

UNIVERSITY of DELAWARE

UNIVERSITY of DE

# Feedback

(Search and Data Mining)

Hui Fang  
Department of Electrical and Computer Engineering  
University of Delaware

1

UNIVERSITY of DELAWARE

UNIVERSITY of DE

## Relevance Feedback

```

graph LR
    User -- Query --> RE[Retrieval Engine]
    RE --> DC[(Document collection)]
    DC --> RE
    RE -- "Results: d1: 3.5, d2: 2.4, ..., dk: 0.5, ..." --> User
    User -- "Judgments: d1: +, d2: -, ..., dk: -" --> FB[Feedback]
    FB -- "Updated query" --> RE
  
```

- **Implicit Feedback:** Use clickthrough data
- **Pseudo Feedback:** Use top-K ranked results

UNIVERSITY of DELAWARE

UNIVERSITY of DE

## Feedback in Vector Space Model

- How to learn from examples to improve retrieval performance?
  - Positive examples: documents that are known to be relevant
  - Negative examples: documents that known to be non-relevant
- General method: query modification
  - Adding new terms
  - Adjusting weights of old terms

3

UNIVERSITY of DELAWARE

UNIVERSITY of DE

## Illustration of Rocchio Feedback

4

UNIVERSITY of DELAWARE

UNIVERSITY of DE

## Rocchio Feedback: Formula

New query

Parameters

$$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

Original query

Rel docs

Non-rel docs

5

UNIVERSITY of DELAWARE

UNIVERSITY of DE

## Rocchio in Practice

- Negative examples are not very important.
- Need to keep relatively high weight on the original query weights.
- Can be used for relevance feedback and pseudo feedback
- Usually robust and effective

6

UNIVERSITY of DELAWARE

UNIVERSITY of DE

### Kullback-Leibler (KL) Divergence

Query likelihood

$$f(q, d) = \sum_{\substack{w \in d \\ w \in q}} [c(w, q)] \left[ \log \frac{p_{\text{seen}}(w|d)}{\alpha_d p(w|C)} \right] + n \log \alpha_d$$

KL-divergence (cross entropy)

$$f(q, d) = \sum_{\substack{w \in d, p(w|\theta_Q) > 0}} [p(w|\theta_Q)] \log \frac{p_{\text{seen}}(w|d)}{\alpha_d p(w|C)} + \log \alpha_d$$

Query LM

$$p(w|\hat{\theta}_Q) = \frac{c(w, Q)}{|Q|}$$

UNIVERSITY of DELAWARE

UNIVERSITY of DE

### Model-based Feedback

Document  $D$   $\rightarrow \theta_D$

Query  $Q$   $\rightarrow \theta_Q$

$\theta_D, \theta_Q \rightarrow D(\theta_Q || \theta_D) \rightarrow \text{Results}$

$\text{Results} \rightarrow \text{Feedback docs } F = \{d_1, d_2, \dots, d_n\}$

$F \rightarrow \text{Generative model} \rightarrow \theta_F$

$\theta_Q' = (1 - \alpha)\theta_Q + \alpha\theta_F$

$\alpha = 0 \rightarrow \theta_Q' = \theta_Q$  (No feedback)

$\alpha = 1 \rightarrow \theta_Q' = \theta_F$  (Full feedback)

UNIVERSITY of DELAWARE

UNIVERSITY of DE

### Mixture Model for Feedback

Background words  $\xrightarrow{\lambda} P(w|C)$

Topic words  $\xrightarrow{1-\lambda} P(w|\theta)$

$w \rightarrow F = \{d_1, \dots, d_n\}$

$$\log p(F|\theta) = \sum_i \sum_w c(w; d_i) \log[(1 - \lambda)p(w|\theta) + \lambda p(w|C)]$$

Maximum likelihood  $\theta_F = \underset{\theta}{\operatorname{argmax}} \log p(F|\theta)$

$\lambda$  = noise in feedback documents

UNIVERSITY of DELAWARE

UNIVERSITY of DE

### Example of learned query model

Query: "airport security"

Mixture model approach

$\lambda = 0.9$

$w$	$P(w \theta_Q)$
security	0.0558
airport	0.0546
beverage	0.0488
alcohol	0.0474
bomb	0.0236
terrorist	0.0217
author	0.0206
license	0.0188
bond	0.0186
counter-terror	0.0173
terror	0.0142
newsnets	0.0129
attack	0.0124
operation	0.0121
headline	0.0121

Web database

Top 10 docs

$\lambda = 0.7$

$w$	$P(w \theta_Q)$
the	0.0405
security	0.0377
airport	0.0342
beverage	0.0305
alcohol	0.0304
to	0.0268
of	0.0241
and	0.0214
author	0.0156
bomb	0.0150
terrorist	0.0137
in	0.0135
license	0.0127
state	0.0127
by	0.0125