

## Description

### Learning to Rank (50 points)

Learning-to-rank is considered to be one of the most important techniques in modern search engines and recommender systems. In this task, we will go through some basic concepts and models commonly used in learning-to-rank. Our experimental test bed is the Microsoft's Learning-to-Rank Dataset (MSLR-WEB10K), which can be found in the following links:

- Description page: <https://www.microsoft.com/en-us/research/project/mslr/>
- Download link: <https://1drv.ms/u/s!AtsMfWUz5l8nb0IoJ6Ks0bEMp78>

MSLR-WEB10K contains 10 thousand queries and 1.2 million documents with a 5-level relevance judgments from 0 (irrelevant) to 4 (perfectly relevant). For each query-document pairs, it provides a feature vector with 136 dimensions. More descriptions can be found in its description page.

In this task, we need to train and test four learning-to-rank models: SVMrank, RankNet, MART, and LambdaMART. There are many open-source resources for these learning-to-rank models. Here, we suggest using the following packages:

- SVMrank: [https://www.cs.cornell.edu/people/tj/svm\\_light/svm\\_rank.html](https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html)
- RankNet, MART, and LambdaMart: <https://sourceforge.net/p/lemur/wiki/RankLib/>

Please read the descriptions and instructions of how to use these learning-to-rank packages before doing the experiments.

### Ranking Features (13 points)

The feature vectors in MSLR-WEB10K cover multiple types of ranking features commonly used in commercial search engines, which include but not restricted to

- Query-document matching features
- Document quality features
- User behavior features

Give an example from MSLR-WEB10K for each type of ranking features listed above (3 points for each). Among those three types of features, which one is expected to be most useful in practice? (4 points)

### Ranking Models (37 points)

The data in MSLR-WEB10K are split into 5 folds for the convenience of cross validation. In **Fold3** of MSLR-WEB10K, we have

- “train.txt”: the feature vectors for query-document pairs in the training set.
- “valid.txt”: the feature vectors for query-document pairs in the validation set.
- “test.txt”: the feature vectors for query-document pairs in the test set.

Consider the following learning-to-rank models: SVMrank, RankNet, MART, and LambdaMART. What types of learning-to-rank models are they? Pointwise, pairwise, or listwise? (3 points for each). Train and test each of them with the training data and test data in MSLR-WEB10K **Fold3**, and report nDCG@5 (5 points for each). Which model performs the best in your case? Why? (5 points)

P.S.: Answers are considered correct as long as they fall into some particular ranges, so you don't need to worry about whether you can produce the exact numbers written in the grading guideline. However, the worst model in SVMrank, RankNet, MART, and LambdaMART should have an nDCG@5 larger than 0.255, and the best model should have an nDCG@5 larger than 0.435. If you cannot get those numbers, please try different parameter settings for the models. Note that you may need to preprocess the feature vectors with [Mean Normalization](#) before the training of some learning-to-rank models (e.g., SVMrank).

## Code Submission

Submit the following files you used:

- A pdf or Word document named *answers.pdf/doc* containing all the answers to questions.
- A plain text document named *command.txt* containing all the commands you used along with comments on the purpose of these commands.
- A directory named *code* containing all the codes you write for this coding assignment. **\*\*Please do not include the open-source packages you used.\*\***
- If you use *Galago* with *batch-search*, please include the json file (naming as *query.json*) you used to run the queries.

Please compress all those files into a single *zip* file and name it as:

StudentID\_StudentName\_Lab3.zip (e.g. 20211202\_FullName\_Lab3.zip)