ALTEGRAD challenge Fall 2020: H-index Prediction

Aymane Berradi, Taoufik Aghris, Badr Laajaj
M2 Data Sciences Ecole Polytechnique
aymane.berradi@polytechnique.edu, taoufik.aghris@polytechnique.edu,
badr.laajaj@polytechnique.edu

Abstract

The *IJCAI–21 Proceedings* will be printed from electronic manuscripts submitted by the authors. The electronic manuscript will also be included in the online version of the proceedings. This paper provides the style instructions.

1 Introduction

1.1 Context and motivation

In this report, we summarize the different research efforts and experiments that we conducted on ALTEGRAD challenge, which is hosted on Kaggle, concerning the prediction of the h-index of research papers authors. We are dealing with a regression problem using texts of articles abstracts and a graph network that matches the authors that co-authored the same papers. The target variable of this problem is the h-index which can be defined as the maximum number of published papers h that have each been cited at least h times. This metric measures the performance of an author with regards to the number of citations of his publications. During the challenge, we worked on a large-scale dataset, and followed the regression problem pipeline, that is we went from cleaning and preprocessing the data, features engineering to choosing the right regression models and tuning their parameters, and finally measuring their performances to choose the best ones for the competition.

1.2 Evaluation Metric

When it comes to measuring the performance of our models, the challenge asses the performance of our models using the **mean absolute error (MAE)**, which is the average of the absolute errors over the dataset. We can express it as:

$$MAE = \frac{1}{N}\sum_{1}^{N}|y_i^* - y_i| \tag{1}$$

Where N is the number of observations in the evaluated dataset, y_i^* the predicted value (h-index) for the ith observation and y_i its actual values. During this competition, we are trying to reduce the errors our prediction, thus we are looking for small MAE values. Finally, after evaluation our model on our train and validation sets, we predict on tests dataset, which we do not know its h-index values, and then submit the csv file into Kaggle the get a real evaluation of our performance.

2 Data Analysis

2.1 Graph Data

We are dealing with unweighted and undirected graph that models the co-authorship network, such that vertices correspond to authors where two authors (vertices) are linked by an edge if they have co-authored at least one paper, we have 231 239 nodes and 1 777 338 edges.
Let's see an example of what our graph looks like for a particular author:

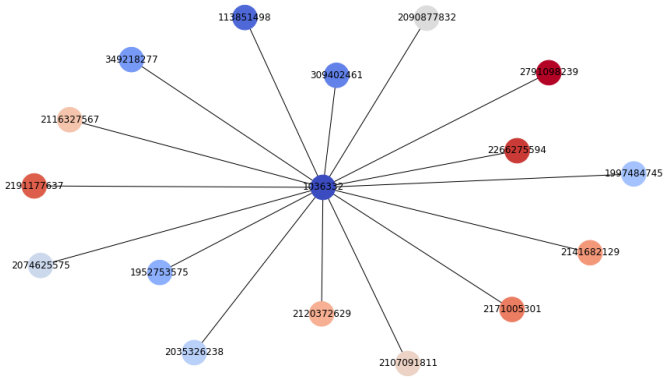

Figure 1: Subgraph of the co-authorship network

In the above example, we notice that the author with $id = 1036332$ has co-authored with 16 authors.
We also observe that there are no isolated nodes in the graph, which means that the author co-authored with at least one author. Moreover, when we examine the `author_papers.txt` document, we realize that even if there are two connected nodes by an edge, we didn't find any single common paper between them and this is due to the fact that the papers set per author contain the top cited papers.
To illustrate what is stated before, we take two nodes with degree 1, which means we should find at least one co-authored paper between them in the `author_papers.txt`.

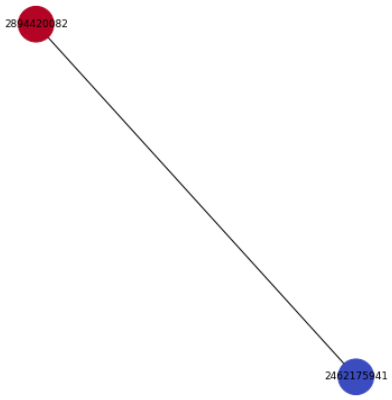

Figure 2: Example of nodes with $degree = 2$

Let's have a look on the papers set for each author:

```
2894420082 : ['2140685354', '2151997344', '2148203223', '2146788893', '2128168975', '2166624128', '2169672506', '2053027574', '1528077854', '1600183845']
2462175941 : ['2130593958', '2079543149', '2168126888', '2159470470', '2039731046', '2144677702', '1568614804', '2115733366', '2273316012', '2295318563']
```

We can figure out easily that there is no common paper between these 2 nodes, another observation is that the papers set for each author in `author_papers.txt` is made of single and co-authored papers.

```
42782019 : [2, 6, 3, 4, 3, 2, 1, 1, 2, 4]
```

In the below example, the author that have $id = 42782019$ wrote 2 papers by his own and co-authored in 8 papers.

2.2 Text Data

For the text data, we have at our disposal a set of papers with their abstracts, there is a total of 1 056 539 papers that are present in `abstract.txt` file, however we have only 1 056 298 papers that have abstracts, so 241 papers have missing abstracts, we noticed also that there is more than one language in the abstracts set.

2.3 Train and test data

The training data contained 23 124 labeled authors, each sample have the author ID with the corresponding h_index, each author has his top-cited papers in `author_papers.txt`. For the test data, we have 208 115 unlabeled authors, the goal is to use the feature extracted from the graph and textual information to predict their corresponding h_index.

3 Text Preprocessing

As we are dealing with papers abstracts which represent text data, we need to clean it from white spaces, English stops words and to preprocess it so we can have same representation of the same word, as example for the verb "write" we can find many formats like "writes", "writing" so we need to normalize it to the same original word and thus have at the end the same vectorial representation of that word. This helps us build a strong corpus that represents main vocabularies of the abstracts and reduce the noise in the text representations. The provided data, `abstracts.txt`, is a .txt file where each paper ID is matched with an inverted index which is a dictionary where the keys are the words, and their values are lists of their corresponding positions in the abstracts. We can summarize this part of data preparation in the following steps:

- Read text from files using `utf-8` encoding to get almost all words present in the abstracts ;
- Loop over each paper id and extract the words in its inverted index ;
- We check if the lowercase of each word is alphabetic and neither a stop word nor empty ;
- If true, we lemmatize the word which consists on changing the inflected form of a word to its original format (as explained in example above with the verb write) so it can be analyzed as a single vocabulary. To do so, we used the **WordNetLemmatizer** from the **nltk** package. This lemmatizer is a pretrained model that inputs the word and its tagging (verb, noun, adjective and adverb) so it can find a meaningful base form based on the context. To get the tag of a word, we built a function `get_wordnet_pos()` that uses **nltk.pos_tag**, a pretrained tagging model from **nltk** package, and output a wordnet tagging as a parameter for our **WordNetLemmatizer** ;
- Now that we have our lemmatized word, we put it in its right positions in our abstract list using the inverted index values ;
- Finally, we output a dictionary with papers ID as keys and list of processed words in their right positions in the abstract.

4 Feature Engineering

In order to describe the structures inherent in our data and explain the problem at hand, we update the dependent variable using logarithm transformation and we extract some meaningful features from the text and authors data.

4.1 Features Extraction

To better understand our new features, we define the following elements:

- Corpus: is a list of unique tokens cited in all documents;
- Author_docs_occurrence: For each author we count the occurrences of his papers in the whole file `authors_papers.txt` ;
- Abstract_ratio: it is the ratio of tokens set (unique items) of each author's abstract to its length. (formula) ;
- Corpus_ratio: It describes the ratio of the tokens set in an abstract to the corpus's length for each author.

Based on the elements above we create the next variables:

- Single_doc: It is the number of documents that are not co-authored for each author ;
- Docs_number: it's the number of documents written by each author ;

- **Max**, **mean** and **min** of Author_docs_occurrence of each author ;
- **Max**, **mean** and **min** of Corpus_ratio of each author ;
- **Max**, **mean** and **min** of Abstract_ratio of each author ;
- Co_authored_sum: The sum of Author_docs_occurence that are co-authored (occurrence is different to 1) for each author.

4.2 Log transformation

We use the function $y \rightarrow log(y+1)$ to guarantee that h_index is strictly positive. This transformation also helps to handle skewed data, reduce the effect of outliers. The 1 is added to avoid infinite values for h_index equals to 0.

Section Headings

Print section headings in 12-point bold type in the style shown in these instructions. Leave a blank space of approximately 10 points above and 4 points below section headings. Number sections with arabic numerals.

Subsection Headings

Print subsection headings in 11-point bold type. Leave a blank space of approximately 8 points above and 3 points below subsection headings. Number subsections with the section number and the subsection number (in arabic numerals) separated by a period.

Subsubsection Headings

Print subsubsection headings in 10-point bold type. Leave a blank space of approximately 6 points above subsubsection headings. Do not number subsubsections.

Titled paragraphs. You should use titled paragraphs if and only if the title covers exactly one paragraph. Such paragraphs should be separated from the preceding content by at least 3pt, and no more than 6pt. The title should be in 10pt bold font and ended with a period. After that, a 1em horizontal space should follow the title before the paragraph's text.

In LaTeX titled paragraphs should be typeset using

```
\paragraph{Title.} text.
```

Acknowledgements

You may include an unnumbered acknowledgments section, including acknowledgments of help from colleagues, financial support, and permission to publish. If present, acknowledgements must be in a dedicated, unnumbered section appearing after all regular sections but before any appendices or references.

Use

```
\section*{Acknowledgements})
```

to typeset the acknowledgements section in LaTeX.

Appendices

Any appendices directly follow the text and look like sections, except that they are numbered with capital letters instead of arabic numerals. See this document for an example.

References

The references section is headed "References", printed in the same style as a section heading but without a number. A sample list of references is given at the end of these instructions. Use a consistent format for references. The reference list should not include publicly unavailable work.

4.3 Citations

Citations within the text should include the author's last name and the year of publication, for example [Gottlob, 1992]. Append lowercase letters to the year in cases of ambiguity. Treat multiple authors as in the following examples: [Abelson *et al.*, 1985] or [Baumgartner *et al.*, 2001] (for more than two authors) and [Brachman and Schmolze, 1985] (for two authors). If the author portion of a citation is obvious, omit it, e.g., Nebel [2000]. Collapse multiple citations as follows: [Gottlob *et al.*, 2002; Levesque, 1984a].

4.4 Footnotes

Place footnotes at the bottom of the page in a 9-point font. Refer to them with superscript numbers.[1] Separate them from the text by a short line.[2] Avoid footnotes as much as possible; they interrupt the flow of the text.

5 Illustrations

Place all illustrations (figures, drawings, tables, and photographs) throughout the paper at the places where they are first discussed, rather than at the end of the paper.

They should be floated to the top (preferred) or bottom of the page, unless they are an integral part of your narrative flow. When placed at the bottom or top of a page, illustrations may run across both columns, but not when they appear inline.

Illustrations must be rendered electronically or scanned and placed directly in your document. They should be cropped outside latex, otherwise portions of the image could reappear during the post-processing of your paper. All illustrations should be understandable when printed in black and white, albeit you can use colors to enhance them. Line weights should be 1/2-point or thicker. Avoid screens and superimposing type on patterns, as these effects may not reproduce well.

Number illustrations sequentially. Use references of the following form: Figure 1, Table 2, etc. Place illustration numbers and captions under illustrations. Leave a margin of 1/4-inch around the area covered by the illustration and caption. Use 9-point type for captions, labels, and other text in illustrations. Captions should always appear below the illustration.

6 Tables

Tables are considered illustrations containing data. Therefore, they should also appear floated to the top (preferably) or bottom of the page, and with the captions below them.

If you are using LaTeX, you should use the `booktabs` package, because it produces better tables than the standard

[1] This is how your footnotes should appear.

[2] Note the line separating these footnotes from the text.

Scenario	δ	Runtime
Paris	0.1s	13.65ms
Paris	0.2s	0.01ms
New York	0.1s	92.50ms
Singapore	0.1s	33.33ms
Singapore	0.2s	23.01ms

Table 1: Latex default table

Scenario	δ (s)	Runtime (ms)
Paris	0.1	13.65
	0.2	0.01
New York	0.1	92.50
Singapore	0.1	33.33
	0.2	23.01

Table 2: Booktabs table

ones. Compare Tables 1 and 2. The latter is clearly more readable for three reasons:

1. The styling is better thanks to using the `booktabs` rulers instead of the default ones.
2. Numeric columns are right-aligned, making it easier to compare the numbers. Make sure to also right-align the corresponding headers, and to use the same precision for all numbers.
3. We avoid unnecessary repetition, both between lines (no need to repeat the scenario name in this case) as well as in the content (units can be shown in the column header).

7 Formulas

IJCAI's two-column format makes it difficult to typeset long formulas. A usual temptation is to reduce the size of the formula by using the `small` or `tiny` sizes. This doesn't work correctly with the current LaTeX versions, breaking the line spacing of the preceding paragraphs and title, as well as the equation number sizes. The following equation demonstrates the effects (notice that this entire paragraph looks badly formatted):

$$x = \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i \quad (2)$$

Reducing formula sizes this way is strictly forbidden. We **strongly** recommend authors to split formulas in multiple lines when they don't fit in a single line. This is the easiest approach to typeset those formulas and provides the most readable output

$$x = \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \\ + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i \quad (3)$$

If a line is just slightly longer than the column width, you may use the `resizebox` environment on that equation. The result looks better and doesn't interfere with the paragraph's line spacing:

$$x = \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i \quad (4)$$

This last solution may have to be adapted if you use different equation environments, but it can generally be made to work. Please notice that in any case:

- Equation numbers must be in the same font and size than the main text (10pt).
- Your formula's main symbols should not be smaller than small text (9pt).

For instance, the formula

$$x = \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j + \prod_{i=1}^{n}\sum_{j=1}^{n} j_i + \prod_{i=1}^{n}\sum_{j=1}^{n} i_j \quad (5)$$

would not be acceptable because the text is too small.

8 Examples, Definitions, Theorems and Similar

Examples, definitions, theorems, corollaries and similar must be written in their own paragraph. The paragraph must be separated by at least 2pt and no more than 5pt from the preceding and succeeding paragraphs. They must begin with the kind of item written in 10pt bold font followed by their number (e.g.: Theorem 1), optionally followed by a title/summary between parentheses in non-bold font and ended with a period. After that the main body of the item follows, written in 10 pt italics font (see below for examples).

In LaTeX We strongly recommend you to define environments for your examples, definitions, propositions, lemmas, corollaries and similar. This can be done in your LaTeX preamble using `\newtheorem` – see the source of this document for examples. Numbering for these items must be global, not per-section (e.g.: Theorem 1 instead of Theorem 6.1).

Example 1 (How to write an example)**.** *Examples should be written using the example environment defined in this template.*

Theorem 1. *This is an example of an untitled theorem.*

You may also include a title or description using these environments as shown in the following theorem.

Theorem 2 (A titled theorem)**.** *This is an example of a titled theorem.*

9 Proofs

Proofs must be written in their own paragraph separated by at least 2pt and no more than 5pt from the preceding and succeeding paragraphs. Proof paragraphs should start with the keyword "Proof." in 10pt italics font. After that the proof follows in regular 10pt font. At the end of the proof, an unfilled square symbol (qed) marks the end of the proof.

In LaTeX proofs should be typeset using the `\proof` environment.

Proof. This paragraph is an example of how a proof looks like using the `\proof` environment. □

Algorithm 1 Example algorithm

Input: Your algorithm's input
Parameter: Optional list of parameters
Output: Your algorithm's output

```
Let t = 0.
while condition do
    Do some action.
    if conditional then
        Perform task A.
    else
        Perform task B.
    end if
end while
return solution
```

10 Algorithms and Listings

Algorithms and listings are a special kind of figures. Like all illustrations, they should appear floated to the top (preferably) or bottom of the page. However, their caption should appear in the header, left-justified and enclosed between horizontal lines, as shown in Algorithm 1. The algorithm body should be terminated with another horizontal line. It is up to the authors to decide whether to show line numbers or not, how to format comments, etc.

In LaTeX algorithms may be typeset using the `algorithm` and `algorithmic` packages, but you can also use one of the many other packages for the task.

Acknowledgments

The preparation of these instructions and the LaTeX and BibTeX files that implement them was supported by Schlumberger Palo Alto Research, AT&T Bell Laboratories, and Morgan Kaufmann Publishers. Preparation of the Microsoft Word file was supported by IJCAI. An early version of this document was created by Shirley Jowell and Peter F. Patel-Schneider. It was subsequently modified by Jennifer Ballentine and Thomas Dean, Bernhard Nebel, Daniel Pagenstecher, Kurt Steinkraus, Toby Walsh and Carles Sierra. The current version has been prepared by Marc Pujol-Gonzalez and Francisco Cruz-Mencia.

A LaTeX and Word Style Files

The LaTeX and Word style files are available on the IJCAI–21 website, https://www.ijcai21.org/. These style files implement the formatting instructions in this document.

The LaTeX files are `ijcai21.sty` and `ijcai21.tex`, and the BibTeX files are `named.bst` and `ijcai21.bib`. The LaTeX style file is for version 2e of LaTeX, and the BibTeX style file is for version 0.99c of BibTeX (*not* version 0.98i). The `ijcai21.sty` style differs from the `ijcai20.sty` file used for IJCAI–PRICAI–20.

The Microsoft Word style file consists of a single file, `ijcai21.doc`. This template differs from the one used for IJCAI–PRICAI–20.

These Microsoft Word and LaTeX files contain the source of the present document and may serve as a formatting sample.

Further information on using these styles for the preparation of papers for IJCAI–21 can be obtained by contacting `pcchair@ijcai-21.org`.

References

[Abelson *et al.*, 1985] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Massachusetts, 1985.

[Baumgartner *et al.*, 2001] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Databases*, pages 119–128, Rome, Italy, September 2001. Morgan Kaufmann.

[Brachman and Schmolze, 1985] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[Gottlob *et al.*, 2002] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, May 2002.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[Levesque, 1984a] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, July 1984.

[Levesque, 1984b] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pages 198–202, Austin, Texas, August 1984. American Association for Artificial Intelligence.

[Nebel, 2000] Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research*, 12:271–315, 2000.