

1 Problem statement

The learning to rank problem comprises data on a set of queries, \mathcal{Q} , denote $\mathbf{q}_i \in \mathcal{Q}$ as the i th search query in the dataset with its corresponding features. Each query, \mathbf{q}_i , has a set of properties (hotels), $\mathcal{P}_i = \{p_{i1}, \dots, p_{in_i}\}$, associated with it. Note that the number of hotels associated with each query need not be the same, i.e., each \mathbf{q}_i has n_i properties associated with it. Every property $p_{ij} \in \mathcal{P}_i$ has a corresponding true relevance score $r_{ij} \in \mathbf{r}_i$. In our problem, the true relevance score for a certain hotel in a query is equal to 5 if the hotel was booked, equal to 1 if it was clicked on, and zero otherwise. Now, the model we want to build has the following structure:

$$M : \mathbf{q}_i \times \mathcal{P}_i \rightarrow \mathbb{R}^{n_i} \text{ given as } M(\mathbf{q}_i, \mathcal{P}_i) = \hat{\mathbf{r}}_i \equiv (\hat{r}_{i1}, \dots, \hat{r}_{in_i}), \quad (1)$$

where $\hat{\mathbf{r}}_i$ denotes the vector of predicted relevance scores of the hotels in the i th query. The optimal model produces relevance scores such that the order of elements in $\hat{\mathbf{r}}_i$ is exactly equal to the order of the true relevance scores \mathbf{r}_i , after both are ranked in descending order based on their values.

2 Evaluation approach

Clearly, a model which produces the optimal ranking for every query, as described in the previous section, would be nearly impossible to build. Therefore, we need a way to evaluate a prediction model based on the ranking it produces; a metric that would be suitable is monotonically increasing or decreasing as in the similarity of the predicted ranking compared to the true ranking. A popular metric that is used to this end is the Normalized Discounted Cumulative Gain (NDCG). Let \mathbf{r}_i^s and $\hat{\mathbf{r}}_i^s$ denote the vector of true relevance scores for the i th query, sorted by the true relevance and the predicted relevance, respectively. The Discounted Cumulative Gain (DCG) and Ideal Discounted Cumulative Gain (IDCG) for the i th query are then defined as:

$$DCG_i = \sum_{j=1}^{n_i} \frac{\hat{\mathbf{r}}_{ij}^s}{\log_2(j+1)}, \quad IDCG_i = \sum_{j=1}^{n_i} \frac{\mathbf{r}_{ij}^s}{\log_2(j+1)}. \quad (2)$$

The NDCG for the i th query is then defined as follows:

$$NDCG_i = \frac{DCG_i}{IDCG_i}. \quad (3)$$

Instead of constructing our evaluation metric on the entire list of items, we could also evaluate it only based on the k most important items in the query, yielding the NDCG@ k , which is obtained by replacing n_i with k in equation (2) and recalculating (3). For our purpose, we employ the NDCG@5, thus considering only the top 5 items in the ranking. The NDCG@5 can be calculated for every query and their respective prediction from our model, the final evaluation score of the model on the test set is then calculated as the average NDCG@5 score over all queries $q_i \in \mathcal{Q}$ in the test set.