Final Report: Ranking of Academic Papers

December 16, 2018

1 Introduction

Objective: Construct a ranking metric to evaluate academic papers and researchers. The h-index, although useful for ranking researchers, is not all-encompassing and its limitations have been widely discussed in the scientific community. Using data available in citation network data-set, we will model our own ranking metric based on parameters extracted from the data-set of academic papers (could be citation count, citation sentiment analysis, network granularity analysis, citation reach across multiple discipline etc.). We will also use additional data-sets and other data sources.

2 Dataset

We used the Aminer-Paper[1] dataset initially and switched to the DBLP-v7[2] as the former was too big to work on at once. Also, to calculate a more robust ranking metric, it was better to work on the whole dataset in order to account for all citations. We also found that there were a lot of sparse fields in the Aminer-Paper[1] dataset while the DBLP-v7[2] dataset had most of the fields populated.

The Dataset provides the following fields :

- *id:* ID of the paper
- *authors:* List of Authors
- *year:* Year in which paper was published
- *title:* Title of the paper
- *journal:* Journal of the paper
- *outlinks:* ID of the papers current paper is referencing

2.1 Data Cleaning

We used 2 methods to fill the missing data columns and providing extra information:

- Use of Python Scholarly library for filling out columns like Journal and year of publishing for papers having null value in these columns.
- Used arXiv API to gather information about other fields like topic of the paper.

2.2 Feature Engineering

We extracted a number of features which we found necessary to perform all the tasks. To extract all these features we first built a Citation Graph network. Basically we knew the ID of the papers referenced (outgoing edges) by a paper. So we collected all the citations/in-links (incoming edges) to each paper. Hence this resulted in a graph with the papers as nodes and features as attributes. The extracted features are as follows:

- **Inlinks of the Paper:** We built a citation graph such that every paper is attributed to all the papers referring it. This forms the list of in-links for each paper.
- **Journal Score:** We calculated the score of the journal using an external dataset obtained from Aminer Conferences[3]. We got this score by normalizing the conference H5-score present in the dataset using min-max normalization so as to bring the values within (0,1) range. But there were some conferences/journals in our dataset which were not present in that Journal Ranking Dataset. Some of these journal names were also erroneous/missing in nature. To assign the Journal Score for these unmatched/missing journals we used a Fuzzy String Matching Algorithm in the following Manner:

  - We fuzzy matched the missing journals against all the journals present in the Journals Ranking Dataset and calculated a Fuzzy Matching Ratio.
  - We assigned the journal score of the journal which gave the highest Fuzzy Match Score as the Journal Score of the unmatched/missing journal.

- **Topic:** The topic of the paper is not present in the Dataset so we fetched it using the arXiv Search API and the Journal of each paper which is a known value.

  This was calculated in the following Manner:

  - The arXiv search API lets us get the top similar papers according to our query.
  - Since we already have the list of Journals/Conferences for all papers in our Dataset, we made a arXiv API call for each Journal/Conference and retrieved this list of 10 papers. Each arXiv response provides us with a list of features for each paper which also includes the 'arXiv-primary-category' field which contains the Primary topic/domain of the paper. The assumption made here is that each journal is most likely associated with one topic. For example, CVPR (Computer Vision and Pattern Recognition) would have papers published in Computer Vision.
  - We found the modal topic/domain from this list of papers and assign it to that particular conference.
  - Then for every paper we can assign a topic using this conference's topic.
  - But here also there were few conferences which returned a NULL list. For these conferences we again used the Fuzzy String Matching method we used earlier while calculating journal score.

- **Topic Score:** We also calculated the score of each topic year wise. This was done to examine the popularity of each topic in a given year and use this feature further in our tasks.

  This was calculated in the following manner:

$$\text{Topic Score}[A][y] = \frac{\text{Number of papers of Topic A published in Year y}}{\text{Total Number of papers published in Year y}} \tag{1}$$

- **Reach Score:** We determined the reach score of each paper by determining the weighted average topic scores of unique topics of the direct inlinks of each paper. This is explained in further detail in Section 3.2.

- **Page Rank Score:** Score calculated for each paper by page rank algorithm.

- **Duration:** We also calculated the duration of paper to tone down the effect of longevity of paper by averaging out citations per year. We calculated the duration by subtracting year of publishing from the current year to determine age of paper.

2.3 External Datasets

The following external datasets were used:

- **Aminer Conference Score Data:** We extracted data from Aminer Conferences[3] for getting the score of the conferences.

- **arXiv Dataset:** We use the arXiv Dataset[4] to assign the topic/category for each paper.

2.4 Authors Dataset

Since we have to do ranking for authors we also built a dataset for Authors from our citation graph. We included the following features for the authors dataset:

- **Name of Author**

- **Paper List:** List of paper published by the author.

- **Citations:** List of length of citations for each paper.

- **Author Score:** Calculated based on the Page rank score of the papers of the Author and factors like author contribution to each paper, duration of the paper and the domain reach of the paper.

- **H index:** h-index of the author calculated based on the algorithm from wikipedia.

3 Tasks and Approaches

We did a lot of tasks for the project which cover almost all the important aspects of ranking the papers and the authors. The five main tasks are as follows:

- Top 100 ranked researchers from multiple disciplines. Comparisons of results with H-index ranking.
- Calculation of a reach score to measure inter-disciplinary influence of each paper and change the score of the paper accordingly while calculating the author score.
- Inferences on differences of our score Metric and H-index Metric.
- time analysis of paper citation count against our metric to check relevance between topic popularity and paper citation.
- identifying papers on arXiv which should become popular or important

3.1 Task 1: Paper and Author Ranking

3.1.1 Paper Ranking

In the baseline model, we used Page Rank Algorithm for calculating the score of the paper based on the citations that paper have.

For improving our ranking we incorporated 3 more factors to our score for papers. These factors are as follows:

- **Journal Score (as explained above)**
- **Reach Score:** We are giving more importance to the paper which is effecting research in many disciplines. So we add a factor to the score of papers according to the number of unique topics of the inlinks of the paper.
  Topics of paper are extracted using arXiv APIs. Then a yearly score is calculated for each topic based on the fraction of paper of that topic in that year.
  Using this score we have calculated a reach function by adding up all the scores of unique topics directly effected by that paper.
- **Longevity:** Since the Pagerank algorithm is dependent on number of citations. The papers which are published long back will tend to have more citations than recent papers. Hence we average out the number of citations over year.

  The longevity factor for a paper A is given by the following equation:

$$longevity_A = log_{10}(\frac{1 + \text{Number of Citations of A}}{\text{Duration of Paper A}}) \tag{2}$$

3.1.2 Author Ranking

The Author score is Calculated through the following steps:

- **For each paper P published by Author A, calculate the following score S and append it to a List L**:

$$S = \text{P.pagerank} * \text{Author\_Contribution\_weight} * \text{P.Journal\_Score} * \text{P.Longevity\_Score} \tag{3}$$

  where,

$$Author\_Contribution\_weight = \frac{1}{2^i} \tag{4}$$

  where, i is the index of the author A in the author List of Paper P. So the first author is given the maximum value and the values decreases as we move from the first author to the last author.

- Then we calculate the Author Score as follows:

$$Score(A) = Median(L) + log_{10}(\frac{\text{Number of Papers of Author A}}{\text{Total Number of Papers in Citation Graph}}) \tag{5}$$

For calculating the score of authors we take the median of the score of all the papers published by that author.

We also incorporated the contribution of the author by the index of author in the paper. We have made a factor of $1/2^i$.

A factor of number of papers published by the author is also added to the author score finally.

3.1.3 Results:

We found that there are 276 common authors in top 1000 authors between our metric score and H-index score. Below are the list of Top Ranked Authors according to both Our Metric and H-index metric.

’Bui Tuong Phong’	’Hector Garcia-Molina’
’M. Kirby’	’Scott Shenker’
’Neil D. McKay’,	’Jiawei Han’
’Vincent D. Park’,	’Christos Faloutsos’
’Burton H. Bloom’,	’Moni Naor’
’Josh Broch’,	’Rakesh Agrawal’
’Xuanli Lisa Xie’,	’Anil K. Jain’
’Gregory K. Wallace’,	’Thomas A. Henzinger’
’Hans Eriksson’,	’Jennifer Widom’
’James W. Layland’,	’Philip S. Yu’
Top 10 authors according to Our Metric	Top 10 authors according to H-index

We also generated topic wise results for top authors in each topic. Results for topic **Database** are as follows:

’Grady Booch’	’Hector Garcia-Molina’
’James E. Rumbaugh’	’David J. DeWitt’
’Peter Cheeseman’	’Jennifer Widom’
’Mike W. Blasgen’	’Rakesh Agrawal’
’Gerhard Weikum’	’H. V. Jagadish’
’David B. Lomet’	’Jeffrey F. Naughton’
’Hector Garcia-Molina’	’Michael J. Carey’
’H. V. Jagadish’	’Michael Stonebraker’
’Richard T. Snodgrass’	’Dan Suciu’
’Christian S. Jensen’	’Raghu Ramakrishnan’
Top 10 authors according to Our Metric	Top 10 authors according to H-index

We also generated similar results for **Machine Learning** topic. The top authors for our metric and h-index are as follows:

’Eric Bauer’	’Philip S. Yu’
’Peter Clark’	’Jiawei Han’
’Alexander Strehl’	’Jennifer Widom’
’Kamal Nigam’	’Rina Dechter’
’Philip S. Yu’	’Yoram Singer’
’Ming-Syan Chen’	’Elisa Bertino’
’Karl J. Friston’	’Manfred K. Warmuth’
’Paul M. Thompson’	’Thomas G. Dietterich’
’Jiawei Han’	’Judea Pearl’
’Arthur W. Toga’	’Thomas Eiter’
Top 10 authors according to Our Metric	Top 10 authors according to H-index

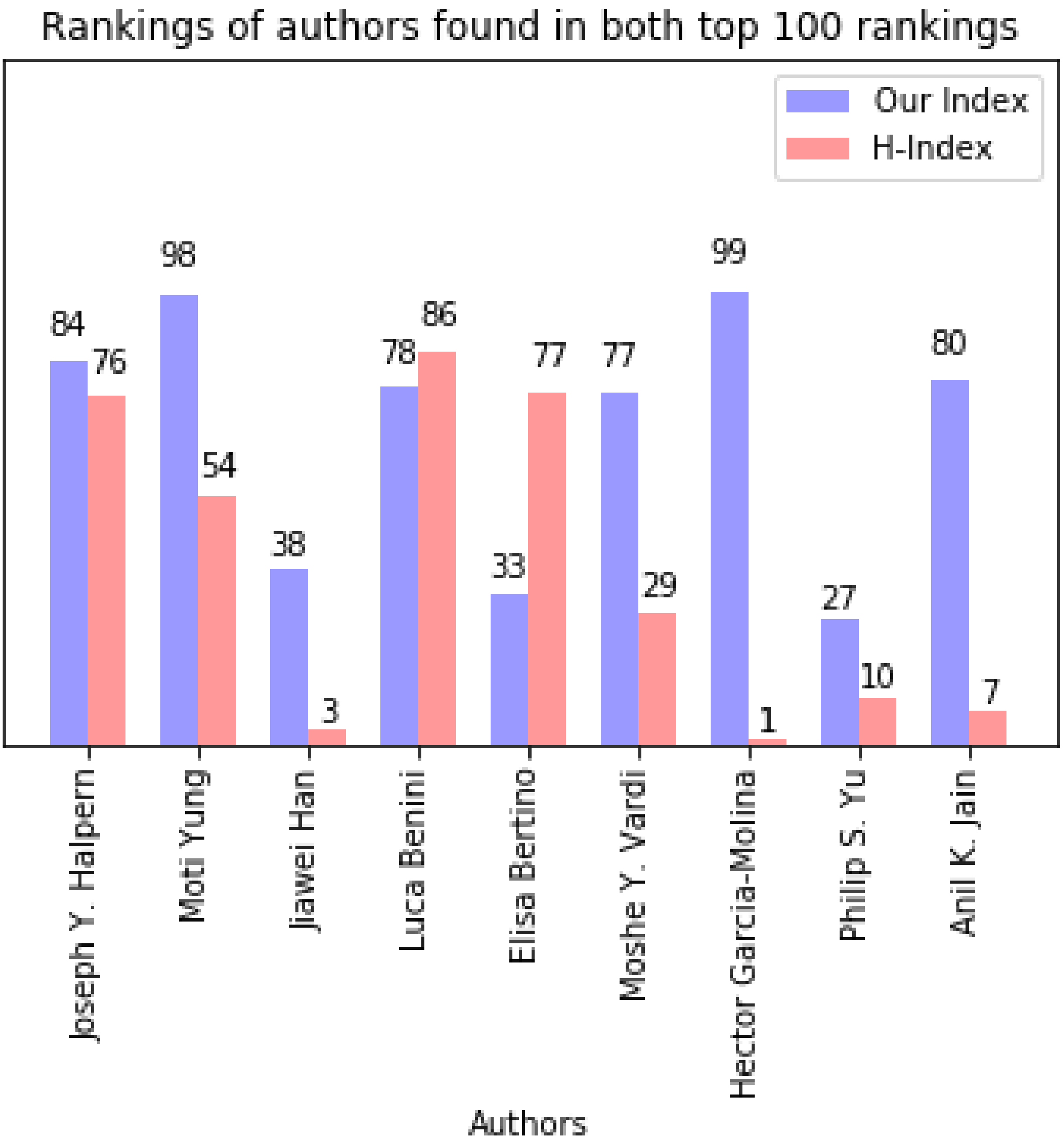


Figure 1: Common Authors in top 100 Authors of H-index and attributes

3.2 Task 2: Calculation of Reach Score

We defined reach as a factor calculated according to the number of disciplines/topics directly influenced by the paper i.e. the topics of in-links of the paper.

Our Reach score for a given paper A consists of three Components:

- **a. Domain Coverage:** The Domains/Topics impacted or "reached" by the paper A. This is calculated in the following manner:

$$\text{Domain Coverage}_A = \frac{\text{Total number of unique topics from citing papers including A's topic}}{\text{Total Number of Unique Topics}} \tag{6}$$

- **b. Weighted Sum:** Weighted Sum of Citing paper's Number of Citations(Number of inlinks) with the Yearly Topic Score as their weights. The Yearly Topic score always lies in the range [0,1] and helps us in assigning a value to each citing paper's contribution according to the Impact and popularity of their respective domains.

$$\text{Weighted Sum}_A = \sum_{i \in inlinks_A} Topic\_Score[Topic_i][Year_i] * Number\_Citations_i \tag{7}$$

- **c. Normalizing Factor:** It is the sume of the number of citations of all the papers citing paper A. This is to ensure that the reach score remains in the range [0,1].

$$\text{Normalizing Factor}_A = \sum_{i \in inlinks_A} Number\_Citations_i \tag{8}$$

The final reach score of the paper A is given by the following equation,

$$\text{reach score}_A = 1 + \frac{\text{Domain Coverage}_A * \text{Weighted Sum}_A}{\text{Normalizing Factor}_A} \tag{9}$$

We have also calculated **year wise topic score** according to the fraction of papers belonging to the topic in a particular year. The trend for Data Structure and Computer Vision topic are as follows:

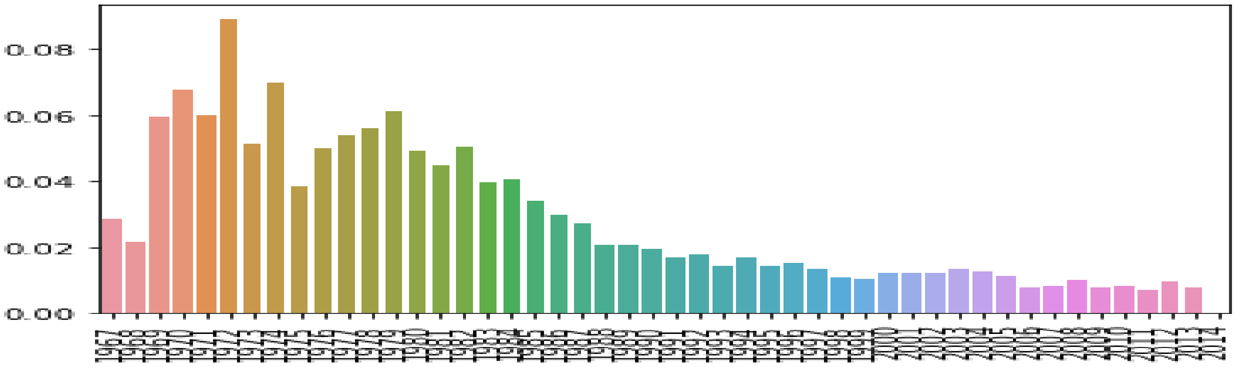


Figure 2: Trend for Data Structure Topic

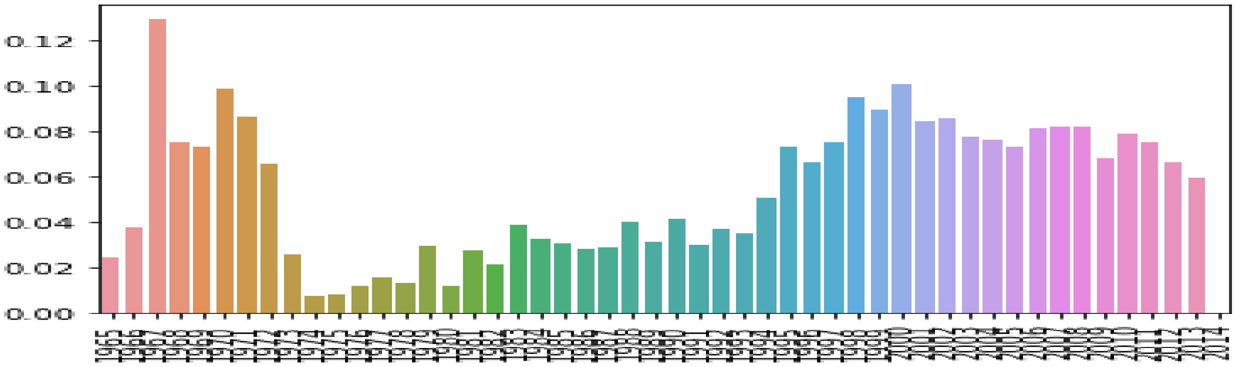


Figure 3: Trend for Computer Vision Topic

3.3 Task 3: Differences in our metric and H-index

There are significant differences between our metric and H-index. Some of the differences are as follows:

- H-index take only the number of citations into consideration whereas in our metric the paper having low number of citations also have high score.
- Our metric also incorporate the impact factor/score of conference for calculating the final score of the paper. Since some of the conferences have really high standards for accepting the papers, Quality of Conference is a important factor that H-index misses out.

- In our metrics we have also considered the contribution of author as a factor. This factor is decided by their index in the authors list. This is not considered in H-index.
- Our metric also tone out the longevity of the papers. Basically the papers that published long back will definitively have more citation than those papers that are relatively newer.
- we are also adding a factor of number of papers published by an author.

3.3.1 Results:

We got the authors with maximum differences in ranks between h-index ranking and our ranking. We saw that most of the authors in this list have 0 h-index and hence h-index ranking is not able to distinguish between these authors. But our algorithm is ranking these authors properly on the basis of factors like Journal Score Reach Score etc. The list for the max difference in ranking is as follows:

our Rank	H-index rank	Name of Author	H-index
70308	1227777	'Roozbeh Izadi-Zamanabadi'	0
57999	1215714	'Iain Woodhouse'	0
39732	1198224	'Hongxing Wei'	0
29539	1188733	'Vahid Meghdadi'	0
41188	1200408	'Robert D. Finn'	0
61567	1220972	'Patrick Pons'	0
69858	1229550	'Philip J. B. Jackson'	0
41743	1201848	'Maurizio Migliaccio'	0
60373	1220509	'Noriaki Miyazaki'	0
68910	1229060	'Gerald M. Maggiora'	0
22859	1183066	'Arnold Beckmann'	0

3.4 Task 4: Time analysis of paper citation count

3.4.1 Approach

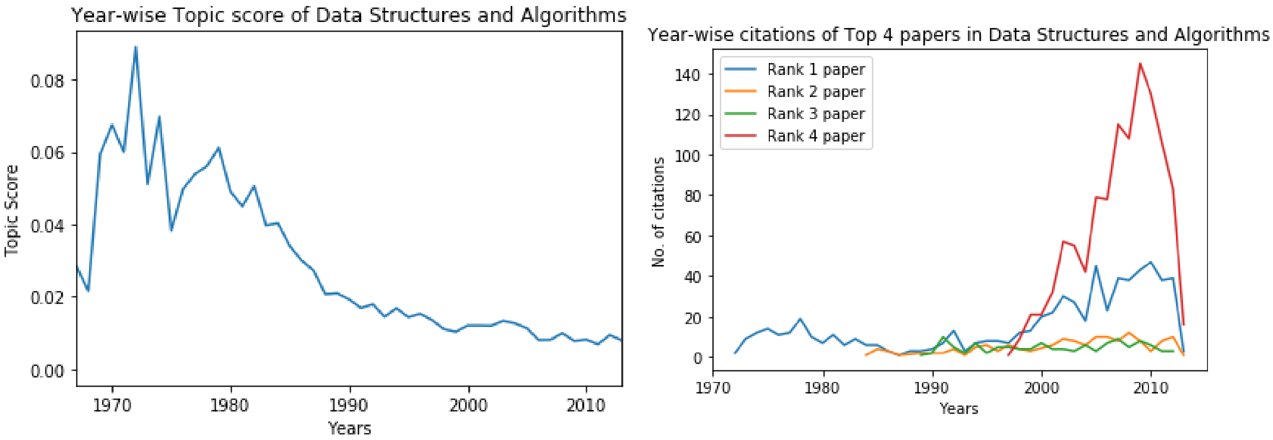
Time analysis was done using year wise topic score (described in Sec 2.2). We used this metric to analyze each topic's popularity over the years and compare it against the top papers in the corresponding topic to see if there is a bias that might be boosting the paper's ranking.

For the latter part of the comparison, we calculated the number of citations per year from the year of publication for each paper in order to visualize the comparison through juxtaposition. This comparison can be seen in the next section.

3.4.2 Results

The images in Fig. 4 and Fig. 5 show the results for two topics namely, Data Structures and Algorithms and Accelerator Physics.

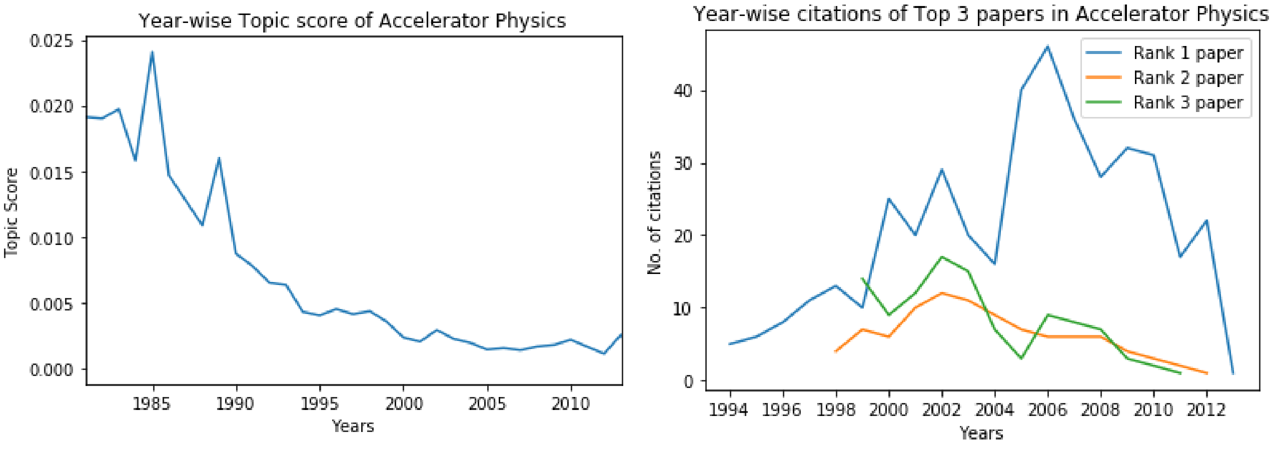
By analyzing the trends we figure out that our metric is already taking care that the top ranked papers are not having citations just because the topic got popular. Our metric works because the topic score factor is calculated for each year and hence normalizes any effect of popularity.



(a) Year-wise Topic scores

(b) Year-wise citations of Top 4 papers

Figure 4: Topic - Data Structures and Algorithms



(a) Year-wise Topic scores

(b) Year-wise citations of Top 3 papers

Figure 5: Topic - Accelerator Physics

3.5 Task 5: Predicting popularity of arXiv Papers

3.5.1 Challenges

- Since we are working with a dataset that only has a subset of all authors present in the world, predicting the popularity of any paper published by an author who is not present in our dataset will be slightly less reliable as we do not have a ranking score for them. To overcome this, we consider only arXiv papers published by authors present in our dataset.

- We also do not have the topic scores of all the topics from 2014 onwards and hence we needed to compute that prior to this task.

3.5.2 Approach

In our dataset, we have data from the year 1936 to 2013. In order to predict whether a paper will become popular or not, we considered a set of 20,000 authors who have published papers in the latest year, 2013 in our dataset. We split these authors in to two halves, train and test.

Since we do not have topic popularity scores for the years from 2014 onwards, we extrapolated these scores for each topic by an exponentially weighted moving average forecast (EWMA).

The idea then is to train a binary classifier which takes a paper and predicts whether it will become popular (1) or not (0). The features we used for this classification task are

- Journal Score
- Topic Score
- Topic (as a categorical feature)
- Author score based on our metric

To annotate each paper on whether it is popular or not, we used a threshold of 0.3 times the maximum citations among all papers in that topic. So, any paper above the threshold is denoted as popular.

Training Phase - We used a LightGBM classifier on the training dataset comprising of around 100,000 papers and the four features mentioned above.

Testing Phase - For the authors in our testing set, we then fetch all their papers on arXiv from 2014 onwards using the arXiv API. We calculated the same set of features for these papers and used our model to predict whether they will become popular or not.

3.5.3 Results

Sample papers that may become popular from our model:

- Mirco Ravanelli, Benjamin Elizalde, Karl Ni, Gerald Friedland - Audio Concept Classification with Hierarchical Deep Neural Networks

- Julien Perret, Maurizio Gribaudi, Marc Barthelemy - Roads and cities of 18th century France

- Francesco Zanlungo, Zeynep Yucel, Takayuki Kanda - The effect of social roles on group behaviour

The characteristics of these papers are that they have either been published in a reputed journal or have obtained a good number of citations within a few months of publications. This shows that our model seems robust in predicting popular papers. Only about 5% of the testing dataset was predicted to be popular and this was in line with our training dataset as well.

4 Observation and Conclusions

There are some keen observations that we can share:

- As shown from the results of Tasks 3 and 4 we have come up with a metric which is able to tackle all the disadvantages of h_index, namely,
  - H_index does not take field dependent factors into account which we have taken while calculating our reach score as well as the journal score.
  - The h-index discards the information contained in author placement in the authors' list, which we have also accounted for using the Author_contribution_weight feature.

  - The h-index is a natural number that reduces its discriminatory power. Our Metric is a real number and thus have higher discriminatory power than h_index.

- As shown from Task 1 we see that although our metric is very different from H_index it still bears some similarities between them as there are 9 common authors in the Top 100 lists of both metrics. So we safely say that although we have tried to get rid of the disadvantages of h_index in our metric we have tried to keep it's advantages as well.

- There were some authors who have the same exact names. But there was no unique identifier presemt in the Database to Segregate them. For Example, Author 'Wei Wang' has Papers in 100 different topics which is highly improbable. There were several other examples of this kind. We feel that author's should also have a unique id just like papers so as to properly identify them and remove any inconsistensies which were caused due to this problem.

References

[1] https://www.openacademic.ai/oag/

[2] http://aminer.org/open-academic-graph

[3] https://aminer.org/ranks/conf

[4] https://arXiv.org/