Review of Preference Judgement

Ziying Yang

September 21, 2016

Models for integrating ranking lists Combining ranking lists retrieved by multiple systems is considered to be more effective to meet user's information need. There were several proposed methods to aggregate search results, such as normalizing and averaging the document scores given by systems, voting algorithms and statistical models. Carterette and Petkova [4] proposed an approach that breaks documents in ranking lists into pairs and discovers the binary preference of each pair so that an ideal ranking list (also called *reference* in their later work by Arguello et al. [1]) integrated from distinct ranking lists can be generated based on *pairwise preferences*.

Carterette and Petkova [4] explored two models to combine lists based on pairwise preferences. Logistic regression model simulates the likelihood of all the possible rankings for all the documents. The ideal ranking can be discovered by maximizing the likelihood function, that is looking for the ranking with the highest probability. This model assumes that each retrieved list is a sample from a relevance distribution and so within a pair of two documents, denoted as $\langle i, j \rangle$, the probability that i is more relevant than j can be estimated. However, SVM model does not require this assumption and its optimization process is more efficient than logistic regression model. [CANNOT UNDERSTAND SVM OPTIMIZATION FUNCTION...]

Preference Judgment This pairwise preference is then considered for the relevance judgment which is an essential criteria for evaluating the performance of IR systems. Carterette et al. [5] proposed *preference judgments* that only record which document is considered to be more relevant than the other one in a pair by assessors. Judges only need to make a preference choice for each pair instead of assigning relevance score for each document in the conventional relevance assessing. Preference judgments are the foundation of learning algorithms such as SVM and RankNet [2].

Compared to absolute judgments using scales such as binary, graded and multi-level, preference judgment can improve the assessing accuracy by reducing the noise associated with the distinction of users' perceptions of relevance. Furthermore, the organizer do not need to choose a specific relevance scale for the assessing task, which may lead relevance judgment variations. It is impossible to tell which document is preferred to another if they received the same relevance score (or classified into a same category). However, with preference judgments, preference between any two documents can be discovered and so an ideal ranking list can be generated. In addition, due to the reduction of assessing complexity , assessors spend less time using preference judgments. But the number of comparisons grows, at least in $O(n \log n)$ if the total number of documents is n. In the conventional judging, assessors only need to assign relevance scores for n documents in some order.

Evaluation Carterette and Bennett [3] extended the metric Precision and Recall using preference judgments. Denote d as the length of the ranking list (searching depth) returned by the system. The number of pairs that system make preference is $d(d-1)/2$ (pairs in the ranking list) plus

$k(n-k)$ (all the documents in the ranking list are preferred to the rest documents in the corpus), that is $k(2n-k-1)/2$, denoted as N_{sys_total}. A pair is said to be *correctly ordered* if users and the system agree with the preference. The number of correctly ordered pairs is denoted as $N_{correct}$. The number of pairs that users make preference is denoted as N_{user_total}. The Precision of preference, *ppref* and Recall of preference, *rpref*, can be expressed as:

$$ppref = \frac{N_{correct}}{N_{sys\_total}}$$

$$rpref = \frac{N_{correct}}{N_{user\_total}}.$$

Arguello et al. [1] evaluate ranking lists by computing their *distances* from the ideal list which is called the *reference* and derived by voting algorithms for aggregating preference data. Although their experiments are about evaluating the presenting of retrieved verticals including documents, images, shoppings and so on, the proposed ideas and methods are useful and can be involved for assessing the performances of IR systems returning ranking lists of text documents.

In their experiments conducted on Amazon's Mechanical Turk (AMT), assessors need to make preference for the given pair of elements $\langle i, j\rangle$. The preference judgment of $\langle i, j\rangle$ can be *direct* (elements are presented to assessors to make choice) or *indirect* (assessors are assumed to be consistently transitive, so that if i is preferred to k and k is preferred to j, we can have i is preferred to j). The element i is said to be preferred over j overall if there are more assessors who agree with it than assessors who disagree. The preference judgments of all pairs are used by voting methods to derive the reference, denoted by σ^*. For a given ranking list σ, denote $\sigma(i)$ is the rank of element i in σ.

Kendall's tau, a well-known rank-based distance metric, is considered by authors to measure the distance between σ and σ^*. The total number of discordant pairs is calculated as: [HOW ABOUT CONCORDANT PAIRS?]

$$D(\sigma^*, \sigma) = \sum_{\sigma^*(i)<\sigma*(j)} G(\sigma(i), \sigma(j))$$

where

$$G(\sigma(i), \sigma(j)) = \begin{cases} 1 & \text{if } \sigma(i) > \sigma(j) \\ 0 & \text{otherwise} \end{cases}$$

To make metric top-weighted, the weight of each discordant pair that encodes the rank positions of elements can be added to this formula. The authors try to use DCG-like discount function. Denote δ_d as the cost of swapping element at rank d and $d-1$, which is computed as:

$$\delta_d = \frac{1}{\log(d)} + \frac{1}{\log(d+1)}, \text{ for } 1 \le d \le n.$$

For element i, the average cost of swapping it from $\sigma^*(i)$ to $\sigma(i)$ is [ABSOLUTE VALUE?] [AVERAGE? ADJACENT?]

$$\bar{p}_i(\sigma^*, \sigma) = \begin{cases} 1 & \text{if } \sigma^*(i) = \sigma(i) \\ \frac{p_{\sigma^*(i)} - p_{\sigma(i)}}{\sigma^*(i) - \sigma(i)} & \text{otherwise} \end{cases}$$

where $p_d = \sum_2^d \delta_d$. Thus the the number discordant pairs in terms of rank-sensitiveness can be adjusted to:

$$D^*(\sigma^*, \sigma) = \sum_{\sigma^*(i)<\sigma*(j)} \bar{p}_i(\sigma^*, \sigma)\bar{p}_j(\sigma^*, \sigma)G(\sigma(i), \sigma(j))$$

For each element, the user agreement of it is measured by Fleiss' Kappa. And the agreements of some pairs of assessors who judged at least 100 elements in common are tested by Cohen's Kappa. Arguello et al. [1] state that both kinds of user agreement are high. Users have no difficulty in making preference for their tasks.

To verify that their metric $D^*(\sigma^*, \sigma)$ correlates with user preference, ranking lists are grouped into high-quality, medium-quality and low-quality bins based on scores evaluated by $D^*(\sigma^*, \sigma)$. Then let users on AMT to choose the ranking that they prefer in a pair of lists, that is making preference on ranking lists instead of blocks (elements) in their previours tasks. Authors find that the user agreement of the preference on ranking lists is greater than the preference on blocks. More than that, agreements of users are low for pairs whose both documents are from same bins, which shows that users have difficulty deciding between two rankings in similar qualities.

References

[1] Jaime Arguello, Fernando Diaz, Jamie Callan, and Ben Carterette. A methodology for evaluating aggregated search results. In *Proc. ECIR*, pages 141–152, 2011.

[2] Paul N. Bennett, Ben Carterette, Olivier Chapelle, and Thorsten Joachims. Beyond binary relevance: preferences, diversity, and set-level judgments. *Proc. SIGIR*, pages 53–58, 2008.

[3] Ben Carterette and Paul N. Bennett. Evaluation measures for preference judgments. In *Proc. SIGIR*, pages 685–686, 2008.

[4] Ben Carterette and Desislava Petkova. Learning a ranking from pairwise preferences. In *Proc. SIGIR*, pages 629–630, 2006.

[5] Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Dumais. Here or there. In *Proc. ECIR*, pages 16–27, 2008.