Retrieval Performance Bound Analysis for Single Term Queries

Peilin Yang and Hui Fang
University of Delaware

The Motivation

- IR Ranking models have been studied for decades
- Many models:
  - are based on “bag-of-terms” assumption
  - only play with Document Term Frequency (TF), Inverted Document Frequency (IDF), Document Length (DL) and other collection statistics

The question is …

- Do we reach the upper bound of such models?
  - if so, what would it be?
  - if not, how can we improve?

Find the performance upper bound:

- It is really hard…
- It might be easier if we focus on the simplest case:
  - the queries with only one query term (Single Term Queries)

when there is only one query term…

BM25

$$f(Q,d) = \cancel{\sum_{t\in Q}\frac{(k_3+)\,c_t^q}{k_3+c_t^q}} \cdot \frac{\cancel{(k_1+1)}\cdot c_t^d}{c_t^d + k_1\cdot(1-b+b\cdot\frac{|d|}{avdl})} \cdot \cancel{\ln\left(\frac{N-N_t+0.5}{N_t+0.5}\right)}$$

Pivoted Document Length Normalization

$$f(Q,d) = \cancel{\sum_{t\in Q}} \frac{1+\ln(1+\ln(c_t^d))}{1-s+s\cdot\frac{|d|}{avdl}} \cdot \cancel{\ln\left(\frac{N+1}{N_t}\right)}$$

Dirichlet Language Model

$$f(Q,d) = \cancel{\sum_{t\in Q}} \cancel{\ln}\left(\frac{c(t,d)+\mu\cdot p(t|C)}{|d|+\mu}\right)$$

Summarization is omitted

IDF is omitted

Ranking invariants are omitted

If there is only one query term…

BM25

$$f(Q,d) = \frac{c_t^d}{c_t^d + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avdl})}$$

Pivoted Document Length Normalization

$$f(Q,d) = \frac{1 + \ln(1 + \ln(c_t^d))}{1 - s + s \cdot \frac{|d|}{avdl}}$$

Dirichlet Language Model

$$f(Q,d) = \frac{c_t^d + \mu \cdot p(t|C)}{|d| + \mu}$$

The simplified model

$$f(c_t^d, |d|) = \frac{\alpha \cdot g(c_t^d) + c_1}{\gamma \cdot c_t^d + \beta \cdot h(|d|) + c_2}$$

- g(*) and h(*) are arbitrary non-linear functions
- alpha, beta, gamma, c1, c2 are constants

Partial list of the models that can be transformed to this form

BM25

Pivoted Normalization

Dirichlet Language Model

F2EXP

BM3

DIR+

…

In order to find the performance upper bound:

- We can use brute force method to find the optimum
- But this is too expensive yet inefficient

Follow the cost/gain analysis of learning-to-rank…

Minimize the cost

Cost: pairwise cross-entropy cost applied to the logistic of the difference of the model scores.

$$C_{ij} = \frac{1}{2}(1 - S_{ij})\sigma(s_i - s_j) + \log(1 + e^{-\sigma(s_i - s_j)})$$

Use gradient boost

$$\frac{\partial C_{ij}}{\partial s_i} = \sigma\left(\frac{1}{2}(1 - S_{ij}) - \frac{1}{1 + e^{\sigma(s_i - s_j)}}\right) = -\frac{\partial C_{ij}}{\partial s_j}$$

Simplified as

$$\frac{\partial C_{ij}}{\partial s_i} = \frac{-\sigma}{1 + e^{\sigma(s_i - s_j)}}$$

Follow the cost/gain analysis of learning-to-rank…

Minimize the cost (cont.)

Reduce the cost via stochastic gradient

$$\mathrm{p_k} \rightarrow \mathrm{p_k} - \eta \frac{\partial C}{\partial \mathrm{p_k}} = \mathrm{p_k} - \eta \left(\frac{\partial C}{\partial s_i} \frac{\partial s_i}{\partial \mathrm{p_k}} + \frac{\partial C}{\partial s_j} \frac{\partial s_j}{\partial \mathrm{p_k}} \right)$$

Unfortunately this is the “optimization” cost NOT the actual cost

Follow the cost/gain analysis of learning-to-rank…

Maximize the gain

Inspired by LambdaRank

$$\lambda_{ij} = \frac{\partial C(s_i - s_j)}{\partial s_i} = \frac{\sigma}{1 + e^{\sigma(s_i - s_j)}} |\Delta_{MAP}|$$

$$\lambda_{ij} = \frac{\sigma}{1 + e^{\sigma(s_i - s_j)}} \frac{1}{|R|} \left(\left| \frac{n}{r_j} - \frac{m}{r_i} \right| + \sum_{k=r_j+1}^{r_i-1} \frac{\mathrm{I}(k)}{k} \right)$$

Experiments: Tested Models

- DIR

$$\frac{c(t,d)+\mu\cdot p(t|C)}{|d|+\mu}$$

- TFDL1

$$\frac{c(t,d)+c_1}{|d|+c_2}$$

- TFDL2

$$\frac{\alpha\cdot c(t,d)+c_1}{\beta\cdot|d|+c_2}$$

Collections and Queries

Collections	Topics	# of queries
disk1&2	57,75,77,78	4
ROBUST04	312,348,349,364,367,379, 392,395,403,417,424	11
WT2G	403,417,424	3
GOV2	757,840	2

Experiment Results

	Models	disk1&2	Robust04	WT2G	GOV2
Basic Models	DIR	0.4009	0.3823	0.3660	0.2083
	BM25	**0.4016**	**0.3824**	**0.4038**	0.2896
	PIV	0.3987	0.3812	**0.4038**	**0.3079**
	F2EXP	0.4000	0.3682	0.3183	0.1950
	BM3	0.4015	0.3823	0.3792	0.2554
	DIR+	0.4009	0.3823	0.3794	0.2083
Upper Bounds	DIR^U	0.4244	0.4136	0.4055	0.2724
	$TFDL1^U$	0.4273	0.4209	0.4095	0.3193
	$TFDL2^U$	**0.4273**	**0.4209**	**0.4095**	**0.3255**

Future Work

- Extend to the queries with multiple terms
- Mathematical prove

Thank You!
Q & A