On the Equilibrium of Query Reformulation and Document Retrieval

Authors:
Shihao Zou (UCL), Guanyu Tao (UCL), Jun Wang (UCL),
Weinan Zhang (SJTU), Dell Zhang (UoL)

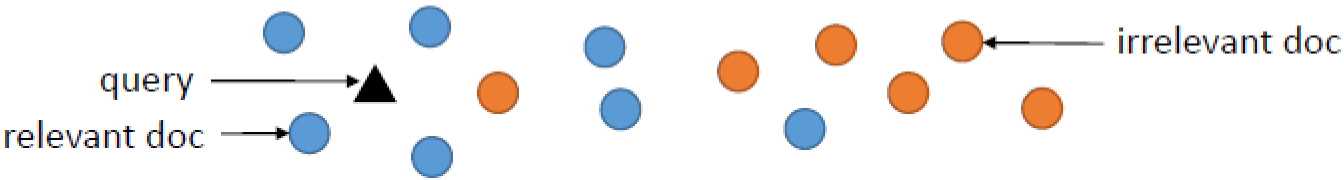
ICTIR 2018, Tianjin, China

Self-introduction – Shihao Zou

- BSc in Beijing Institute of Technology in 2017.06
- Master of Research in University College London, 2017.09 – now
- I am going to pursue PhD in University of Alberta, Canada in Jan 2019
- Research interest: machine learning in data mining topics, reinforcement learning

Introduction

Two challenges in information retrieval:



- how to formulate optimal queries to best represent the user's information needs
- relevance estimation for the document given the information need representation

- Query reformulation (relevance feedback)
- Retrieval model

Equilibrium theory of information retrieval:

- a strategic game, simultaneously playing between the query reformulation and the retrieval model

Introduction

Intuition:

- The query reformulation would refine the query that is the **best response** to the relevance estimation given by retrieval model

- The retrieval model would also need to produce the document relevant estimation that is the **best response** toward the formulated query

- Two components shall cooperate to achieve the best response to each other. (an equilibrium state)

Definition: IR Strategic Game

An IR Strategic Game is a tuple (P, S, U) , where :

- $P = \{Q, M\}$ is the set of two players: query formulator Q and retrieval model M.
- $S = S_Q \times S_M$ are finite sets of strategies available to player Q and M.
- $s_q \in S_Q$ denotes whether the term is included in the query or not.
- $s_m \in S_M$ denotes relevance estimation by retrieval model.
- An equilibrium state: both players have no incentive to change their strategies s_m^* and s_q^*, so that
  $u_Q(s_q^*, s_m^*) \geq u_Q(s_q, s_m^*),\ u_M(s_q^*, s_m^*) \geq u_M(s_q^*, s_m)$

IR Game with Relevance Feedback

Common utility:

$$u(\mathbf{s}_q, \mathbf{s}_m) = \frac{1}{|D_r|} \sum_{\mathbf{d}_i \in D_r} \log p(r = 1|\mathbf{d}_i, \mathbf{q}; \theta) - \frac{1}{|D_n|} \sum_{\mathbf{d}_i \in D_n} \log p(r = 0|\mathbf{d}_i, \mathbf{q}; \theta),$$

Toy example:

Table 1: An IR game example (relevance feedback).

	$\mathbf{d}_1$	$\mathbf{d}_2$
t_1	1	0
t_2	0	1
r	1	0

(a) Corpus

	$\mathbf{s}_{m_1} = \{1, 0.2\}$	$\mathbf{s}_{m_2} = \{0.2, 1\}$
$\mathbf{s}_{q_1} = \{1, 0\}$	-1.0064	-1.2913
$\mathbf{s}_{q_2} = \{0, 1\}$	-1.4913	-2.0064

(b) Utilities of Strategies

$$p(r = 1|\mathbf{d}_i, \mathbf{q}; \theta) = \text{sigmoid}\Big(\theta_1 \mathbf{q}_1 \mathbf{d}_{i1} + \theta_2 \mathbf{q}_2 \mathbf{d}_{i2}\Big)$$

$$p(r = 1|\mathbf{d}_1, \mathbf{q}; \theta) = \text{sigmoid}(1 \times 1 \times 1 + 0.2 \times 0 \times 0) = 0.7311$$

$$p(r = 1|\mathbf{d}_2, \mathbf{q}; \theta) = \text{sigmoid}(1 \times 1 \times 0 + 0.2 \times 0 \times 1) = 0.5$$

$$u(\mathbf{s}_q, \mathbf{s}_m) = \log p(r = 1|\mathbf{d}_1, \mathbf{q}; \theta) + \log p(r = 0|\mathbf{d}_2, \mathbf{q}; \theta) = -1.0064.$$

IR Game with Pseudo Relevance Feedback

Utility for retrieval model:

$$u_M(\mathbf{s}_q, \mathbf{s}_m) = \frac{1}{|D_r|} \sum_{\mathbf{d}_i \in D_r} \log p(r = 1 | \mathbf{d}_i, \mathbf{q}; \theta) - \frac{1}{|D_n|} \sum_{\mathbf{d}_i \in D_n} \log p(r = 0 | \mathbf{d}_i, \mathbf{q}; \theta).$$

Utility for query reformulation (top-k):

$$u_Q(\mathbf{s}_q, \mathbf{s}_m) = \frac{1}{|D_k|} \sum_{\mathbf{d}_i \in D_k} \log p(r = 1 | \mathbf{d}_i, \mathbf{q}; \theta) - \frac{1}{N - |D_k|} \sum_{\mathbf{d}_i \notin D_k} \log p(r = 0 | \mathbf{d}_i, \mathbf{q}; \theta),$$

Top-k makes the two utilities different

IR Game with Pseudo Relevance Feedback

Toy example for pseudo relevance feedback:

Table 2: An IR game example (pseudo relevance feedback).

	$\mathbf{d}_1$	$\mathbf{d}_2$
t_1	1	0
t_2	0	1
r	1	0

(a) Corpus

	$\mathbf{s}_{m_1} = \{1, 0.2\}$	$\mathbf{s}_{m_2} = \{0.2, 1\}$
$\mathbf{s}_{q_1} = \{1, 0\}$	(-1.0064, -1.0064)	(-1.2913, -1.2913)
$\mathbf{s}_{q_2} = \{0, 1\}$	(-1.2913, -1.4913)	(-1.0064, -2.0064)

(b) Utilities of Strategies (u_Q, u_M)

$$p(r = 1|\mathbf{d}_1, \mathbf{q}; \theta) = \text{sigmoid}(1 \times 0 \times 1 + 0.2 \times 1 \times 0) = 0.5$$

$$p(r = 1|\mathbf{d}_2, \mathbf{q}; \theta) = \text{sigmoid}(1 \times 0 \times 0 + 0.2 \times 1 \times 1) = 0.5498$$

$$u_Q(\mathbf{s}_{q_2}, \mathbf{s}_{m_1}) = \log p(r = 1|\mathbf{d}_2, \mathbf{q}; \theta) + \log p(r = 0|\mathbf{d}_1, \mathbf{q}; \theta) = -1.2913$$

$$u_M(\mathbf{s}_{q_2}, \mathbf{s}_{m_1}) = \log p(r = 1|\mathbf{d}_1, \mathbf{q}; \theta) + \log p(r = 0|\mathbf{d}_2, \mathbf{q}; \theta) = -1.4913$$

Experiment: text retrieval

Five training schemes:

- Case 1: No iteration (Naïve)
- Case 2: Update once (Rocchio)
- Case 3: Query Iteration (Conv-Q)

$$\theta_i = \text{sigmoid}(\mathbf{q}^\top \mathbf{d}_i) = \frac{1}{1 + e^{-\mathbf{q}^\top \mathbf{d}_i}}$$

$$\frac{\partial u_Q(\mathbf{s}_q, \mathbf{s}_m)}{\partial \mathbf{q}} = \frac{1}{|D_r|} \sum_{d_i \in D_r} (1 - \theta_i)\mathbf{d}_i - \frac{1}{|D_n|} \sum_{d_i \in D_n} \theta_i \mathbf{d}_i$$

- Case 4: Retrieval Model Iteration (Conv-M)

$$\theta_i = \text{sigmoid}\left(\sum_{k=1}^{K} w_k \cdot (\mathbf{d}_i^k)^\top \mathbf{q}^k\right)$$

Logistic regression of K weight schemes

$$\frac{\partial u_M(\mathbf{s}_q, \mathbf{s}_m)}{\partial w_k} = \frac{1}{|D_r|} \sum_{\mathbf{d}_i \in D_r} (1 - \theta_i) \cdot (\mathbf{d}_i^k)^\top \mathbf{q}^k - \frac{1}{|D_n|} \sum_{\mathbf{d}_i \in D_n} \theta_i \cdot (\mathbf{d}_i^k)^\top \mathbf{q}^k$$

- Case 5: Equilibrium of the Query and Retrieval Model (Equil-Q&M)

Experiment: text retrieval

Dataset: TREC disks 4 & 5

Utility after each iteration in training stage:

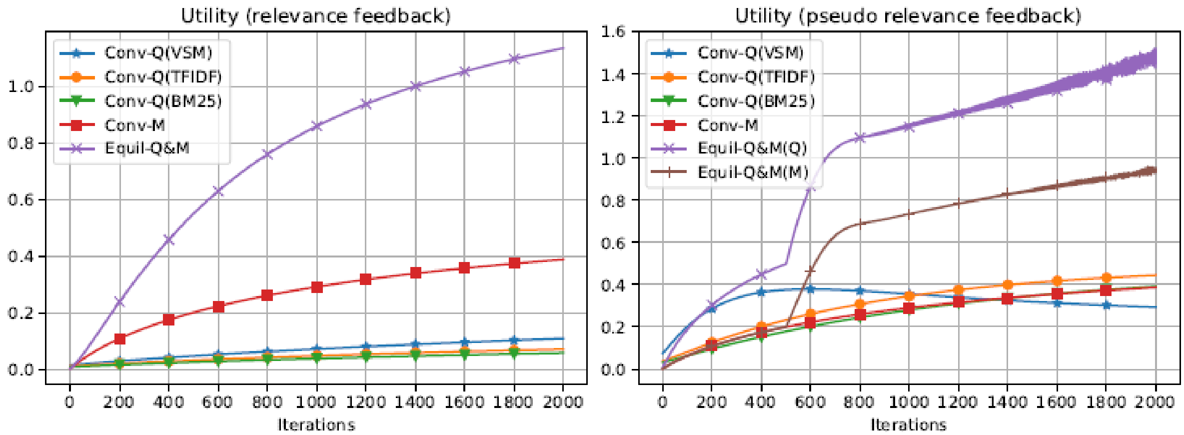


Figure 1: Utility in both cases of relevance feedback

Observations:

- Iterations on Q&M in both RF and PRF help improve the ranking performance (utility) in training stage

Text retrieval (RF)

Algorithm	NDCG@10	NDCG@30	MRR
Naive (VSM)	0.395±0.37	0.412±0.32	0.352±0.38
Naive (TFIDF)	0.511±0.37	0.528±0.33	0.478±0.41
Naive (BM25)	0.504±0.37	0.517±0.32	0.459±0.40
Rocchio (VSM)	0.407±0.37	0.422±0.32	0.367±0.39
Rocchio (TFIDF)	0.519±0.38	0.536±0.33	0.487±0.41
Rocchio (BM25)	0.518±0.37	0.531±0.32	0.474±0.40
Conv-Q (VSM)	0.527±0.34	0.554±0.29	0.475±0.39
Conv-Q (TFIDF)	0.568±0.35	0.571±0.30	0.530±0.40
Conv-Q (BM25)	0.563±0.35	0.573±0.30	0.522±0.40
Conv-M	0.463±0.38	0.482±0.34	0.431±0.41
Equil-Q&M	**0.583±0.34**	**0.601*±0.29**	**0.537*±0.39**
Algorithm	**P@10**	**P@30**	**MAP**
Naive (VSM)	0.152±0.18	0.134±0.15	0.184±0.16
Naive (TFIDF)	0.221±0.22	0.179±0.18	0.263±0.23
Naive (BM25)	0.217±0.22	0.178±0.17	0.262±0.23
Rocchio (VSM)	0.162±0.18	0.139±0.15	0.193±0.17
Rocchio (TFIDF)	0.225±0.22	0.186±0.18	0.276±0.24
Rocchio (BM25)	0.221±0.21	0.183±0.17	0.272±0.24
Conv-Q (VSM)	0.245±0.23	0.212±0.18	0.288±0.22
Conv-Q (TFIDF)	0.264±0.24	0.220±0.20	0.317±0.25
Conv-Q (BM25)	0.265±0.24	0.214±0.20	0.319±0.25
Conv-M	0.190±0.20	0.160±0.16	0.238±0.21
Equil-Q&M	**0.278*±0.24**	**0.233±0.19**	**0.331*±0.25**

- Datasets
  - TREC disks 4 & 5
- Task
  - Text retrieval ranking
- Key observations
  - Conv-Q shows better performance than Naïve and Rocchio, esp. Conv-Q(TFIDF)
  - Conv-M fails to perform well on test set although well on training set
  - The best Equil-Q&M indicates the effectiveness of coordinating Q and M

Text retrieval (PRF)

Algorithm	NDCG@10	NDCG@30	MRR
Naive (VSM)	0.323±0.38	0.378±0.29	0.287±0.36
Naive (TFIDF)	0.463±0.36	0.493±0.30	**0.413±0.38**
Naive (BM25)	0.439±0.35	0.474±0.28	0.375±0.36
Rocchio (VSM)	0.323±0.36	0.378±0.30	0.285±0.36
Rocchio (TFIDF)	0.460±0.36	0.493±0.30	0.410±0.38
Rocchio (BM25)	0.444±0.35	0.477±0.29	0.386±0.37
Conv-Q (VSM)	0.245±0.34	0.308±0.29	0.228±0.33
Conv-Q (TFIDF)	0.428±0.37	0.465±0.32	0.370±0.38
Conv-Q (BM25)	0.400±0.36	0.456±0.30	0.349±0.36
Conv-M	0.415±0.37	0.447±0.31	0.367±0.39
Equil-Q&M	**0.469*±0.37**	**0.499±0.31**	0.397±0.38
Algorithm	**P@10**	**P@30**	**MAP**
Naive (VSM)	0.112±0.14	0.100±0.12	0.158±0.15
Naive (TFIDF)	0.200±0.21	0.142±0.14	0.239±0.22
Naive (BM25)	0.187±0.20	0.137±0.13	0.226±0.21
Rocchio (VSM)	0.108±0.14	0.100±0.12	0.157±0.16
Rocchio (TFIDF)	0.207±0.22	0.145±0.14	0.244±0.23
Rocchio (BM25)	0.193±0.20	0.141±0.14	0.233±0.22
Conv-Q (VSM)	0.095±0.15	0.090±0.12	0.138±0.15
Conv-Q (TFIDF)	0.211±0.23	0.150±0.16	0.253±0.24
Conv-Q (BM25)	0.180±0.21	0.143±0.15	0.234±0.23
Conv-M	0.154±0.17	0.122±0.13	0.205±0.19
Equil-Q&M	**0.223*±0.16**	**0.162*±0.16**	**0.257±0.23**

- Datasets
  - TREC disks 4 & 5
- Task
  - Text retrieval ranking
- Key observations
  - Conv-Q shows worse performance than Naïve and Rocchio (One cannot fully rely on the top-k retrieved docs from the model to update the query)
  - The best Equil-Q&M indicates the coordination of Q and M help overcome the issue of bad query representation

Experiment: User-based item recommendation

- Revise a linear retrieval model as $\theta_i = p(r = 1|\mathbf{d}_i, \mathbf{q}_u) = \mathrm{sigmoid}(\mathbf{q}_u^\top \mathbf{W} \mathbf{d}_i)$

  $\mathbf{q}_u$: target user's profile, $\mathbf{d}_i$: memory user's profile.

Results:

Algorithm	NDCG@10	NDCG@30	MRR
Rocchio	0.194±0.31	0.220±0.28	0.167±0.28
Conv-Q	0.201±0.31	0.234±0.28	0.172±0.29
Conv-M	0.199±0.32	0.223±0.29	0.170±0.30
Equil-Q&M	**0.204*±0.31**	**0.237±0.28**	**0.174*±0.29**
Algorithm	**P@10**	**P@30**	**MAP**
Rocchio	0.111±0.20	0.039±0.07	0.021±0.03
Conv-Q	0.113±0.19	0.043±0.07	0.024±0.03
Conv-M	0.111±0.21	0.040±0.07	0.022±0.03
Equil-Q&M	**0.116±0.19**	**0.045*±0.07**	**0.025±0.03**

- Datasets
  - Movielens(100k)
- Task
  - Item recommendation
- Key observations
  - Conv-Q and Conv-M outperform Rocchio
  - Equil-Q&M slightly exceeds the other three cases.

Summary of this work:

- We study the interactions between query reformulation and retrieval model relevance estimation in a game theoretical framework.
- The performance of an equilibrium solution from relevance feedback consistently outperforms other separate cases.
- Larger dataset is required to investigate more interesting things in the equilibrium solution.
- We shall perform a deeper inquiry of the utility design in the proposed normal-form IR game.

Thank you for listening.

Questions?