On the distribution of user persistence for rank-biased precision

Laurence A. F. Park
Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010 Australia
lapark@csse.unimelb.edu.au

Yuye Zhang
NICTA Victoria Research Laboratory,
Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010 Australia
zhangy@csse.unimelb.edu.au

Abstract *Rank-biased precision (RBP) is a new method of information retrieval system evaluation that takes into account any uncertainty due to incomplete relevance judgements for a given document and query set. To do so, RBP uses a model of user persistence. In this article, we will present a statistical analysis of the RBP user persistence model to observe how the user persistence value affects the user persistence distribution. We also provide a method of fitting data from existing users to the persistence model, in order to compute their persistence value. Using the Microsoft MSN query log, we were able to demonstrate a typical distribution of the user persistence value and show that it closely resembles a reverse lognormal distribution, with a mean of* $p = 0.78$.

Keywords Evaluation, rank-biased precision, persistence distribution

1 Introduction

To evaluate an information retrieval system, the documents retrieved by the system are compared to a list of relevance judgements for each (document, query) pair using some evaluation metric. Therefore, the evaluation requires manually judging each pair.

As information storage and retrieval algorithms improve, and computer processing power and storage grows, so too does the expected number of documents indexed by a retrieval system. A problem that is encountered by researchers in the information retrieval field is the manual judgement of each (document, query) pair when faced with large document collections. A simple method of dealing with these large sets is to manually judge only a subset of the documents for each query, where the subset hopefully contains most of the documents relevant to that query. When evaluating a system, if a document is retrieved that was not in the judged set, it can be assumed irrelevant to the query. Although this method still provides reliable results, overall accuracy has been compromised due to uncertainty stemming from the unjudged documents.

A new method of information retrieval evaluation called rank-biased precision (RBP) [3] deals with unjudged documents by offering uncertainty in the evaluation. Therefore, the evaluation score provides a range covering the evaluation scores that would have been obtained if the unjudged documents were relevant or irrelevant.

RBP evaluation is based on a user persistence model that requires the choice of a user persistence value before the evaluation can take place. In this article, we examine the statistical properties of the user persistence model and the effect of choosing a certain user persistence value. We also examine how we can deduce the persistence value to suit a desired audience, and hence model that audience. This article makes the following contributions:

- a statistical analysis of the properties of the user persistence model and effect of the user persistence value (p).
- a method of computing the persistence value (p) from a query log in order to model a set of users
- an analysis showing that the user persistence (p) is a reverse lognormal distribution when modelled over a large audience.

The article will proceed as follows: in section 2 we will discuss the method of rank-biased precision and its associated user persistence model. In section 3 we will analyse the user persistence distribution and examine the effect of changing the user persistence value (p). Section 4 describes a method of modelling the user persistence value when given an appropriately detailed query log. Finally, in section 5 we will examine the

Proceedings of the 12th Australasian Document Computing Symposium, Melbourne, Australia, December 10, 2007.

distribution of the user persistence value (p) computed from a query log provided by Microsoft.

2 Evaluation of large documents collections

In this section, we examine a new method of evaluating retrieval systems called rank-biased precision (RBP) that allows us to easily account for uncertainty in relevance judgements. We begin the section by outlining the method in which relevance judgements are obtained for large document collections and examine the problems associated to existing popular retrieval system metrics that are induced by the judgement process. We then introduce RBP and show how it is more suited to dealing with the uncertainty in modern document collection relevance judgements.

2.1 Obtaining relevance judgements

The ideal method of evaluating a retrieval system is to obtain a document set, a set of queries and a relevance judgements of every document for each query. Relevance judgements are assigned to each (document, query pair) manually, implying that human judges must examine every document for each query and assign a relevance score (usually 1 for relevant and 0 for irrelevant). Once these scores are obtained, the system in question is used to rank each document for each query and the results are compared to the relevance judgements using some pre-defined metric.

Modern retrieval experiments are performed on document collections containing millions of documents, so unfortunately, human relevance judgements are not possible for every (document, query pair). To obtain an estimate of the most relevant documents, TREC[1] have implemented a pooling method of document evaluation in an attempt to obtain the best estimate of relevant documents per query [4]. For each of the retrieval systems taking part in TREC, the set of top ranked documents for a given query are placed into a pool. Therefore, the pool for a query will contain the set of documents that have been highly ranked by each of the retrieval systems. The pool is then considered as a set of candidate documents that should contain most of the documents relevant to each query. Each of these documents are then manually judged and the documents that do not appear in the pool are considered irrelevant to the query. Unfortunately, the pooling method places great importance on the initial set of retrieval systems that are used, since any documents that are not added to the pool (that is, not highly ranked by any retrieval system) are considered irrelevant.

2.2 Uncertainty in relevance judgements

Rather than assuming that unjudged documents are irrelevant, we should simply take into account this uncertainty during the system evaluation process. For example, if a query retrieves documents that all have associated relevance judgements, we should be able to precisely evaluate the system, but if one or more documents do not have relevance judgements, then the evaluation should contain an associated error margin depicting the uncertainty.

Unfortunately, many of the popular information retrieval metrics do not allow for this uncertainty. It has been shown that scores produced by retrieval metrics such as mean average precision (MAP) and bpref [1] can provide drastically different scores when uncertainty is introduced due to their dependence on the number of documents relevant to each query. Consequently, such metrics are unsuited for handling uncertainty within the retrieved results in these instances.

2.3 Rank-biased precision

A new metric called rank-biased precision (RBP) [2, 3] has been designed to take into account uncertainty in relevance judgements. RBP has shown to provide error margins that converge as the uncertainty reduces.

RBP is designed around the model that a user examining a list of retrieved documents will start from the top ranked document and when examining each document, will proceed to the next document with a probability p, or finish the search with probability $1 - p$. The score provided to the system increases if a user examines a relevant document, therefore the RBP is computed as the sum of the probability of examining each relevant document:

$$RBP(p) = (1-p)\sum_{i=1}^{\infty} r_i p^{(i-1)} \qquad (1)$$

where $r_i \in [0, 1]$ is the relevance judgement of the ith ranked document, and the $(1-p)$ factor is used to scale the RBP within the range $[0, 1]$. The probability of a user examining the next document is also the *persistence* of the user. We can see that a user with low persistence (p close to zero) is not likely to examine past the first document, while a user with a high persistence value (p close to 1) is likely to examine many documents.

We will now provide an example of how RBP is used to obtain an intuition of how uncertainty is dealt with. Given a user with persistence of $p = 0.5$, if a given system returns a ranked document list such that the ranked document have the associated relevance judgements $(1, 1, 0, 1, ?, 0, 0, 1)$, where 1 denotes relevance, 0 denotes irrelevance and ? denotes not judged, then the RBP is computed as: $(1 - 0.5) \times (0.5^0 + 0.5^1 + 0.5^3 + 0.5^7) = 0.816$. By taking into account the uncertain relevance judgements, we can compute the uncertainty in the RBP as: $(1 - 0.5) \times (0.5^4 + \sum_{i=9}^{\infty} 0.5^{i-1}) = (1 - 0.5) \times 0.5^4 + 0.5^8 = 0.0352$. Implying that the RBP lies within the bounds $[0.816, 0.852]$, due to the uncertainty produced by the pooling process. We can see from these steps that as the number of relevance

[1] http://trec.nist.gov

judgements increase, the number of uncertain relevance judgements decrease and hence the uncertainty in the RBP decreases.

The RBP metric relies on the user persistence model and hence on the choice of the user persistence (p). In the remainder of this article we will examine properties of this persistence distribution and the effect of p. We will also examine how to choose p, when given a set of users' statistics.

3 The distribution of user persistence

Once a user has received a ranked list of search results, the typical behaviour is to scan from the top of the list to the bottom of the list in the hope that a document relevant to his or her information need is found. If a document appears to be relevant, based on its title or snippet, the user will open the document and examine it further. When the user believes that there will be no more relevant documents further down the ranked list, the user will finish examining the list. We can treat the depth at which the user stops examining the ranked list as the user's persistence. For example, a user that only examines the first two documents will have a low persistence value, while a user that examines the first twenty documents will have a higher persistence value.

The rank-biased precision evaluation metric uses the assumption that a user has a specific persistence p, which is the probability of examining the next document in the ranked list. Therefore, if we begin from the top of the list, the probability of examining the ith document is:

$$P(E = i|p) = p^{i-1} \tag{2}$$

Given that the probability of examining the next document is p, we can deduce that the probability of not examining further documents to be $1-p$. Therefore, the probability that a user examines the first i documents in the list but no more is:

$$P(L = i|p) = p^{i-1}(1-p) \tag{3}$$

We will now refer to this probabilistic distribution as the *persistence distribution*. Figure 1 provides an example of the persistence distribution for $p = 0.5, 0.8$ and 0.9. Throughout this article, we investigate the properties of the persistence distribution and examine how to select the persistence value p for a given user population. To obtain further understanding of the persistence distribution properties, we will derive its mean and variance, and examine the shape of the distribution.

The mean value of the persistence distribution (or the expected number of documents examined), derived in appendix A.1, is:

$$\mu_L(p) = \frac{1}{1-p} \tag{4}$$

where $\mu_L(p)$ is the expected number of documents examined before leaving the search list ($\mathbb{E}[L]$), given a user's persistence (p).

Persistence (p)	$\mu_L(p)$	$\sigma_L(p)$
0	1	0
0.5	2	1.414
0.666	3	2.449
0.75	4	3.464
0.8	5	4.472
0.833	6	5.477
0.857	7	6.481
0.875	8	7.483
0.888	9	8.485
0.9	10	9.487
0.95	20	19.494
0.98	50	49.497
0.99	100	99.499
1.0	∞	∞

Table 1: A list of user persistence values (p) and the associated expected number of pages examined in a ranked list of search results ($\mu_L(p)$) and standard deviation ($\sigma_L(p)$).

The standard deviation ($\sigma_L(p)$) of the persistence distribution (or the expected difference in the number of documents examined and the expected number of documents examined) derived in appendix A.2, is:

$$\sigma_L(p)^2 = \frac{p}{(1-p)^2} \tag{5}$$

where $\sigma_L(p)^2$ is the variance associated to the number of documents examined before leaving the search list, when given a user's persistence (p). Table 1 provides a list of user persistence values and the associated mean and standard deviation.

From the equations of mean and standard deviation, we can see that the standard deviation approaches the mean as the persistence value approached infinity:

$$\mu_L(p) = \lim_{p \to \infty} \sigma_L(p) \tag{6}$$

This implies that it is harder to predict the number of pages examined by more persistent users (users who are expected to examine many documents), due to the high standard deviation.

4 Modelling user persistence

To examine persistence, we must examine how many documents a user will examine before leaving the query.

4.1 Satisfied and unsatisfied queries

When examining the behaviour of a search engine user, we find that by observing the rank of the final document inspected does not necessarily provide us with a good estimate of the user's persistence. For example if a user, who was very persistent, found the relevant document located at rank one, they would have no reason to examine the list further. On the other hand if a very persistent user was presented with a list with no relevant documents, that user would examine many of the documents in the list. A user with a low persistence

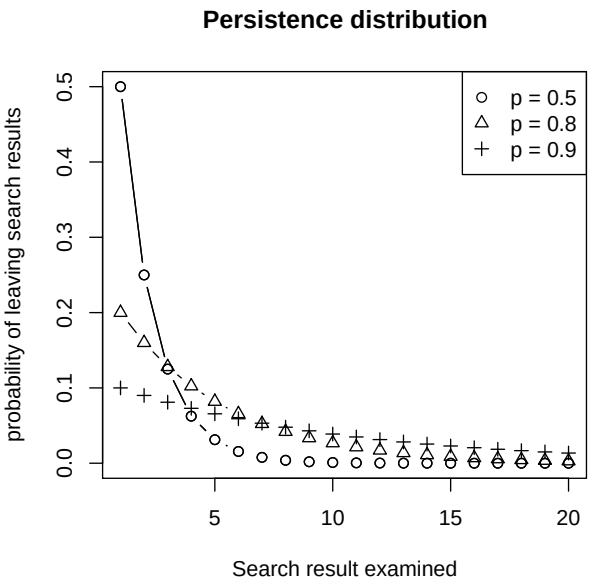

Figure 1: The persistence distribution for persistence values, $p = 0.5$, 0.8 and 0.9 on the first twenty ranked search results.

value would examine only a few documents, regardless of the relevance of those documents. So we can see from this example that if we did not take into account the relevance of the returned documents, any modelling would be biased towards a low persistence value.

As an attempt to remove bias caused by a user who stops examining documents because a relevant document is found, we split the set of queries into two. The first set, labelled *Unsatisfactory*, contains all of the queries which did not satisfy the user's information need before the user's persistence wore out. The second set, labelled *Satisfactory*, contains all queries where the user found a relevant document and hence finished scanning the list even though the user's persistence had not worn out.

From the Unsatisfactory set, we can find where the users persistence had run out by observing the last document the user examined for each query. We can compute the users persistence by fitting the model in equation 3, which involves computing the p that provides the maximum likelihood:

$$L(p) = \prod_{i \in U} p^{i-1}(1-p) \tag{7}$$

where U is the Unsatisfactory set of queries.

Computing the persistence from the Satisfactory set is not as simple , since we do not have a measure of where the user's persistence ended, and their information need was satisfied. The only information we have is how far down the results list the user examined until a relevant document was found. Therefore, in order to compute the persistence, we need to perform survival analysis.

The survival function shows that probability of an event occurring after a certain point, in our case it is the probability of the users persistence wearing out after the ith document ($P(L > i)$). To compute the survival function, we must first compute the cumulative probability function ($P(L \leq i)$). The probability that the user gives up at or before rank i is:

$$P(L \leq i) = \sum_{x=1}^{i} p^{x-1}(1-p) \tag{8}$$

$$= (1-p)\sum_{x=1}^{i} p^{x-1} \tag{9}$$

$$= (1-p)\left(\frac{1-p^i}{1-p}\right) \tag{10}$$

$$= 1 - p^i \tag{11}$$

Therefore the survival function is:

$$P(L > i) = 1 - P(L \leq i) \tag{12}$$

$$= 1 - (1 - p^i) \tag{13}$$

$$= p^i \tag{14}$$

4.2 Maximum likelihood estimation of persistence

If we assume that for each session (where the user issues one or more queries to the retrieval system), there is a set of queries that satisfy the user and a set that do not, then we can compute the persistence of the user using the likelihood function:

$$L(p) = \prod_{i \in \mathrm{U}} P(L = i|p) \prod_{j \in \mathrm{S}} P(L > i|p) \tag{15}$$

$$= \prod_{i \in \mathrm{U}} p^{i-1}(1-p) \prod_{j \in \mathrm{S}} p^j \tag{16}$$

where the first product is associated to the set of unsatisfactory queries U and the second product is associated to the set of satisfactory queries S.

Therefore, given a set of queries from a user and the last document examined for each query, the p associated to the user provides the maximum value for the likelihood function $L(p)$. To obtain the maximum, we must find where the derivative of the likelihood function is zero.

Rather than work with the products in this likelihood function, we are able to work with sums in the log-likelihood function, since $\log()$ is a monotonically increasing function (shown in appendix B.1):

$$l(p) = \log(L(p)) = \log\left(\prod_{i \in U} p^{i-1}(1-p) \prod_{j \in S} p^j\right)$$

$$= \sum_{i \in U} (i-1)\log(p) + \sum_{i \in U} \log(1-p) + \sum_{j \in S} j \log(p)$$

By differentiating with respect to p, we are able to locate the turning point of $l(p)$ and hence derive the equation for the most likely value of the persistence p (shown in appendix B.2):

$$p = \frac{\sum_{i \in U}(i-1) + \sum_{j \in S} j}{\sum_{i \in U \cup S} i} \tag{17}$$

where U is the set ranks of final documents examined in each search when the document was unsatisfactory to the user's information need, and S is the set ranks of final documents examined where the documents were satisfactory to the user's information need. Therefore $U \cap S = \emptyset$, the empty set.

5 Experiments

In the previous section we demonstrated how to model a set of users based on their usage history. In this section we use the modelling methods we derived to compute the persistence values for a set of typical Web search engine users. We begin by describing the data that is used and follow with the modelling of the data.

5.1 MSNSearch query log statistics

The Microsoft MSNSearch query log [5] consists of 5,684,599 user sessions, where each session consists of one or more queries to the Microsoft search engine, from a particular user. To examine the distribution of the persistence value across the set of users, we first must identify each user. Unfortunately, the queries have been anonymized, therefore we must assume that each session is associated to a unique user.

Additionally, each query is associated to a set of clickthrough information, which shows rankings of the search results that were been clicked on for that query instance. So for any given query, there may be zero or many associated clickthroughs.

To model the persistence of individual users, we obtained an estimate of p for each session (correlating to a unique user) using the maximum likelihood method. A histogram of the session lengths is shown in figure 2. We can see from the log scale for frequency, that there is an exponential decay in the frequency of sessions of length n as n increases. This implies that there is a very large proportion of sessions that contain only one query.

5.2 User modelling process

We showed earlier that we are able to compute the maximum likelihood estimate of the user persistence (p) from sampling the rank of lowest ranked document examined per query. Therefore, to compute a single user's persistence from the MSN query log, we must:

- select the the associated session
- compile the set C containing the lowest ranked clickthrough for each query in the session
- split the set C into the set of unsatisfactory queries U and satisfactory queries S

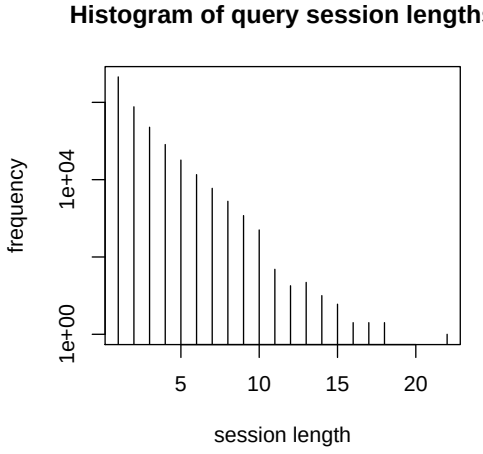

Figure 2: A histogram showing the distribution of query session lengths, using a log scale for the frequency. The linearity of the histogram shows that the frequency decreases exponentially as the session length increases.

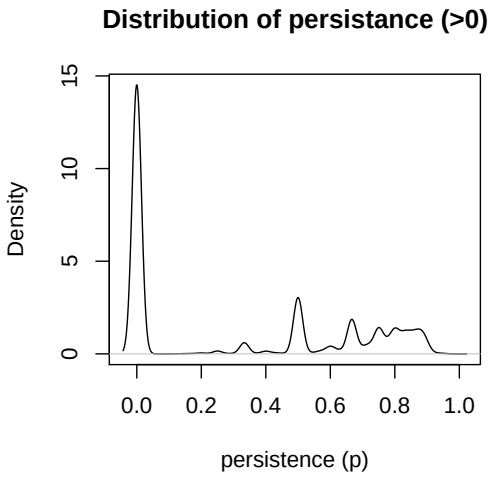

Figure 3: The kernel density estimate of the persistence (p) distribution for all sessions, assuming that all queries were unsatisfactory.

- use U and S to compute the maximum likelihood value of the user persistence from equation 17

Once this is done for each user, we are able to plot the distribution of the fitted persistence values across all users.

5.3 All queries unsatisfactory

We first examined the distribution of p using the assumption that all of the the queries in each session were unsatisfactory. The resulting distribution is shown in figure 3. The distribution shows a large peak at $p = 0$, implying that the majority of users examine only the top ranked document before beginning a new search.

This result maybe an artifact of the many sessions of length one found in the query logs, where the session

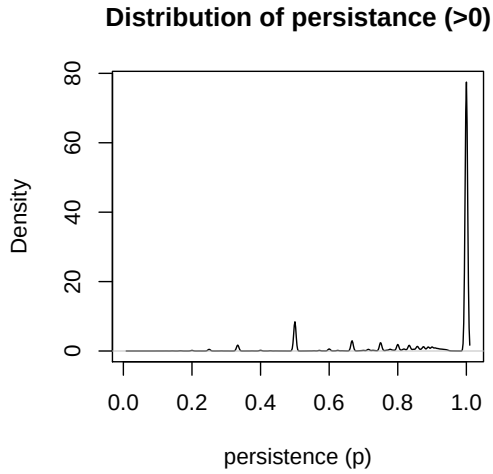

Figure 4: The kernel density estimate of the persistence (p) distribution for all sessions, assuming that all but the last query for the session was unsatisfactory.

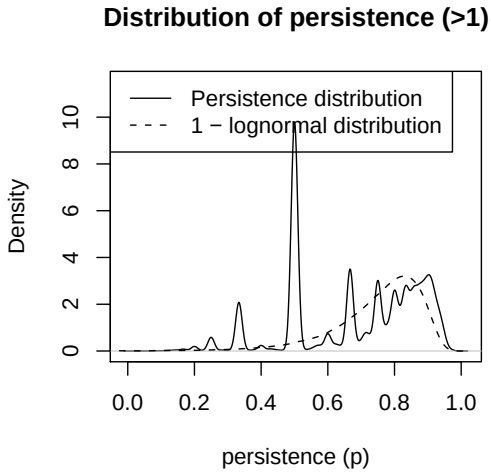

Figure 5: The kernel density estimate of the persistence (p) distribution for all sessions containing at least two search attempts, assuming that all but the last query for the session was unsatisfactory.

shows the user examining only the top ranked document.

5.4 Last queries satisfactory

If a user examines only one document and finishes the session, we have to ask why the user did not issue another search. We may imply from this that the user was satisfied with the search results. From this we can assume that each session is finished with the user being satisfied with the search results. Therefore, we have generated a distribution for p where all but the last queries of each session are unsatisfactory and the last query is satisfactory. The distribution of p is shown in figure 4.

We can see that the distribution now has a large proportion of users with $p = 1$. We can easily show that this artifact is also due to the sessions of length one. If we assume that the last query in each session is satisfactory, the sessions of length one only contain one satisfactory query and no unsatisfactory queries. Therefore the maximum likelihood estimate of p reduces to:

$$p = \frac{i}{i} = 1 \tag{18}$$

where i is the lowest ranked document examined for the satisfactory query. With no unsatisfactory queries, we are unable to compute p, since we have no cases where the users persistence has worn out. Therefore to obtain a better estimate of p, we will compute its distribution using all session of length two or greater. The distribution is shown in figure 5. This figure, we can see in three large spikes at approximately $p = 0.5$, 0.66 and 0.75. Amongst these spikes, we can see a smooth curve following a reverse lognormal distribution that peaks at around $p = 0.9$.

To obtain an estimate of p for a user, we need instances of many issued queries and the resulting last document examined for each query. Therefore the smaller the number of queries per session, the rougher the estimate of p obtained for the user. By choosing only those sessions that contain more than n queries, we will be able to obtain a better estimate of p for the chosen sessions, but we will also be sampling a subset of the population. Table 2 shows the number of sessions used when constrained to a minimum session lengths.

Session length	Total sessions
> 0	5,684,599
> 1	1,675,949
> 2	679,111
> 3	321,272

Table 2: Total number of sessions in the query log with a certain session length.

To examine the effect of using only sessions of length greater than two and three on the user persistence, we have provided the distributions for these data subsets in figures 6 and 7. We can see that as we increase the session length threshold, the resulting persistence distribution becomes smoother and becomes more like a reversed lognormal distribution. We can also see the large spike that was at 0.5 in figure 5, move towards $p = 0$ as we limit our analysis to longer sessions. This spike is to due to the set of sessions where the user has clicked on the top ranked result only, providing a p of $1/$session length.

It is interesting to note that the fitted reversed lognormal distribution provides a mean close to $p = 0.78$ and standard deviation that provides a 95% confidence interval of $(0.387, 0.920)$.

6 Conclusion

Rank-biased precision (RBP) is a new method of information retrieval system evaluation that takes into

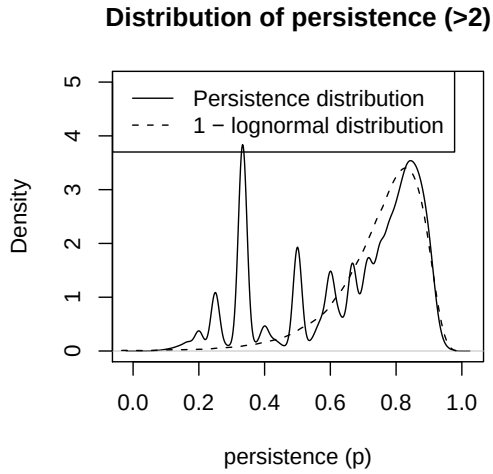

Figure 6: The kernel density estimate of the persistence (p) distribution for all sessions containing at least three search attempts, assuming that all but the last query for the session was unsatisfactory.

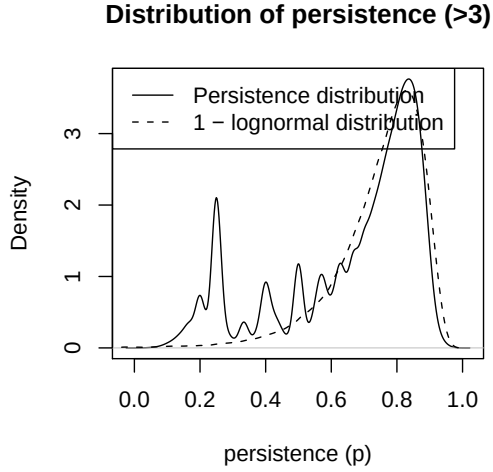

Figure 7: The kernel density estimate of the persistence (p) distribution for all sessions containing at least four search attempts, assuming that all but the last query for the session was unsatisfactory.

account any uncertainty due to incomplete relevance judgements for a given document and query set. To do so, RBP uses a model of user persistence.

In this article, we presented a statistical analysis of the user persistence model to observe how the user persistence value affects the user persistence distribution. We followed this with a method of modelling the user persistence value from user statistics.

Using the Microsoft MSN query log, we were able to demonstrate a typical distribution of the user persistence value and show that it closely resembles a reverse lognormal distribution, with a mean of $p = 0.78$.

References

[1] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM Press.

[2] Alistair Moffat, William Webber and Justin Zobel. Strategic system comparisons via targeted relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 375–382, New York, NY, USA, 2007. ACM Press.

[3] Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. *Under review*, 2007.

[4] Ellen M. Voorhees and Donna K. Harman (editors). *The Ninth Text REtrieval Conference (TREC-9)*, Gaithersburg, Md. 20899, December 2000. National Institute of Standards and Technology Special Publication 500-249, Department of Commerce, National Institute of Standards and Technology.

[5] Yuye Zhang and Alistair Moffat. Some observations on user search behavior. In *Proc. 11th Australasian Document Computing Symposium*, pages 1–8, 2006.

A Derivation of the persistence distribution statistical properties

In this section we derive the mean and variance of the persistence distribution. To perform the derivations, we use the equation:

$$\sum_{i=0}^{\infty} p^i = \frac{1}{1-p} \tag{19}$$

A.1 Expected pages examined

The expected number of pages examined is simply the sum of the page rank times the probability of examining the page:

$$\begin{aligned}
\mathbb{E}[L] &= \sum_{i=1}^{\infty} i p^{i-1}(1-p) \\
&= (1-p)\sum_{i=1}^{\infty} i p^{i-1} \\
&= (1-p)\sum_{i=1}^{\infty} \frac{d(p^i)}{dp} \\
&= (1-p)\frac{d(\sum_{i=1}^{\infty} p^i)}{dp} \\
&= (1-p)\frac{d(\sum_{i=0}^{\infty} p^i - p^0)}{dp} \\
&= (1-p)\frac{d(1/(1-p)-1)}{dp} \\
&= (1-p)\frac{1}{(1-p)^2} \\
&= \frac{1}{(1-p)}
\end{aligned}$$

A.2 Variance of pages examined

The variance is the mean deviation of the pages examined from the expected number of pages examined. This can be simplified to:

$$\sigma^2 = \mathbb{E}[L^2] - \mathbb{E}[L]^2 \tag{20}$$

where σ^2 is the variance. Therefore, to compute the variance, we must first obtain the value of $\mathbb{E}[L^2]$:

$$\begin{aligned}
\mathbb{E}[L^2] &= \sum_{i=1}^{\infty} i^2 p^{i-1}(1-p) \\
&= (1-p)\sum_{i=1}^{\infty} i^2 p^{i-1} \\
&= (1-p)\sum_{i=1}^{\infty}\left[\frac{d^2(p^{i+1})}{dp^2} - \frac{d(p^i)}{dp}\right] \\
&= (1-p)\left[\frac{d^2(\sum_{i=1}^{\infty} p^{i+1})}{dp^2} - \frac{d(\sum_{i=1}^{\infty} p^i)}{dp}\right] \\
&= (1-p)\left[\frac{d^2(\sum_{j=2}^{\infty} p^j)}{dp^2} - \frac{d(\sum_{i=1}^{\infty} p^i)}{dp}\right] \\
&= (1-p)\left[\frac{d^2(\sum_{j=0}^{\infty} p^j - p^0 - p^1)}{dp^2} - \right. \\
&\quad \left. \frac{d(\sum_{i=0}^{\infty} p^i - p^0)}{dp}\right] \\
&= (1-p)\left[\frac{d^2(1/(1-p) - 1 - p)}{dp^2} - \right. \\
&\quad \left. \frac{d(1/(1-p) - 1)}{dp}\right] \\
&= (1-p)\left[\frac{d(1/(1-p)^2 - 1)}{dp} - \frac{1}{(1-p)^2}\right] \\
&= (1-p)\left[\frac{2}{(1-p)^3} - \frac{1}{(1-p)^2}\right] \\
&= \frac{2}{(1-p)^2} - \frac{1}{(1-p)}
\end{aligned}$$

where $j = i + 1$. Using the value of $\mathbb{E}[L^2]$, we can compute the variance (σ^2) using the mean shown in appendix A.1:

$$\begin{aligned}
\sigma^2 &= \mathbb{E}[L^2] - \mathbb{E}[L]^2 \\
&= \frac{2}{(1-p)^2} - \frac{1}{(1-p)} - \frac{1}{(1-p)^2} \\
&= \frac{1}{(1-p)^2} - \frac{1}{(1-p)} \\
&= \frac{1}{(1-p)^2} - \frac{1-p}{(1-p)^2} \\
&= \frac{p}{(1-p)^2}
\end{aligned}$$

B Derivation of maximum likelihood value of persistence

B.1 Simplification of log-likelihood

The log-likelihood function (equation 16) is simplified using the following process:

$$\begin{aligned}
l(p) = \log(L(p)) &= \log\left(\prod_{i\in U} p^{i-1}(1-p)\prod_{j\in S} p^j\right) \\
&= \sum_{i\in U}\log\left(p^{i-1}(1-p)\right) + \sum_{j\in S}\log\left(p^j\right) \\
&= \sum_{i\in U}\log\left(p^{i-1}\right) + \sum_{i\in U}\log(1-p) + \\
&\quad \sum_{j\in S}\log\left(p^j\right) \\
&= \sum_{i\in U}(i-1)\log(p) + \sum_{i\in U}\log(1-p) + \\
&\quad \sum_{j\in S} j\log(p)
\end{aligned}$$

B.2 Maximum of log-likelihood function

To obtain the maximum p for the given log-likelihood function $l(p)$, we must first find the derivative of the log-likelihood function:

$$\frac{dl(p)}{dp} = \frac{\sum_{i\in U}(i-1)}{p} + \frac{\sum_{i\in U} -1}{1-p} + \frac{\sum_{j\in S} j}{p} \tag{21}$$

By equating the derivative to zero and solving for p, we obtain the equation to compute the maximum likelihood of p:

$$\begin{aligned}
&\Rightarrow \frac{\sum_{i\in U} 1}{1-p} = \frac{\sum_{i\in U}(i-1) + \sum_{j\in S} j}{p} \\
&p\sum_{i\in U} 1 = (1-p)\left(\sum_{i\in U}(i-1) + \sum_{j\in S} j\right) \\
&p\sum_{i\in U} 1 + p\left(\sum_{i\in U}(i-1) + \sum_{j\in S} j\right) = \sum_{i\in U}(i-1) + \sum_{j\in S} j \\
&p\left(\sum_{i\in U} 1 + \sum_{i\in U}(i-1) + \sum_{j\in S} j\right) = \sum_{i\in U}(i-1) + \sum_{j\in S} j \\
&p\left(\sum_{i\in U} i + \sum_{j\in S} j\right) = \sum_{i\in U}(i-1) + \sum_{j\in S} j \\
&p\sum_{i\in U\cup S} i = \sum_{i\in U}(i-1) + \sum_{j\in S} j \\
&p = \frac{\sum_{i\in U}(i-1) + \sum_{j\in S} j}{\sum_{i\in U\cup S} i}
\end{aligned}$$