11-791 Project Individual 9: **Learning**

Name: Ruochen Xu Andrew id: ruochenx

1. Learning process

The hyperparameter to tune in this project is the ridge term in logistic regression model. As described in Weka Logistic class documentation, the objective is:

L = -sum[i=1..n]{sum[j=1..(k-1)](Yij * ln(Pj(Xi)))+(1 - (sum[j=1..(k-1)]Yij)) * ln(1 - sum[j=1..(k-1)]Pj(Xi))} + ridge * (B^2)

We tuned the ridge through cross-validation. We let ridge varies from 10^-3 to 10^2, with multiplication factor of 10(i.e. 10^-3, 10^2, …, 10^1, 10^2). At each fold, we compute P@1 for each ridge value, and finally we averaged P@1 for over k folds of runs. Then we picked up the best ridge with highest P@1, and then use it to train with all data. The learned weight is then assigned to the composite ranker and we evaluated its performance over all questions and passages.

2. Learning result

The best hyperparameter got through cross-validation is: 10^-3
The optimized weights trained with above hyperparameter is:

[1.8765966599496826, -0.9417691024612707]

With k = 5, the average of Precision@1 over the k-validation sets is:

ridge	10^-3	10^-2	10^-1	1	10	100
P@1	0.305683	0.305683	0.305683	0.305683	0.305683	0.305079

The performance achieved by learning model(on all questions and passages):

P@1	P@5	MRR	MAP
0.305085	0.285714	0.435666	0.404378

3. Analysis

As we can see the averaged P@1 from cross-validation is nearly the same as the one got from all data. That is because in cross-validation we evaluate on held-out validation data, instead when training with all data, we ended up evaluating with the same data. Usually this will cause P@1 be much higher when trained with all data, however, in our case the model is very simple(with only two parameters). Therefore the performance stayed the same as we train and evaluate with all data.

The performance we achieved in PI5 is summarized below:

P@1	P@5	MRR	MAP
0.297821	0.325424	0.450739	0.418420

We can see that only P@1 increased with the learning to rank method. This is expected since in our cross-validation, we tuned hyperparameter to achieve better P@1. There was no improvement for other measures probably because the other ranker(used cosine similarity) in addition to n-gram ranker is not good enough. Combining it with n-gram ranker is not very helpful.

In the experiment, I use k=5. I choose this number because it is large enough to lower the randomness of spliting data and not too large to increase the computation time significantly.