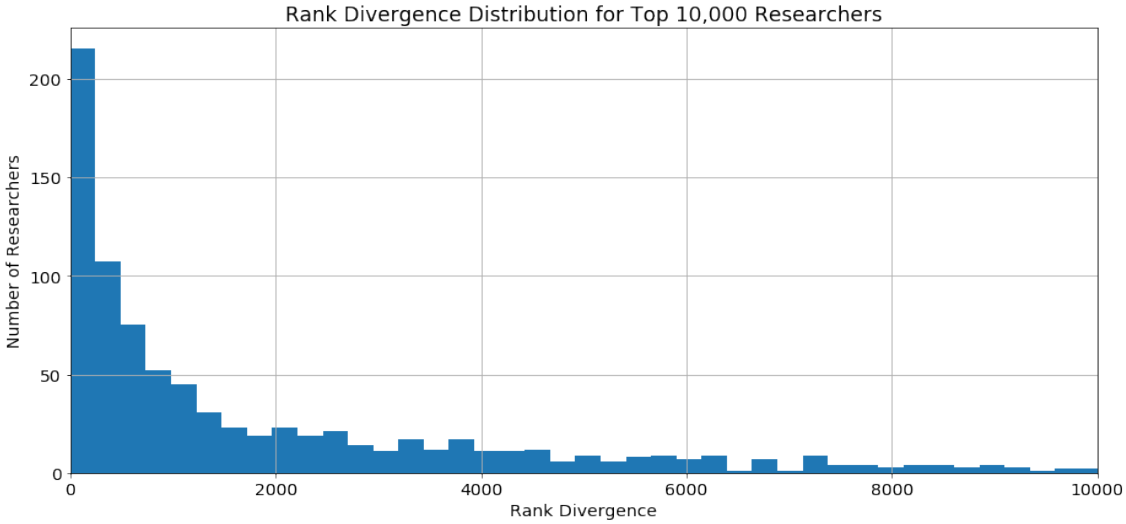


Figure 11: Rank divergence distribution for n = 10,000

9 Predictions on arXiv papers

Figure 12 shows our pipeline for predictions on the arXiv dataset. To predict the popularity of recent papers in Computer Science on arXiv, we built a machine learning model to predict the j-index of these papers. Based on a threshold for this predicted j-index, which was obtained from K-Means Clustering, we further classified these papers into two classes - "popular" and "not popular".

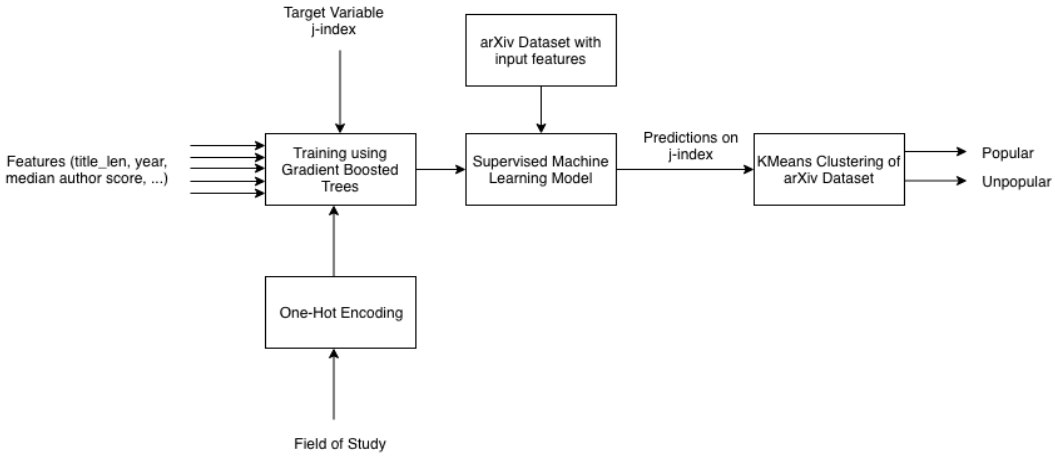


Figure 12: Prediction pipeline for arXiv publications

We trained on our master DBLP-Scimagojr joined dataset with the following features:

1. Title length
2. Abstract length
3. Number of authors
4. Field of study
5. Year of publication
6. Aggregated author s-indices: max, median and min of s-indices of the authors.

Feature engineering efforts concentrated on the following:

1. Performed Min-Max Scaling on the numerical features.
2. One-hot encoding of the categorical features such as the field of study.
3. Zero value imputation of abstract and title length when missing titles and abstracts are present.

The machine learning model we used to obtain the best accuracy was Scikit-Learn's Gradient Boosted Regressor (boosted Decision Trees) trained on a 70-30 train-test split. We leaned towards decision trees simply due to their general performance prowess and due to the presence of categorical features such as field of study. Our hyper-parameters were: num_estimators = 200 and depth = 3. The mean square error (MSRE) for our model was 0.005.

Before we ran our predictions on a subset of the arXiv dataset, we had to pre-process the data in the following manner:

1. The fields of study for these publications were listed as tags such as "cs.AI" [13]. We reconciled these 40 tags manually to the fields of study present in our master dataset.
2. Paper titles and names of authors were fuzzy-joined with our master dataset in order to account for their inconsistencies.

Thereafter, we predicted the j-indices of the entire arXiv dataset. Subsequently, we fed the predicted j-indices of these publications into K-Means clustering with $k = 2$, *random_state=0, init='k-means++'* - an attempt to find a threshold for the j-index in order to classify publications into two categories: "Popular" and "Not popular". Figure 13 shows the result. There are roughly two clusters with 10,287 papers classified as "Popular" and 27,760 papers classified as "Not Popular". In Figure 13, we can see clearly a separation boundary between the two clusters at j-index = 0.37. This boundary can be confirmed by looking at Figure 14. We can see a separation into two clusters forming at j-index = 0.37 in the bimodal distribution. The bimodal distribution only confirms that two distinct clusters exist.

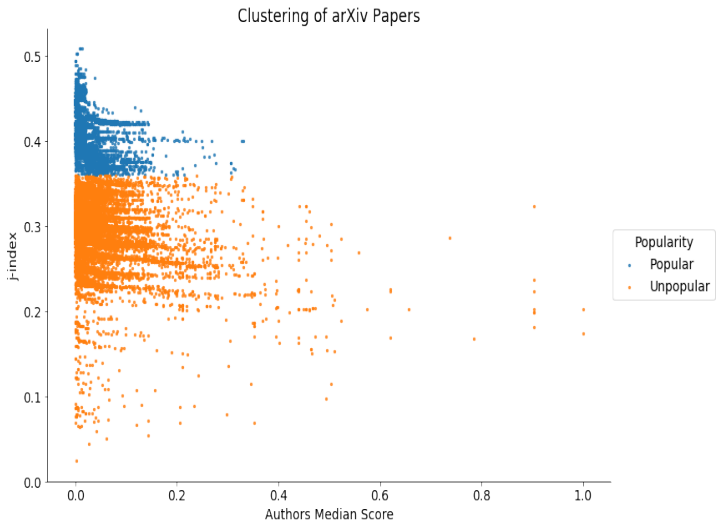


Figure 13: K-Means Clustering of arXiv publications

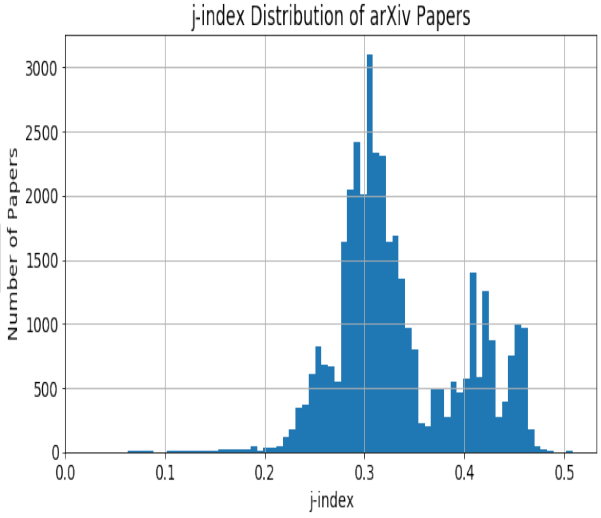


Figure 14: Bimodal distribution of j-indices of arXiv publications

Figure 15 shows a random sample of publications classified as "Popular", published after 2017. We immediately observe a spread across various fields of study - AI, Networks, Vision and Computational Linguistics. These works are clearly promising, with already a significant number of citations, in spite of their recent publication. We also observe that eminent researchers have contributed to these papers. We describe some of them below:

"Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection"[19], we see that it has 24 citations already and also features the eminent Dawn Song (UC Berkeley)[20], who has an h-index of 96, h-index rank of 121 and 47,788 citations. Similarly, we see Li Fei Fei (Stanford)[21], key contributor to Imagenet and having an h-index of 81, h-index rank of 281 and 54,089 citations, pop up in the work: "Characterizing and Improving Stability in Neural Style Transfer" [22], which has already been cited 20 times. We can also see that "On the Fine-Grained Complexity of Empirical Risk Minimization: Kernel Methods and Neural Networks"[23] has racked up 10 citations already. The paper is co-authored by Piotr Indyk (MIT)[24], who has a h-index of 70, h-index rank of 539 and has 28,172 citations.

	title	authors	year	fos	j-index	Popularity
0	Neural Network-based Graph Embedding for Cross-Platform Binary Code\n Similarity Detection	[Xiaojun Xu, Chang Liu, Qian Feng, Heng Yin, Le Song, Dawn Song]	2017	Computer Networks and Communications	0.460165	Popular
1	Lexicographic choice functions	[Arthur Van Camp, Gert de Cooman, Enrique Miranda]	2017	Artificial Intelligence	0.397581	Popular
2	Feedforward and Recurrent Neural Networks Backward Propagation and\n Hessian in Matrix Form	[Maxim Naumov]	2017	Artificial Intelligence	0.448130	Popular
3	Deep Convolutional Decision Jungle for Image Classification	[Seungryul Baek, Kwang In Kim, Tae-Kyun Kim]	2017	Computer Vision and Pattern Recognition	0.406176	Popular
4	Active Learning for Structured Prediction from Partially Labeled Data	[Mehran Khodabandeh, Zhiwei Deng, Mostafa S. Ibrahim, Shinichi Satoh, Greg Mori]	2017	Computer Vision and Pattern Recognition	0.420529	Popular
5	On the Fine-Grained Complexity of Empirical Risk Minimization: Kernel\n Methods and Neural Networks	[Arturs Backurs, Piotr Indyk, Ludwig Schmidt]	2017	Computational Theory and Mathematics	0.450381	Popular
6	Characterizing and Improving Stability in Neural Style Transfer	[Agrim Gupta, Justin Johnson, Alexandre Alahi, Li Fei-Fei]	2017	Computer Vision and Pattern Recognition	0.373163	Popular
7	When Slepian Meets Fiedler: Putting a Focus on the Graph Spectrum	[Dimitri Van De Ville, Robin Demesmaeker, Maria Giulia Preti]	2017	Artificial Intelligence	0.421744	Popular
8	Attention-based Wav2Text with Feature Transfer Learning	[Andros Tjandra, Sakriani Sakti, Satoshi Nakamura]	2017	Language and Linguistics	0.376907	Popular
9	Robust Associative Memories Naturally Occuring From Recurrent Hebbian\n Networks Under Noise	[Eliott Coyac, Vincent Gripon, Charlotte Langlais, Claude Berrou]	2017	Artificial Intelligence	0.462608	Popular

Figure 15: Prediction pipeline for arXiv publications

10 Conclusion

Our ranking does achieve a more holistic appreciation of research and researchers - with rewards along several axes apart from raw citation count such as the reach of research and quality of publication venue. At the same time, it corrects for temporal and field biases. The predictions on the arXiv dataset, especially the ones on newer publications proved to be quite reasonable, even in the absence of key features such as citation count.

Most of our challenges revolved around the limitations that each dataset had. For instance our dataset is limited to publications in Computer Science, which was a roadblock for inter-disciplinary reach calculation. However, it does have subject categories within Computer Science which we considered as different fields of study. This led to a metric that performs well, and is readily extensible to a larger dataset. Integrating disparate datasets is a certain challenge and we realized that fuzzy matching is indeed a very powerful technique. We also gained valuable experience from coming up with the reach function - especially the integration of inter-disciplinary reach. We saw how we could adjust the weights to control the quality of our overall ranking metric. We also saw the efficacy of Gradient Boosted Decision Trees as a machine learning technique and realized its applicability to datasets featuring categorical features, Finally, we found that K-Means Clustering effective in finding the threshold to separate the "Popular" from the "Not Popular".

11 References

1. h-index - https://en.wikipedia.org/wiki/H-index
2. Ranking scientists: S. N. Dorogovtsev and J. F. F. Mendes
3. EigenFactor Project - http://eigenfactor.org/index.php
4. Scimago Journal ranking calculation - https://www.scimagojr.com/SCImagoJournalRank.pdf
5. Altmetrics - https://en.wikipedia.org/wiki/Altmetrics
6. DBLP v10 dataset - https://static.aminer.cn/lab-datasets/citation/dblp.v10.zip
7. Scimago Journal ranking dataset - https://www.scimagojr.com/journalrank.php
8. fuzzymatcher library - https://github.com/RobinL/fuzzymatcher
9. Probabilistic Record Linkage - https://www.scimagojr.com/SCImagoJournalRank.pdf
10. Open Academic Graph - https://aminer.org/open-academic-graph
11. Guide2Research h-index ranking - http://www.guide2research.com/scientists/ranking
12. Kaggle - ARXIV data from 24,000+ papers - https://www.kaggle.com/neelshah18/arxivdataset
13. arXiv CS subject categories - https://arxiv.org/archive/cs
14. Pagerank - http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf
15. Herbert A. Simon - https://en.wikipedia.org/wiki/Herbert_A._Simon
16. Terrence Sejnowski-https://en.wikipedia.org/wiki/Terry_Sejnowski
17. Jiawei Han - https://en.wikipedia.org/wiki/Jiawei_Han
18. Azriel Rosenfeld - https://en.wikipedia.org/wiki/Azriel_Rosenfeld
19. Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection - https://goo.gl/EUSJkJ
20. Dawn Song - https://people.eecs.berkeley.edu/ dawnsong/
21. Li Fei Fei - http://vision.stanford.edu/feifeili/
22. Characterizing and Improving Stability in Neural Style Transfer - https://goo.gl/PT3GRN
23. On the Fine-Grained Complexity of Empirical Risk Minimization: Kernel Methods and Neural Network - https://goo.gl/GzXXJN
24. Piotr Indyk - http://people.csail.mit.edu/indyk/