Analysis

The experiments have been performed with the same neural network architectures and the same learning rate. I.e. 500 hidden units, and 0.025 learning rate. And during the experiments it has been observed that by setting higher learning rate and larger neural networks several epochs are sufficient to produce a good results. So in the experiments the author has set the number of epochs to 1, and 3 (there are 2 different files in the folder). Finally, to avoid overfitting, the L1 and L2 regularisation have been used.

The results of the 5-fold cross validation are shown in the cross_validation_results.jpg file. As we can see the point-wise model performs the worst. The explanation could be that it tries to correctly predict all zero document labels which have a dominant number for any query, and is biased to predict 1s also as 0s. This is where pairwise RankNet model starts to show better performance, because of lambdas for each document we are able to push relevant document to the top and non-relevant to the bottom more efficiently. Document lambdas can be interpreted as weights or forces that if are positive push documents up, and if negative push them down the rank.
Unfortunately, the original RankNet model has a performance drawback caused by a double loop in the lambda computations, where we need to consider all the possible pairs of documents. The alternative version of the RankNet that the author of the homework proposes does not have a double loop but retains equal or even slightly higher mNDCG. The brief discussion of the model is presented in the following section. Finally we can see that list-wise LambdaRank performs the best which could be explained by the fact that it considers NDCG measure during the training phase, and therefore is able to show better results when NDCG is used. However, it's not entirely clear if the last model will be also the best if another measure is used, for example, ERR. This question remains open for the future work. Unfortunately, LambdaRank model suffers from the same runtime issue as RankNet, and addition as we need to compute NDCGs there is an extra cost overhead. The runtimes of models during 5-fold cross-validations are shown in runtime.jpg. As we can see the LambdaRank is the slowest one.

Modified RankNet

As has been mentioned previously the origin model has to create all pairs of documents for each query during the document lambdas computation. This is a slow process as it involves a double loop. The author proposes a modification of the original model that avoids that double loop.

The general idea behind the modified RankNet is to produce a SERP for each query, and optimise w.r.t only pairs present in the query and try to fix their relative positions. For example, if we have one query with document labels [0,1,4], then we shall consider only pairs [[0,1], [1,4]]. And we shall compute document lambdas with respect to those pairs. After the optimisation if we observe the same query, we shall compute its SERP and try to fix relative positions of documents again, such that relevant documents go up the rank, and negative go down.

As we can see from cross_validation_results.jpg, the modified version does not have lower mNDCG and approaches lambdaRank, but from runtime.jpg, we can observe that the model is much faster than the original one and very much similar to the point-wise model's runtime. However, it retains all the benefits of the pair-wise model.

It's important to note that the author does not claim that this method will generalise to other setups, and more experiments are necessary to make an objective conclusion, because the runtime difference could be caused a naive implementation. This remains a subjects of the future work.