MERC/IST 2013

Extracção e Análise de Dados na Web

Lab 05: Learning to Rank

The goal of this exercise is to implement a simple Learning to Rank (L2R) solution, using a linear regression.

To achieve this, you are given the following material:

a) A function to compute a linear regression

The following function takes as input a list of tuples containing the input values for the regression ($X = x_1, x_2, \ldots, x_N$), where each x_i is a D-dimensional tuple, and a list containing the expected output values ($y = y_1, y_2, \ldots, y_N$). The function returns a tuple containing the coefficients $w = w_1, w_2, \ldots, w_D, w_{D+1}$ of the equation:

$$y_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_D x_{iD} + w_{D+1}$$

where $1 \leq i \leq N$.

```
import numpy as np
from numpy import linalg

def lregression(X,y):
    l = len(y)
    A = np.vstack([np.array(X).T, np.ones(l)])
    return linalg.lstsq(A.T,y)[0]
```

Example of a linear regression where X is 2-dimensional:

```
X = [(0,3), (2,3), (2.5,3.6), (4,4.8)]
y = [7.3, 8.6, 8.5, 9.0]
w = lregression(X,y)
print w
```

b) A training dataset

The file `aula05_features.txt` contains a dataset of ranking signals (or *features*) that can be used to learn a ranking function. Each line contains the following values, for a document-query pair:

Column 1 The BM25 similarity value between the query and the document;

Column 2 The Cosine similarity value between the query and the document;

Column 3 The PageRank value of the document;

Column 4 The expected output value of the ranking function (1 if the document is relevant, 0 otherwise).

1

Assume you are using a linear combination of three signals (*BM25*, *Cosine similarity*, and *PageRank*) to rank the documents. Using the above mentioned function and dataset compute the coefficients needed for the combination.

2

Using Whoosh, implement a simple search engine that applies the coefficients found in the previous exercise to rank the documents.

> **Note:** Since Whoosh can only compute one similarity value when a search is performed, you will need to search twice to get the required signals (one using BM25 and another using TF-IDF).

3

As in previous lessons, implement a script that reads the `aula03_queries.txt` file and, for each query, executes the corresponding search and measures the precision, recall and F1 of the results. Use the ranking combination implemented in the previous item.

The script should print out each query and its evaluation values and a final average value for each measure.