Unbiased Learning To Rank Review

Eric Hans Lee

1 Premise

The authors want to develop an unbiased Learning To Rank (LTR) method. To be more precise, we have a space of queries $\mathbf{X}$ and a space of documents $\mathbf{Y}$. We also have the relevance score $r(x, y) : \mathbf{X} \times \mathbf{Y} \to \{0, 1\}$ which signifies whether or not a document is relevant to a query. Then I believe the point here, given a set of input data of the form $r(x_i, y_i)$ for $i \in 1 \dots N$, we want to learn some sort of ranking function, which the authors denote as

$$f(x, y) = w\phi(x, y)$$

for some feature vector $\phi(x, y)$ and some scaling factor w.

I was however a bit unclear on the precise plan to do so. The authors introduce the concept of Distributed Cumulative Gain

$$DGS(x, \mathbf{y}) = \sum_{y \in \mathbf{y}} \frac{r(x, y)}{log(rank(y|\mathbf{y}) + 1)} \tag{1}$$

which intuitively seems to quantify the quality of some ranking $\mathbf{y}$ (the higher the DGS, the better the ranking from my understanding). As a bit of notational confusion, I wasn't sure what the difference between $\mathbf{Y}$ and $\mathbf{y}$ was.

The authors then propose an optimization problem to solve

$$\min_{w,\xi} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{i=1}^{N} \frac{1}{q_i} \frac{1}{log(\sum_{y \in \mathbf{y}} \xi_{iy} + 2)} \tag{2}$$

subject some complicated looking constraints. The claim is that this minimization problem in fact produces a good ranking. The variables $\xi_{i,y}$ are not explained; I assume they relate somehow to $\phi(x, y)$ but I could be wrong. In general, the connections between solving (2), calculating (1), and learning $w\phi(x, y)$ were not made very clear to me. I am not an expert in this area but I do think the authors could state things a bit simpler.

At any rate, the authors plan to solve (2), which is a convex optmization problem, via CVXPY using some standard data sets, which I think is a good idea and should be straightforward to solve. In this light, I think the authors have a high probability of success. I can't give any alternative ideas as I didn't fully understand the problem and am not a domain expert.

The authors also (tentatively) plan to code a faster solver for larger data sets. While not a bad idea, I think this would probably be far more time-consuming and far less important compared to the former part of the project (actually solving the problem).