Scribes: Gilly Leshed, N. Sadat Shami

Outline

1. Review
2. Mixture of Poissons (2 Poisson) model
3. BM25/Okapi method
4. Relevance feedback

1. Review

In discussing probabilistic models for information retrieval we covered the Robertson and Spärck Jones (1976) scoring function (aka the RSJ model). Our own version of the RSJ scoring function for a document *d* is:

$$\prod_{j:q_j>0,a_j(d)>0} \frac{P(A_j = a_j(d) \mid R = \mathrm{y})}{P(A_j = a_j(d))} \cdot \frac{P(A_j = 0)}{P(A_j = 0 \mid R = \mathrm{y})}$$

which is equivalent for ranking purposes to the log thereof:

$$\sum_{j:q_j>0,a_j(d)>0} \log\left(\frac{P(A_j = a_j(d) \mid R = \mathrm{y})}{P(A_j = a_j(d))} \cdot \frac{P(A_j = 0)}{P(A_j = 0 \mid R = \mathrm{y})}\right).$$

Here we assumed binary A_j and following Croft & Harper assumed that $P(A_j = a_j(d) \mid R = y)$ is the same for all query terms [1]. This yielded an *idf* term weighting scheme.

We then tried something more complicated and came up with the 'Isolated Poisson model' where adopting some reasonable assumptions allowed us to obtain a *tf x idf* scheme. We will tweak the probabilistic model even further with the 'Mixture of Poissons' model (commonly known as the 2 Poisson model) and improve performance.

2. Mixture of Poissons model

In order to understand the assumptions behind the 'Mixture of Poissons' model we need to comprehend the idea of 'eliteness' originally proposed by Harter [2]. Occurrences of a term in a document have a random or stochastic element, which nevertheless reflect a real but hidden distinction between those documents that are "about" the topic represented by the term and those that are not. Those documents that are "about" this topic are described as "elite" for the term.

The 'Mixture of Poissons model' assumes that the distribution of within-document frequencies is Poisson for the elite documents, and also (but with a different mean) for the non-elite documents.

We assume term $v^{(j)}$ corresponds to some topic. The question then arises, why does $v^{(j)}$ occur in document *d*?

There are two cases.

Case 1: *d* is about $v^{(j)}$*'s* topic ≡ *d* is 'elite' for $v^{(j)}$

Case 2: occurrence is incidental.

We then look for the degree of match between the query and *d.*
Within the RSJ model, we can introduce a binary random variable

$T_j(d) \equiv T_j = \begin{cases} \text{y, d is about } v^{(j)}\text{'s topic} \\ \text{n, o.w.} \end{cases}$

The fact that $P(B) = \sum_z P(Z = z, B)$, this conveniently allows us to introduce random variable T_j into RSJ.

$$P(A_j = a \mid R = y) = \sum_{t=\{y,n\}} P(A_j = a, T_j = t \mid R = y)$$
$$= \sum_{t=\{y,n\}} P(A_j = a \mid T_j = t, R = y) P(T_j = t \mid R = y)$$

This is essentially a generative model wherein the following independence assumption is justified.

Independence assumption
A_j is conditionally independent of R given T_j, since if the author knows what the document is about, whether they use a term does not depend on whether the document is relevant to the user. Thus $P(A_j = a \mid R = y)$ can be re-written as

$$\sum_{t=\{y,n\}} P(A_j = a \mid T_j = t) P(T_j = t \mid R = y)$$

This allows us to drop one occurrence of a random variable.

Let $P(A_j = a \mid T_j = y) \sim Poisson\ (\tau)$,

where τ is the expected number of occurrences of term $v^{(j)}$ in a document on $v^{(j)}$'s topic for a given document length.

Similarly, $P(Aj = a \mid Tj = n) \sim Poisson\ (\mu)$, for off-topic documents

And $\mu < \tau$, that is, if the document is about the topic that is associated with a term, we expect to see the term in the document more frequently than if the document is not about the topic.

But we don't know μ, τ, or $P(T_j = y \mid R = y)$. We also get an additional unknown variable via $P(A_j = a)$. If we plug all this into the RSJ model, we get a very big expression that is a function of a_j's.

As $a_j \to \infty$ it's monotone increasing to an asymptote, which if $T_j = y$ and $R = y$ were correlated, would be the RSJ weight (idf).

Thus the unwieldy formula can be used to suggest a much simpler formula. We can use this to approximate the weighting function as

$$\left(\frac{aj}{K + aj}\right) \text{x idf}_j$$

where K is an unknown constant.

Length effects

We assumed fixed document length in order to use the Poisson as a suitable model. According to Robertson & Walker, there are at least two reasons why documents may vary in length [3]. Some documents simply cover more topics than others. An opposite view is that some documents are longer because they simply use more words. It seems likely that real document collections contain a mixture of these effects.

The simplest way to take document length into account is to normalize for document length. This can be expressed as

$$k = k_1\left((1-b) + b\frac{length(d)}{average\ \ document\ \ length}\right)$$

where k_1 and b are free parameters.

3. BM25/Okapi model

The symmetry of the retrieval situation as between documents and queries suggests that we could treat within-query term frequency in a similar fashion to within-document term frequency. A query term weighting function could thus be written as:

$$\frac{qtf_j}{k_3 + qtf_j}$$

where k_3 is another constant.

Combining all these leads us to the BM25/Okapi weight scheme. Virtually all the major systems in TREC now use the "BM25/Okapi formula". The following table taken from [4] shows the Okapi scoring function.

tf = the term's frequency in document
qtf = the term's frequency in query
N = the total number of documents in the collection
df = the number of documents that contain the term
dl = the document length (in bytes), and
avdl = the average document length

Okapi weighting based document score[1]:

$$\sum_{t \in Q,D} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1((1-b) + b\frac{dl}{avdl})) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

k_1 (between 1.0-2.0), b (usually 0.75), and k_3 (between 0-1000) are constants.

4. Relevance Feedback

Let's rethink the entire 'Probability Model' approach. Looking back on it, the lack of relevance information has been a big stumbling block. Can we make the hidden variables not hidden? The relevance feedback approach proposed by Rocchio in 1965 provides us an avenue for this [5]. There are two ways to go about this: (a) an explicit approach, and (b) an implicit approach. The explicit approach involves obtaining query dependent relevance labels from the user. The drawback of this approach is that it is expensive for the user. The implicit approach involves using clickthrough data, email, document and webpage views etc. The drawback of this approach is that there are many privacy concerns about providing access to personal documents and web browsing behavior. Perhaps we will cover this concept more extensively in a future lecture.

Next Lecture

In considering the RSJ model, we see that the query is not well integrated. We assumed that the query is fixed. In the next lecture we will cover a technical fix by Lafferty & Zhai '03 where the score is given by $P(R = y \mid D = d, Q = q)$.

References

[1] W. Bruce Croft and D. Harper. Using probabilistic models of information retrieval without relevance information. *Journal of Documentation* 35: 285--295 (1979).
[2] Stephen P. Harter. A probabilistic approach to automatic keyword indexing, part I: On the distribution of specialty words in a technical literature. *Journal of the American Society for Information Science* 26(4) (1975).

[1] Note that the class handout had a typo in Singhal's equation for Okapi/BM25: $(k_1(1\text{-}b)+b(dl/avdl))+tf$ should be $k_1((1\text{-}b)+b(dl/avdl))+tf$. This way, we get k_1+tf when dl=avdl.

[3] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *SIGIR*, pp. 232--241 (1994).
[4] Amit Singhal. Modern Information Retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* 24(4) (2001).
[5] J.J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 313-323, Englewood Cliffs, NJ, 1971.

Exercise: Probabilistic models and BM25/Okapi model

Assume a junior-high school student needs to learn about the Olympic games in ancient Greece. He types into Google the query "olympic greece" and retrieves the following top five results in a descending order (obtained on Feb, 20, 2006, 13:00):

1. http://www.athens2004.com/: the Athens 2004 Olympic games official website.
2. http://www.hol.gr/greece/olympic.htm: a Hellas On-Line webpage with a brief explanation of the origin of the Olympic games in ancient Greece and their revival in modern times.
3. http://education.nmsu.edu/webquest/wq/olympics/olympicwq.html: an educational website for the Olympic games in ancient Greece.
4. http://gogreece.about.com/cs/intlairlines/gr/olympicair.htm: an About.com website providing a review about the Olympic Airways airline.
5. http://195.167.49.234/: The Olympic Airways airline official website.

a) Go to the above links and compare them based on the following factors:
Topic
Relevancy to the needs of the student
Discuss the differences between the documents in terms of these factors.

b) Count the number of times each of the query terms appears in the documents. Define other terms that you think are discriminatory between the topics you identified in (a), and count their frequency in the documents.
Discuss the results in light of the two-Poisson model we discussed in class.

c) Recall from the handout the scoring functions for the BM25/Okapi and pivoted normalization weighting functions (from Singhal 2001):

tf	is the term's frequency in document
qtf	is the term's frequency in query
N	is the total number of documents in the collection
df	is the number of documents that contain the term
dl	is the document length (in bytes), and
$avdl$	is the average document length

Okapi weighting based document score: [23]

$$\sum_{t \in Q,D} ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{k_1((1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

Note the typo correction

k_1 (between 1.0–2.0), b (usually 0.75), and k_3 (between 0–1000) are constants.

Pivoted normalization weighting based document score: [30]

$$\sum_{t \in Q,D} \frac{1 + ln(1 + ln(tf))}{(1-s) + s\frac{dl}{avdl}} \cdot qtf \cdot ln \frac{N+1}{df}$$

s is a constant (usually 0.20).

Use these functions to compute the scores for each of the retrieved documents. Compare the results and discuss them.

For convenience, assume the values of the parameters are: k_1= 1.5; b=0.75; k_3=500; and s=0.20. Additionally, assume that N=1000.

Solution

a) The following table presents the topics of the pages, the rank based on the student's needs.

	Document	Google rank	Topic	Relevancy (user rank)
d1	http://www.athens2004.com/	1	Olympic games (modern)	3
d2	http://www.hol.gr/greece/olympic.htm	2	Olympic games (ancient, modern)	2
d3	http://education.nmsu.edu/webquest/wq/olympics/olympicwq.html	3	Olympic games (ancient)	1
d4	http://gogreece.about.com/cs/intlairlines/gr/olympicair.htm	4	airline	5
d5	http://195.167.49.234/	5	airline	4

As can be seen from the table, there is a discrepancy between how the user would rank the documents and how Google ranked the documents.
The user was interested in the Olympic games, and accurately received the top three documents about this topic. However, although the user was looking for information about the ancient games, the top ranked document was about the recent 2004 games. This perhaps results from the fact that the 2004 games took place in Greece.
Similarly, documents about a Greek airline service, Olympic Airways, were retrieved in ranks 4 and 5, although they are completely off-topic from what the user was looking for. The airline names were perhaps retrieved by Google because they are based in Greece.

b) The following table presents the frequencies of the query terms in each of the documents, along with other terms that could discriminate between the topics identified in (a) and their frequencies.

	Query term frequency		**Other discriminatory terms frequency**			
	olympic	**greece**	**ancient**	**2004**	**games**	**airline**
d1	33	3	0	31	20	0
d2	15	1	3	0	12	0
d3	19	8	15	0	12	0
d4	17	23	0	2	0	4
d5	3	0	0	0	0	2

The table shows that all documents have the query terms in them (except the term "greece" in d5). Following the discussion from the previous question, these terms are expected to appear in each of the topics we identified: the Olympic games, and the Olympic airline.

When looking at the other terms, we notice that some terms are expected to appear in one topic but not in others. This is exactly why we define them as discriminatory: they allow discrimination between the topics. The term "airline", for example, is anticipated in the airline topic documents, but not in the games topic documents. Similarly, the term "games" is expected to appear in the games topic documents, and not in the airline topic documents.
This understanding is conceptualized in the two-Poisson model, in which the probability of term occurrence in a document given that it is on-topic or off-topic can be represented by a Poisson distribution.
Specifically:
$P(A_j = a \mid T_j = y) \sim \text{Poisson}(\tau)$
$P(A_j = a \mid T_j = n) \sim \text{Poisson}(\mu)$

We assumed that $\tau > \mu$, that is, if the document is about the topic that is associated with a term, we expect to see the term in the document more frequently than if the document is not about the topic. This is why we expect to see the word "games" in documents about the games topic more frequently than in documents about the airline topic.

One issue that is apparent in the results is not captured by the two-Poisson model: The model assumes a fixed-document length, an assumption that does not hold in the real world. The following table presents the document length in KB and in number of words. The term "airline" appears in d4 four times and in d5 twice, both about the airline topic. However, d5 is also much smaller than d4, and therefore the difference in the frequency is understandable.

	Length (KB)	Length (#words)
d1	36.7	632
d2	2.86	289
d3	7.18	503
d4	23.7	522
d5	10.7	82

c) BM25/Okapi and pivoted normalization weighting scores:

Note that these functions only use the terms that appear both in the query and in the document.

<u>BM25/Okapi scores calculation:</u>

General terms:

	k1	**b**	**k3**	**N**	**dl (Bytes)**	**avdl (Bytes)**
d1	1.5	0.75	500	1000	36700	16228
d2	1.5	0.75	500	1000	2860	16228
d3	1.5	0.75	500	1000	7180	16228
d4	1.5	0.75	500	1000	23700	16228
d5	1.5	0.75	500	1000	10700	16228

'olympic':

	tf	**df**	**qtf**	$ln\frac{N-df+0.5}{df+0.5}$	$\frac{(k_1+1)tf}{k_1((1-b)+b\frac{dl}{avdl})+tf}$	$\frac{(k_3+1)qtf}{k_3+qtf}$	**product of terms**
d1	33	5	1	5.198497	2.296821	1	11.94001652
d2	15	5	1	5.198497	2.407972	1	12.51783719
d3	19	5	1	5.198497	2.390208	1	12.42548711
d4	17	5	1	5.198497	2.234726	1	11.61721518
d5	3	5	1	5.198497	1.821815	1	9.470700281

'greece':

	tf	**df**	**qtf**	$ln\frac{N-df+0.5}{df+0.5}$	$\frac{(k_1+1)tf}{k_1((1-b)+b\frac{dl}{avdl})+tf}$	$\frac{(k_3+1)qtf}{k_3+qtf}$	**product of terms**
d1	3	4	1	5.400172	1.26706018	1	6.8423426
d2	1	4	1	5.400172	1.58904861	1	8.58113539
d3	8	4	1	5.400172	2.25409238	1	12.172486
d4	23	4	1	5.400172	2.29834578	1	12.4114619
d5	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Okapi weighting scores:

	Sum over terms: $\sum_{t \in Q,D}$ **…**	**Rank**
d1	18.78	4
d2	21.10	3
d3	24.59	1
d4	24.03	2
d5	9.47	5

Pivoted normalization scores calculation:

General terms:

	s	N	dl (Bytes)	avdl (Bytes)
d1	0.2	1000	36700	16228
d2	0.2	1000	2860	16228
d3	0.2	1000	7180	16228
d4	0.2	1000	23700	16228
d5	0.2	1000	10700	16228

‘olympic’:

	tf	**df**	**qtf**	$\frac{1+ln(1+ln(tf))}{(1-s)+s\frac{dl}{avdl}}$	qtf	$ln\frac{N+1}{df}$	**multiplication**
d1	33	5	1	2	1	5.3	10.5931
d2	15	5	1	2.77	1	5.3	14.65925
d3	19	5	1	2.67	1	5.3	14.14942
d4	17	5	1	2.15	1	5.3	11.37274
d5	3	5	1	1.87	1	5.3	9.902204

‘greece’:

	tf	**df**	**qtf**	$\frac{1+ln(1+ln(tf))}{(1-s)+s\frac{dl}{avdl}}$	qtf	$ln\frac{N+1}{df}$	**multiplication**
d1	3	4	1	1.390457	1	5.52	7.678746
d2	1	4	1	1.19725	1	5.52	6.611764
d3	8	4	1	2.391418	1	5.52	13.20651
d4	23	4	1	2.215579	1	5.52	12.23545
d5							

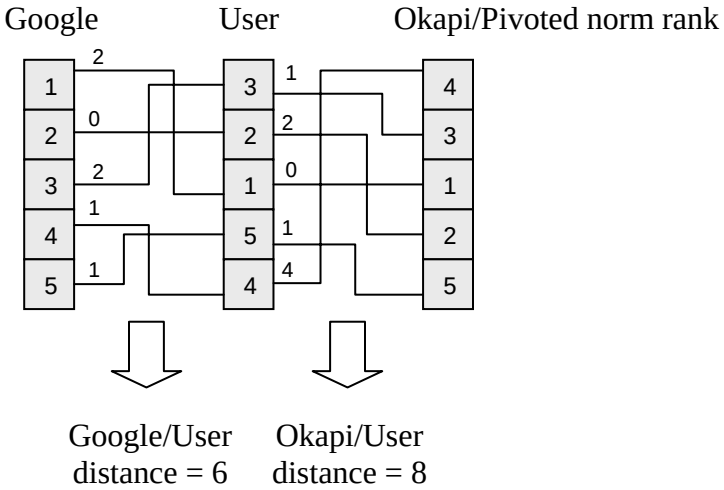
Pivoted normalization weighting scores:

	Sum over terms: $\sum_{t \in Q,D}$ …	**Rank**
d1	18.27	4
d2	21.27	3
d3	27.36	1
d4	23.61	2
d5	9.90	5

Results:

Document	Google rank	Okapi rank	Pivoted norm rank	User rank
d1	1	4	4	3
d2	2	3	3	2
d3	3	1	1	1
d4	4	2	2	5
d5	5	5	5	4

There are few interesting results in the above table:
First, the Okapi ranking and the pivoted normalization ranking are identical. This illustrates how the theoretically-derived probabilistic model achieves results similar to the empirically-derived VSM model.
Second, we can compare the different rankings by measuring how close they are. This can be done by measuring, for each document, the distance in ranking steps between the user and the system and summing the distances over all the documents. The idea is illustrated in the following diagram:



This illustrates that although Google missed the top ranked document for the user while the Okapi/Pivoted normalization ranking identified it, in total, Google's ranking is closer to the user's ranking than the functions we used for the calculations. One reason could be that Google uses additional algorithms beyond simple comparison of the documents and the query, for example working with related terms (e.g. 'Olympic' and 'games' are related) and link analysis.

References:
[1] Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*.