Relevance Feedback

Jiangtao Gou
SAS® Text Analytics

July 13, 2012

Table of contents

Why Relevance Feedback?

- Have you noticed that when using key words to search, sometimes you don't get the most relevant documents you want?
- Actually, there might be a gap between the key words and the user's real searching need.
- By learning more information from the feedback, we can either adjust the weights of the terms in the original query, or add more words to the query[1].

IDEA: you may not know what you are looking for, but you will know when you see it[2].

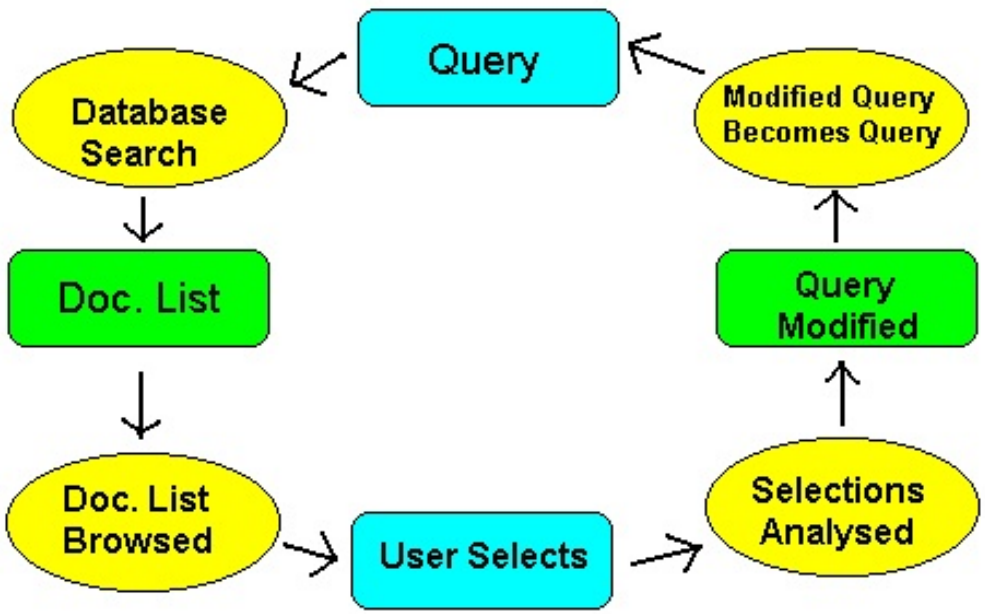
---

[1] http://www.cs.cmu.edu/~chenminl/

[2] http://www.umiacs.umd.edu/~jimmylin/LBSC796-INFM718R-2006-Spring/lecture7.ppt

Basic Framework

1. Take the results that are initially returned from a given query.
2. Use information about whether or not those results are relevant to perform a new query[3].



[3] http://instruct.uwo.ca/gplis/601/week6/feedmod.html

Basic Questions in Relevance Feedback (RF)

- Q1 How could a system compute the initial ranking list?
- Q2 How to choose documents for relevance feedback so that the system can learn most from the feedback information?
- Q3 How could we get the relevance feedback from users?
- Q4 How to update the ranking list when a system has relevance feedback?
- Q5 How to measure the effectiveness of a relevance feedback system?

Q1: How could a system compute the initial ranking list?

- Relevance Algorithm without relevance information
- Non-Textual relevance features may be included in

Q2: How to choose documents for relevance feedback so that the system can learn most from the feedback information?

Three basic ways

- Top K
  System returns the top K documents in the initial ranking list.
- Gapped Top K
  System selects documents with a gap according to the initial ranking list.
- K cluster centroid
  System performs explicit clustering.

Q2: How to choose documents for relevance feedback so that the system can learn most from the feedback information?

The user may not be able to clearly describe the documents which the user want, so the system should return relevant documents from different categories in the initial result. The more diverse the collection is, the most likely the documents which the user is interested in are in the collection.

- Active Feedback: Balance the relevance and diversity

Active Feedback: Balance the relevance and diversity

* Maximal Marginal Relevance (MMR)

$$s_j^* = \arg \max_{s_j \in \mathcal{D} \setminus S_{j-1}} \left(\lambda \cdot sim_1(s_j, q) - (1 - \lambda) \max_{s_i \in S_{j-1}} sim_2(s_j, s_i) \right),$$

* Probabilistic Latent MMR
  Topic Models are involved in by using latent Dirichlet Allocation (LDA)
* Portfolio theory of information retrieval

$$\max \mathbf{O}_n = \mathrm{E}[\mathbf{R}_n] - b\mathrm{Var}[\mathbf{R}_n].$$

Q3: How could we get the relevance feedback from users?

Original Query

→ User Judgement (Explicit RF)

→ User Behavior (Implicit RF)

→ Top-K retrieved (Pseudo/Blind RF)

Expanded/Re-weighted Query

* Explicit RF is obtained from assessors of relevance indicating the relevance of a document retrieved for a query.
* Implicit RF is inferred from user behavior, such as noting which documents they do and do not select for viewing, the duration of time spent viewing a document, or page browsing or scrolling actions.
* Pseudo RF automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction[4].

[4] http://en.wikipedia.org/wiki/Relevance_feedback

Implicit Relevance Feedback

- Clicks are informative but biased. Relative preferences derived from clicks are reasonably accurate on average.
- Copy and Save are noninformative.

Q4: How to update the ranking list when a system has relevance feedback?

* Probabilistic Relevance Framework (PRF)

  Idea (Probability ranking principle): "If documents are ranked by decreasing probability of relevance, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of the data available to the system."

* Language Model (LM)

  Idea: Each term is generated from a linear combination of one of different specific models and one background models.

* Vector Space Model (VSM)

  Idea: Measure the similarities between query vectors and document vectors.

* Combined Model

Q5: How to measure the effectiveness of a relevance feedback system?

* Normalized Discounted Cumulative Gain $\mathrm{nDCG_p}$

  The most popular measure in machine learning relevance research

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i)}.$$

  Where rel_i is the graded relevance of the result at position i. $\mathrm{nDCG_p}$ is a normalized version of $\mathrm{DCG_p}$.

* Average r/recall, average p/precision, p-r plot, Spearman's rank correlation coefficient

Outline

Relevance Feedback: Improve the quality of search results

- Content-based Image Search

  Users judge the relevance of a picture much faster than a document.
- Market Research with Social Media

  Customers may evaluate a certain product and express their ideas on Twitter, Facebook. This company needs to have high quality data by text mining to predict the future market.
- Industries, goverments

Benefits

- (Hiemstra and Robertson) Average precision increases 7% - 21% by Language Model (LM), and 6% - 24% by Probabilistic Relevance Framework (BM25) on the TREC collection.
- (Lee) Spearman correlation coefficient ρ increases 20% - 22% by Vector Space Model (VSM), and 14% - 17% by Probabilistic Relevance Framework (BM25) on the TREC collection. The combination of VSM and BM25 may change the results between -1% and 7%.
- (Wang and Zhu) By using the portfolio theory to choose document for relevance feedback, an additional improvement between 0% and 14% are observed on the TREC collection.

Lemur: A Language Modeling & Information Retrieval Toolkit

- The Lemur toolkit supports the construction of basic text retrieval systems using language modeling methods, as well as traditional methods such as those based on the vector space model and Okapi.
- Lemur is particularly useful for researchers in language modeling and information retrieval who do not want to write their own indexers but would rather focus on developing new techniques and algorithms.
- The toolkit which is written by C^{++} has been used to carry out experiments on several different aspects of language modeling. For example, query expansion methods to estimate query models on standard TREC collections[5].

[5] http://classes.soe.ucsc.edu/ism293/Spring09/material/bootcamp_3_Lemur.pdf

Lemur supports relevance feedback

Lemur currently supports five different models[6].

- The popular TFIDF retrieval model
- The Okapi BM25 retrieval function
  a Probabilistic Relevance Model (PRF)
- The KL-divergence language model based retrieval method
  a Language Model (LM)
- The InQuery (CORI) retrieval model
- Cosine similarity model
  a Vector Space Model (VSM)

[6] http://www.lemurproject.org/doxygen/lemur/html/RelFBEval.html

Outline

Basic Idea of Probabilistic Relevance Framework

Summation of term's log odds ratio which appears in the query

$$\Pr\left(Rel = 1 | \mathbf{D}, \mathbf{Q}\right)$$
$$\propto_q \sum_{i \in \mathbf{Q}} \log \left(\frac{\Pr\left(TF_i = tf_i | Rel = 1\right)}{\Pr\left(TF_i = tf_i | Rel = 0\right)} \cdot \frac{\Pr\left(TF_i = 0 | Rel = 0\right)}{\Pr\left(TF_i = 0 | Rel = 1\right)} \right)$$

where $\propto_q$ indicates rank equivalence, Rel is a binary random variable which indicates the relevance of a document given an information need, $\mathbf{D}$ indicates a document, $\mathbf{Q}$ indicates a query, and TF_i indicates the term frequency of term i.

BM25 Formula

BM25 is a model under Probabilistic Relevance Framework[7].

$$w_i^{BM25}(tf) = qtf_i \cdot \frac{tf_i}{k_1 B + tf_i} w_i^{RSJ},$$

where

$$w_i^{RSJ} = \log\left(\frac{r_i + 0.5}{R - r_i + 0.5} \cdot \frac{N - R - n_i + r_i + 0.5}{n - r_i + 0.5}\right),$$

where N is the number of documents, R is the number of relevant documents, n_i is the number of documents that contain the term i, r_i is the number of relevant documents which contain the term i.

[7] Refer to Summer Report 1 section 2 and Handout 1 for details about the deduction of relevant formula

Basic Idea of Language Model

The metric is defined as the relative change in the document likelihoods, as the likelihood ratio of the conditional and the prior probabilities, which is

$$S(\mathbf{D}, \mathbf{Q}) = \frac{\Pr(\mathbf{D}|\mathbf{Q})}{\Pr(\mathbf{D})},$$

$$\log S(\mathbf{D}, \mathbf{Q}) \propto_q \sum_{i \in \mathbf{Q}} q_i(\mathbf{Q}) \log \left(\Pr(T_i = 1|\mathbf{D})\right).$$

Linear Interpolation

Instead of $p(T_i = 1|\mathbf{D})$, we use a linear combination to estimate $\Pr(T_i = 1|\mathbf{D})$, which is

$$\tilde{p}\,(T_i = 1|\mathbf{D}) = \pi p(T_i = 1|\mathbf{D}) + (1 - \pi)p(T_i = 1),$$

We treat π as a parameter, and k_i's as hidden variables. When $k_i = 0$, we generate a word in query by $\Pr(T_i = 1)$, whichi is the background model. When $k_i = 1$, we generate a word in query by $\Pr(T_i = 1|\mathbf{D})$, which is a specific model by document $\mathbf{D}$. We can use EM (Expectation-maximization) algorithm to compute the suitable π[8].

[8] Refer to Summer Report 1 section 3 and Handout 1 for details about the deduction of relevant formula

Basic Idea of Vector Space Model

The similarity is measured by cosine distance.

$$sim(\mathbf{d}_j, \mathbf{q}) = \cos(\mathbf{d}_j, \mathbf{q}) = \frac{\sum_{i=1}^{|\mathbf{V}|} w_{ij} w_{iq}}{\sqrt{\sum w_{ij}^2} \sqrt{\sum w_{iq}^2}}$$

If we have relevance information, the optimal query will try to maximizes the similarity with (the center of) relevant documents, while minimize the similarity with (the center of) non-relevant documents, which results the expression

$$\begin{aligned} \mathbf{q}_{opt} &= \frac{1}{|\sum_{\mathbf{d}_j \in C_R} \mathbf{d}_j|} \sum_{\mathbf{d}_j \in C_R} \mathbf{d}_j - \frac{1}{|\sum_{\mathbf{d}_j \in C_{NR}} \mathbf{d}_j|} \sum_{\mathbf{d}_j \in C_{NR}} \mathbf{d}_j \\ &\approx \frac{1}{|C_R|} \sum_{\mathbf{d}_j \in C_R} \mathbf{d}_j - \frac{1}{|C_{NR}|} \sum_{\mathbf{d}_j \in C_{NR}} \mathbf{d}_j, \end{aligned}$$

where $N = |C_R| + |C_{NR}|$ is the total number of documents.

Rocchio Algorithm

Rocchio formula (Rocchio 1971)

$$\mathbf{q}_m = \alpha\mathbf{q} + \frac{\beta}{|D_R|}\sum_{\mathbf{d}_j \in C_R}\mathbf{d}_j - \frac{\gamma}{|D_{NR}|}\sum_{\mathbf{d}_j \in C_{NR}}\mathbf{d}_j,$$

where $\mathbf{q}_m$ means modified $\mathbf{q}$, and D_R and D_{NR} are known relevant and known non-relevant document.

Ide-regular (Ide 1971)

$$\mathbf{q}_m = \alpha\mathbf{q} + \beta\sum_{\mathbf{d}_j \in C_R}\mathbf{d}_j - \gamma\sum_{\mathbf{d}_j \in C_{NR}}\mathbf{d}_j,$$

Ide-dec-hi (Ide and Salton 1971)

$$\mathbf{q}_m = \alpha\mathbf{q} + \beta\sum_{\mathbf{d}_j \in C_R}\mathbf{d}_j - \gamma\mathbf{d}_j^{maxNR},$$

where $\mathbf{d}_j^{maxNR}$ is the non-relevant document which received the highest ranking from the IR system.

Outline

There are 7 documents.

Doc 1: Nobel Prize Alfred Nobel scientist inventor Nobel Foundation
Doc 2: Physics Nobel Prize effect great American scientist
Doc 3: great prize invention great prize
Doc 4: American physicist great American
Doc 5: physics effect physics effect prize
Doc 6: Nobel award award award
Doc 7: Olympics award award great American

Suppose there is a user who wants to seek information about Richard Feynman, but he forgets Richard Feynman's name. He only remembers that Feynman has won the Nobel Prize. So he uses the query

- Nobel Prize

to search in the corpus, which are word 8 and word 11.
We know that Doc 2, Doc 4, and Doc 5 are relevant to the information which the user wanted.

Term Frequency of Toy Example

	word	1	2	3	4	5	6	7
1	alfred	1	0	0	0	0	0	0
2	american	0	1	0	2	0	0	1
3	award	0	0	0	0	0	3	2
4	effect	0	1	0	0	2	0	0
5	foundation	1	0	0	0	0	0	0
6	great	0	1	2	1	0	0	1
7	invent	1	0	1	0	0	0	0
8	nobel	3	1	0	0	0	1	0
9	olympics	0	0	0	0	0	0	1
10	physics	0	1	0	1	2	0	0
11	prize	1	1	2	0	1	0	0
12	science	1	1	0	0	0	0	0

Probabilistic Relevance Framework

Table: BM25 term-weighting when No relevance information

word	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7
1	0.70	0	0	0	0	0	0
2	0	0.13	0	0.23	0	0	0.15
3	0	0	0	0	0	0.81	0.67
4	0	0.40	0	0	0.67	0	0
5	0.70	0	0	0	0	0	0
6	0	-0.13	-0.21	-0.16	0	0	-0.15
7	0.38	0	0.47	0	0	0	0
8	0.22	0.13	0	0	0	0.16	0
9	0	0	0	0	0	0	0.88
10	0	0.13	0	0.16	0.21	0	0
11	-0.12	-0.13	-0.21	0	-0.15	0	0
12	0.38	0.40	0	0	0	0	0

Probabilistic Relevance Framework

Table: BM25 term-weighting with relevance information

word	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7
1	0.01	0	0	0	0	0	0
2	0	0.86	0	1.52	0	0	1.01
3	0	0	0	0	0	-0.53	-0.43
4	0	1.22	0	0	2.03	0	0
5	0.01	0	0	0	0	0	0
6	0	0.56	0.93	0.72	0	0	0.66
7	-0.24	0	-0.31	0	0	0	0
8	1.45	0.86	0	0	0	1.10	0
9	0	0	0	0	0	0	0.12
10	0	0.86	0	1.10	1.43	0	0
11	0.52	0.56	0.93	0	0.66	0	0
12	1.14	1.22	0	0	0	0	0

Probabilistic Relevance Framework

IDFP of Toy Model

Doc	6	1	2	4	7	5	3
IDFP	0.165	0.097	0.000	0.000	0.000	-0.150	-0.213

RSJ of Toy Model

Doc	1	2	6	3	5	4	7
RSJ	1.973	1.422	1.104	0.929	0.657	0.000	0.000

RSJ of Toy Model with Expanded Query

Doc	2	1	5	6	3	4	7
RSJ	3.871	3.113	2.686	1.104	0.929	0.000	0.000

Language Model

$p_{ML}(T_i = 1|\mathbf{D})$ and $p_{DF}(T_i = 1)$

word	Doc	1	2	3	4	5	6	7	$p_{DF}(T_i = 1)$
1	alfred	0.125	0	0	0	0	0	0	0.036
2	american	0	0.143	0	0.5	0	0	0.2	0.107
3	award	0	0	0	0	0	0.75	0.4	0.071
4	effect	0	0.143	0	0	0.4	0	0	0.071
5	foundation	0.125	0	0	0	0	0	0	0.036
6	great	0	0.143	0.4	0.25	0	0	0.2	0.143
7	invent	0.125	0	0.2	0	0	0	0	0.071
8	nobel	0.375	0.143	0	0	0	0.25	0	0.107
9	olympics	0	0	0	0	0	0	0.2	0.036
10	physics	0	0.143	0	0.25	0.4	0	0	0.107
11	prize	0.125	0.143	0.4	0	0.2	0	0	0.143
12	science	0.125	0.143	0	0	0	0	0	0.071

Language Model

Each term's $\pi^{(i)}$, EM iterations, document set $\mathcal{D}$ contains all documents

word \ Iter	0	1	2	3	4	5	6	7	8
1	0.50	0.11	0.04	0.02	0.01	0.00	0.00	0.00	0.00
2	0.50	0.29	0.21	0.16	0.13	0.12	0.10	0.10	0.09
3	0.50	0.25	0.20	0.19	0.18	0.18	0.18	0.18	0.18
4	0.50	0.22	0.14	0.10	0.08	0.07	0.06	0.05	0.05
5	0.50	0.11	0.04	0.02	0.01	0.00	0.00	0.00	0.00
6	0.50	0.35	0.27	0.21	0.18	0.15	0.13	0.12	0.10
7	0.50	0.20	0.10	0.06	0.03	0.02	0.01	0.01	0.01
8	0.50	0.29	0.21	0.16	0.13	0.11	0.09	0.08	0.08
9	0.50	0.12	0.06	0.04	0.03	0.02	0.01	0.01	0.01
10	0.50	0.29	0.21	0.16	0.13	0.11	0.10	0.09	0.08
11	0.50	0.33	0.23	0.17	0.13	0.10	0.08	0.07	0.05
12	0.50	0.19	0.09	0.04	0.02	0.01	0.01	0.00	0.00

Language Model

Each term's $\pi^{(i)}$, EM iterations, document set $\mathcal{D}$ contains only relevant documents

word \ Iter	0	1	2	3	4	5
1	0.50	0.00	0.00	0.00	0.00	0.00
2	0.50	0.46	0.45	0.44	0.43	0.43
3	0.50	0.00	0.00	0.00	0.00	0.00
4	0.50	0.51	0.51	0.51	0.51	0.51
5	0.50	0.00	0.00	0.00	0.00	0.00
6	0.50	0.38	0.30	0.24	0.20	0.17
7	0.50	0.00	0.00	0.00	0.00	0.00
8	0.50	0.19	0.08	0.03	0.02	0.01
9	0.50	0.00	0.00	0.00	0.00	0.00
10	0.50	0.69	0.82	0.91	0.95	0.98
11	0.50	0.36	0.27	0.20	0.15	0.12
12	0.50	0.22	0.12	0.07	0.04	0.03

Vector Space Model

Table: Normalized TF*IDF weights of Toy Example

word	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7
1	0.47	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.35	0.00	0.86	0.00	0.00	0.25
3	0.00	0.00	0.00	0.00	0.00	0.98	0.75
4	0.00	0.51	0.00	0.00	0.81	0.00	0.00
5	0.47	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.23	0.55	0.28	0.00	0.00	0.17
7	0.30	0.00	0.62	0.00	0.00	0.00	0.00
8	0.61	0.35	0.00	0.00	0.00	0.22	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.58
10	0.00	0.35	0.00	0.43	0.55	0.00	0.00
11	0.13	0.23	0.55	0.00	0.18	0.00	0.00
12	0.30	0.51	0.00	0.00	0.00	0.00	0.00

Vector Space Model

Table: Standard Rocchio Algorithm, $\alpha = 1$, $\beta = 0.75$ and $\gamma = 0.15$, initial query and modified query

	1	2	3	4	5	6	7	8	9	10	11	12
q0	0	0	0	0	0	0	0	0.71	0	0	0.71	0
qm	-0.03	0.19	0.00	0.28	-0.03	0.10	-0.05	0.66	0.00	0.19	0.59	0.26

Table: Standard Rocchio Algorithm, $\alpha = 1$, $\beta = 0.75$ and $\gamma = 0.15$, ranking by initial query and ranking by modified query

Document	1	2	3	6	5	4	7
simillarity	0.524	0.409	0.392	0.156	0.129	0.000	0.000
Document	2	1	5	3	4	6	7
simillarity	0.789	0.517	0.433	0.347	0.265	0.144	0.063

Thank you!

at SAS, Cary, North Carolina