COMP_542DL_exams

April 16, 2021

1 Data Programming - COMP542-DL: Final Exams

1.1 Lecturer: Ioannis Partalas

1.2 Semester: Spring 2021

2 Exercise 1 (20%)

Create a matrix X of size $m \times m$ where approximately 80% of the values are zeros and the rest are random integers between 1 and 30. Then factorize this matrix using `Singular Value Decomposition`:

$$SVD(X) = U\Sigma V^T$$

Then truncate the first k dimensions of the left singular matrix and multiply with the corresponding truncation of the diagonal matrix.

$$E = U_{:,1:k}\Sigma_{1:k}^{\alpha}$$

where $\alpha = 0.5$.

3 Exercise 2 (40%)

In this exercise you will explore data from OECD.

Specifically you will explore the following indicators:

- Tax on personal income (`oecd_tax_income.csv`).
- Tax on corporate profits (`oecd_corporate_profit.csv`).
- Tax on property (`oecd_tax_on_property.csv`).
- Unemployment rate (`oecd_unemployment.csv`).

You will have to complete the following activities:

1. Remove any average measurements from the datasets such as OECD average (OAVG) or EU averages.
2. Drop the column `FLAG_CODES` from all the datasets.
3. Plot for few selected countries (5) the tax on personal income across the years.
4. Plot the distributions of the tax on personal income for the full dataset.
5. Plot the same distributions per continent (Europe, Asia, America).
6. Find the average and median tax on corporate profits per country.
7. Calculate the ratio of the tax on personal income versus the tax on property for each country. Then plot across the years for few selected countries (5).

8. Calculate the unemployment rate per country per year.
9. Do the same per continent.
10. Standardize the unemployment rate values per continent. 11 Rank the countries in decreasing order of the average unemployment rate for the years 2011-2015.

4 Exercise 3 (40%)

In this exercise you will have to develop a class that implements the BM25 retrieval function. More specifically, given a query document Q and a document D from a collection of documents C, the BM25 score is computed as follows:

$$BM25(Q, D) = \sum_{i=1}^{|Q|} IDF(q_i) \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1(1 - b + b\frac{|D|}{avgdl})}$$

where index i goes over each term (q_i) of query Q, $|D|$ is the length of the document D (in terms of tokens) and $avgdl$ is the average length of documents in the collection. k_1 and b are free parameters to be specified.

$IDF(q_i)$ is the Inverse Document Frequency of term q_i and is calculated as follows for a collection of documents C:

$$IDF(q_i) = \log\left(\frac{N - n_{q_i} + 0.5}{n_{q_i} + 0.5} + 1\right)$$

where $N = |C|$ is the total number of documents in the collection and n_{q_i} is the number of documents in the collection containing the term q_i.

4.1 Details

The class you will implement should have a method `fit()` that will calculate the appropriate statistics in the collection of documents C.

Another method called `score()` that will calculate the $BM25$ score of a query document Q against all the documents in the collection. Then, these scores should be used to rank the documents with decreasing score and keep the top 10 documents. Then, you will have to display the query document and the top 10 most similar ones.

Note also the following:

- As pre-processing steps remove punctuation and special characters.
- For tokenization you can just use whitespace.
- For the free parameters use $k = 1.2$ and $b = 0.75$.