Interactive Exploratory Search for Multi Page Search Results

Xiaoran Jin and Marc Sloan[*]and Jun Wang
Department of Computer Science
University College London
ucabxij@ucl.ac.uk, {M.Sloan, J.Wang}@cs.ucl.ac.uk

ABSTRACT

Modern information retrieval interfaces typically involve multiple pages of search results, and users who are recall minded or engaging in exploratory search using ad hoc queries are likely to access more than one page. Document rankings for such queries can be improved by allowing additional context to the query to be provided by the user herself using explicit ratings or implicit actions such as clickthroughs. Existing methods using this information usually involved detrimental UI changes that can lower user satisfaction. Instead, we propose a new feedback scheme that makes use of existing UIs and does not alter user's browsing behaviour; to maximise retrieval performance over multiple result pages, we propose a novel retrieval optimisation framework and show that the optimal ranking policy should choose a diverse, exploratory ranking to display on the first page. Then, a personalised re-ranking of the next pages can be generated based on the user's feedback from the first page. We show that document correlations used in result diversification have a significant impact on relevance feedback and its effectiveness over a search session. TREC evaluations demonstrate that our optimal rank strategy (including approximative Monte Carlo Sampling) can naturally optimise the trade-off between exploration and exploitation and maximise the overall user's satisfaction over time against a number of similar baselines.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Relevance feedback; Search process; Selection process

Keywords

Interactive Ranking and Retrieval; Exploratory Search; Diversity

1. INTRODUCTION

A well established problem in the traditional Information Retrieval (IR) paradigm of a user issuing a query to a search system, is that a user is quite often unable to precisely articulate their information need in a suitable query. As such, over the course of their search session they will often resort to engaging in *exploratory search*, which usually involves a short, probing query that loosely encapsulates their need, followed by a more rigorous exploration of the search documents and finally the extraction of information [18].

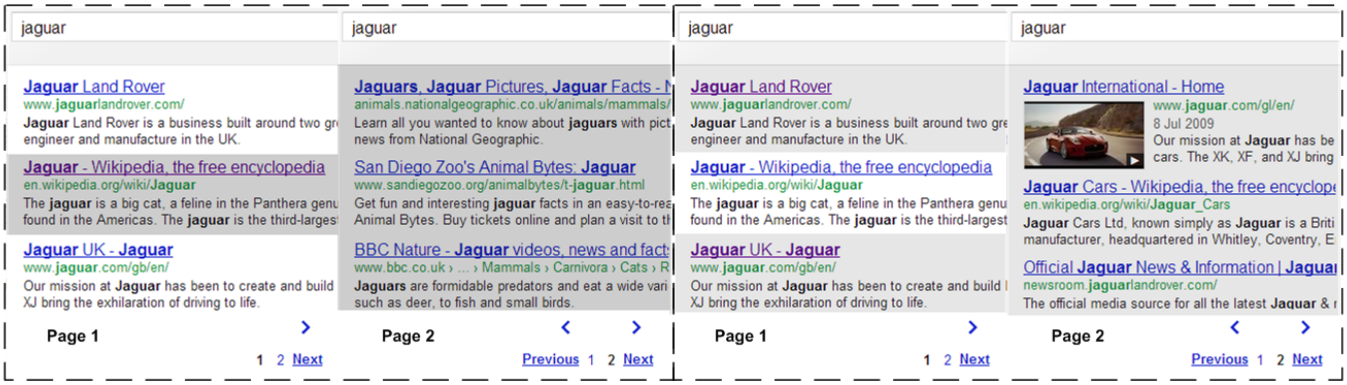
Such short queries are often ambiguous and highly dependent on context; a common example being the query "jaguar", which can be interpreted as an animal, a car manufacturer or a type of guitar. Recent research has addressed this problem by introducing the idea of *diversity* into search ranking, where the objective is to maximise the satisfaction of all users of a search system by displaying search rankings that reflect the different interpretations of a query [2].

An early ranking algorithm employing diversity is the Maximal Marginal Relevance technique [8], whereby existing search results would be re-ranked using a similarity criterion that sought to repel lexically similar documents away from one another. A more modern approach is Radlinski's [21] application of multi-armed bandit algorithms to the online diversification of search results. A common problem found in diversity research is in the evaluation of diversifying algorithms, this is due to the fact that the Probability Ranking Principle (PRP) [25] no longer applies in such cases due to the lack of independence between documents, and as such, common IR metrics such as MAP and nDCG must be interpreted carefully [11]. More recently, the PRP has been generalised to the diverse case by applying portfolio theory from economics in order to diversify documents that are dependent on one another [37].

Consequently, there exists a balance between novelty and relevance [8] which can vary between queries and users. One way to resolve this is to introduce elements of relevance feedback into the search process so as to infer the context for the ambiguous query. In relevance feedback, the search system extracts information from a user in order to improve its results, this can be: *explicit feedback* such as ratings and judgements given directly by the user, for example the well known Rocchio algorithm [26]; *implicit feedback* such as clickthroughs or other user actions; or *pseudo relevance feedback*, where terms from the top n search results are used to further refine a query [7]. A problem with relevance feedback methods is that historically, users are reluctant to respond to them [19]. An analysis of search log data found that typically less than 10% of users made use of an available relevance feedback option during searches, even when results were often over 60% better as a result [33]. In addition, modern applications of the technique such as pseudo relevance feedback have come under criticism [5]. Nonetheless, it has proven effective in areas such as image and video retrieval [27, 42].

[*]The first two authors have equal contribution to this work.

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.



jaguar

Jaguar Land Rover
www.**jaguar**landrover.com/
Jaguar Land Rover is a business built around two gr
engineer and manufacture in the UK.

Jaguar - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Jaguar**
The **jaguar** is a big cat, a feline in the Panthera genu
found in the Americas. The **jaguar** is the third-larges

Jaguar UK - **Jaguar**
www.**jaguar**.com/gb/en/
Our mission at **Jaguar** has been to create and build
XJ bring the exhilaration of driving to life.

Page 1
1 2 Next

jaguar

Jaguars, **Jaguar** Pictures, **Jaguar** Facts - N
animals.nationalgeographic.co.uk/animals/mammals/
Learn all you wanted to know about **jaguars** with pict
news from National Geographic.

San Diego Zoo's Animal Bytes: **Jaguar**
www.sandiegozoo.org/animalbytes/t-**jaguar**.html
Get fun and interesting **jaguar** facts in an easy-to-rea
Animal Bytes. Buy tickets online and plan a visit to th

BBC Nature - **Jaguar** videos, news and fact
www.bbc.co.uk › ... › Mammals › Carnivora › Cats › R
Jaguars are formidable predators and eat a wide vari
such as deer, to fish and small birds.

Page 2
Previous 1 2 Next

jaguar

Jaguar Land Rover
www.**jaguar**landrover.com/
Jaguar Land Rover is a business built around two gre
engineer and manufacture in the UK.

Jaguar - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Jaguar**
The **jaguar** is a big cat, a feline in the Panthera genu
found in the Americas. The **jaguar** is the third-largest

Jaguar UK - **Jaguar**
www.**jaguar**.com/gb/en/
Our mission at **Jaguar** has been to create and build
XJ bring the exhilaration of driving to life.

Page 1
1 2 Next

jaguar

Jaguar International - Home
www.**jaguar**.com/gl/en/
8 Jul 2009
Our mission at **Jaguar** has be
cars. The XK, XF, and XJ bring

Jaguar Cars - Wikipedia, the free encyclope
en.wikipedia.org/wiki/**Jaguar**_Cars
Jaguar Cars Ltd, known simply as **Jaguar** is a Briti
manufacturer, headquartered in Whitley, Coventry, E

Official **Jaguar** News & Information | **Jaguar**
newsroom.**jaguar**landrover.com/
The official media source for all the latest **Jaguar** & r

Page 2
Previous 1 2 Next

Figure 1: Example application, where *Page 1* contains the diversified, exploratory relevance ranking, and *Page 2* contains a refined, personalised re-ranking of the next set of remaining documents, triggered by the "Next" button and depending on the documents viewed on *Page 1* (the left *Page 2* ranking contains documents about animals, the right ranking documents about cars). The same search process can continue in the remaining results pages until the user leaves.

Relevance Feedback can be generalised into the field of Interactive Retrieval, whereby a user interacts with a search system throughout a search session, and the system responds to the user's actions by updating search results and improving the search experience. Interaction can be split into two categories, *system driven* and *user driven*. In system driven interaction, the system directly involves the user in the interactive process; for instance, a user can narrow down an e-commerce search by filtering based on item attributes the system has intentionally specified [44]. Somewhat related is the field of adaptive filtering, where a user refines an information stream over time by giving feedback [24].

User driven interactions can allow the user to opt-in, or remain unaware of the interaction due to the use of implicit feedback, which can come from a number of sources, including scrolling behaviour, time spent on search page and clickthroughs, the latter two of which correspond well with explicit user satisfaction [13]. Due to their abundance and the low cost of recording them, clickthroughs have emerged as a common measure of implicit feedback, with much research into using search click logs for learning ranking algorithms [16].

When implicit feedback is used in interactive retrieval, the research can be split into *long term* and *short term* objectives. Long term interactive retrieval methods are usually concerned with search personalisation or learning to rank over time, for instance, incorporating user behavior from search logs into a learning to rank algorithm [1], re-ranking search results for specific users by accumulating user data over time [35] and using clickthroughs to learn diverse rankings dynamically over time [21]. Such techniques are useful for improving a search system over time for popular queries, but are less able to deal with ad hoc web search and tail queries. On the other hand, short term methods aim to improve the immediate search session by incorporating user context such as clickthroughs and search history [29], but may have issues with user privacy concerning using user data.

In this paper, we propose a technical schema for short term, interactive ranking that is able to use user feedback from the top-ranked documents in order to generate contextual re-rankings for the remaining documents in ad hoc retrieval. The feedback could be collected either implicitly, such as from clickthroughs and search page navigation, or explicitly, such as from user ratings. The proposed feedback scheme naturally makes use of the fact that in many common retrieval scenarios, the search results are split into multiple pages that the user traverses across by clicking a "next page" button. A typical use case would be a user examining a Search Engine Results Page (SERP) by going through the list, clicking on documents (e.g., to explore them) and returning to the SERP in the same session. Then, when the user indicates that they'd like to view more results (i.e. the 'Next Page' button), the feedback elicited thus far is used in a Bayesian model update to unobtrusively re-rank the remaining documents (which are not seen as yet), on the client-side, into a list that is more likely to satisfy the information need of the user, which are shown in the next SERP, as illustrated in Figure 1.

We mathematically formulate the problem by considering a Bayesian sequential model; we specify the user's overall satisfaction over the Interactive Exploratory Search process and optimise it as a whole. The expected relevance of documents is sequentially updated taking into account the user feedback thus far and the expected document dependency. A dynamic programming approach is used to optimise the balance between exploration (learning the remaining documents' relevancy) and exploitation (presenting the most relevant documents thus far). Due to the optimisation calculation being intractable, the solution is approximated using Monte Carlo sampling and a sequential ranking decision rule. Our formulation requires no changes to the search User Interface (UI) and the update stage can be performed on the client side in order to respect user privacy and deal with computational efficiency. We test the method on TREC datasets, and the results show that the method outperforms other strong baselines, indicating that the proposed interactive scheme has significant potentials for exploratory search tasks.

In Section 2 we continue our discussion about the related work; in Section 3 we present the proposed dynamical model. It's insights and it's approximate solutions are presented in Sections 4 and 5 respectively. The experiments are given in Section 6 and conclusions are presented in Section 7.

2. RELATED WORK

A similar approach was explored by Shen [30], where an interactive re-ranking occurred when a user returned to the SERP after clicking on a document in a single session. Aside from the UI differences, they also did not perform active learning (exploration) on the initial ranking, and as such, did not take into account issues such as document dependence and diversity, and they immediately make use of implicit feedback rather than first collecting different examples of it. An alternative approach displayed dynamic drop-down menus containing re-rankings to users who clicked on individual search results [6], allowing them to navigate SERPs in a tree like structure (or constrained to two levels [23]). Again, this method only takes into account feedback on one document at a time, arguably, it could be considered intrusive to the user in some cases and it does not protect user privacy by working on the client side, but it does also perform active learning in the initial ranking in order to diversify results.

A common feature in the above-mentioned literature is the use of alternative UIs for SERPs. Over the years many such interfaces have been researched, such as grouping techniques [43] and 3D visualisation [14], often with mixed user success. Likewise, attempting to engage the user in directly interacting with the search interface (such as drop down menus) or altering their expectations (by changing the results ranking) may prove similarly detrimental. In early relevance feedback research, it was found that interactively manipulating the search results was less satisfactory to users than offering them alternative query suggestions [17], and likewise with automatic query expansion [28]. Users appeared to be less happy with systems that automatically adjusted rankings for them and more satisfied when given control over how and when the system performed relevance feedback [39]. Our technique requires no change to the existing, familiar UI and is completely invisible to the user, but it is still responsive to their actions and only occurs when a user requests more information; for the user who is interested in only the first page or is precision-minded, no significant changes occur.

A similar area of work is in the field of active learning, although there remain some fundamental differences. Whilst Shen [31] and Xu [41] researched methods that could be used to generate inquisitive, diverse rankings in an iterative way in order to learn better rankings, their aims were in long term, server-side learning over time that also lacked the ability to adjust the degree of learning for different types of query or user. [15, 32] also investigated the balance between exploration and exploitation when learning search rankings in an online fashion using implicit feedback, although their goal was to optimise rankings over a period of time for a population of users.

We also consider the fact that the same search techniques are not employed uniformly by users across queries; for instance users exhibit different behaviour performing *navigational* queries than they do *informational* queries [3]. It is not only non-intrusive to use a Next Page button to trigger the personalised re-rank, but most importantly, the user deciding to access the next page sends a strong signal that there might be more relevant documents for the query and that the user is interested in exploring them. In addition, our formulation contains a tunable parameter that can adjust the level of active learning occurring in the initial stage, including none at all. This parameter is a reflection of how the user is likely to interact with the SERP, with different settings for those queries where users will follow the exploratory search pattern [18]. That our technique is able to handle search tasks of varying complexity using implicit feedback, and occurs during the middle, exploratory stage of a search session (when the user is changing SERPs) makes it an ideal candidate for implicit relevance feedback [40].

We see parallels with this work and other areas of online learning. Our model and sequential search problem can be considered as a Partially Observable Markov Decision Process (POMDP) [9], where the unknown state of the POMDP is the probability of relevance for each document, the belief state is given by the multivariate Gaussian distribution, the observation is the user feedback, the reward is the IR metric being measured and curiously there is no transition function as we assume that the document relevancy doesn't change during interaction (however, the belief states of relevance keep updating). Considering the problem in this manner could foster further research into approximative solution techniques or hint at a more general formulation. In addition, the exploration and exploitation of the documents in our formulation (and other active learning methods) bears similarities with multi-armed bandit (MAB) theory [20, 21].

3. PROBLEM FORMULATION

Our formulation concerns the exploratory, multi page interactive search problem; where a user performs an initial search which returns a set of ranked results that take into account the covariance of the documents, allowing for diversification and exploration of the set of documents so as to better learn a ranking for the next pages. The user then engages in usual search behaviour such as rating or checking the snippets and clicking on documents and returning to the SERP to view the remaining ranked documents. Upon clicking on the next page button to indicate that they would like to view more results, the feedback obtained so far is combined with the document covariance to re-rank the search results on the next pages. The Interactive Exploratory Search process continues until the user leaves the system.

In traditional IR ranking, we would order documents according to some IR score or probability of relevance in decreasing order, calculated using some prior knowledge such as the match between query and document features [25]. This score is an estimate of the underlying relevance score, which is fixed but unknown. For our model, we assume that this true relevance is Normally distributed around the IR score estimate, which is equivalent to assuming that the estimate is correct subject to some Gaussian noise. This is defined for all documents using the multivariate Normal distribution

$$\boldsymbol{R}^t \equiv [R_1^t, R_2^t, \ldots, R_N^t] \backsim \mathcal{N}(\boldsymbol{\theta}^t, \boldsymbol{\Sigma}^t) \tag{1}$$

where N is the number of documents, R_i^t the random variable for the relevance score of document i on page t, $\boldsymbol{\theta}^t \equiv [\theta_1^t, \ldots, \theta_N^t]$ the estimates of the relevance score for the documents and $\boldsymbol{\Sigma}^t$ the covariance matrix.

Because we initially are not given any feedback information, $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ can be calculated using an existing IR model, for example, $\boldsymbol{\theta}$ could be derived from the BM25 [34] or Vector Space Model score and $\boldsymbol{\Sigma}$ could be approximated using a document similarity metric, or by topic clustering [36, 38].

Once the user feedback has been retrieved after the first page, we can then update our model as a conditional distribution.

For a given query, the search system displays M documents in each result page and ranks them according to an objective function. This rank action is denoted as vector $\boldsymbol{s}^t = [s_1^t, \ldots, s_M^t]$ where s_j^t is the index of the document retrieved at rank j at the tth page i.e., $\boldsymbol{s}^1$ is the page 1 (initial) ranking and $\boldsymbol{s}^2$ is the page 2 (modified) ranking. We let $\boldsymbol{s}$ represent all rank actions $\boldsymbol{s}^1 \ldots \boldsymbol{s}^T$. We denote $\boldsymbol{r}^{\boldsymbol{t}} = [r_1^t, \ldots, r_K^t]$ as the vector of feedback information obtained from the user whilst using page t, where K is the number of documents given feedback with $0 \leq K \leq M$, and r_i^t is the relevance feedback for document i, either by measuring a direct rating from the user or by observing click-throughs.

When considering the objective from page t until page T, we use a weighted sum of the expected DCG@M scores of the rankings of the remaining result pages, denoted here by (note that $R_{s_j}^t \equiv R_{s_j^t}^t$)

$$U_{\boldsymbol{s}}^t = \sum_t^T \left(\lambda^t \sum_{j=1+(t-1)M}^{tM} \frac{E(R_{s_j}^t)}{\log_2(j+1)} \right) \tag{2}$$

where $U_{\boldsymbol{s}}^t$ represents the user's overall satisfaction for rank action $\boldsymbol{s}$, where $E(R_{s_j}^t) = \theta_{s_j}^t$ is the expected relevance of a document at rank j in result page t. We have chosen the objective function as it is simple and both rewards finding the most relevant documents and also ranking them in the correct order, although other IR metrics can be adopted similarly. The rank weight $\frac{1}{\log_2(j+1)}$ is used to give greater weight to ranking the most relevant documents in higher positions. The tunable parameter $\lambda^t \geq 0$ is used to adjust the importance of result pages and thus the level of exploration in the initial page(s). When λ^1 is assigned to be zero, the initial ranking is chosen so as to maximise the next page(s) ranking and thus the priority is purely on exploration [41]. It could also be a decay function reflecting the fact that less users are likely to reach the later result pages. An extreme case is when $\lambda^t = 1 \; \forall t$, meaning there is no exploration at all and the documents are ranked according to the PRP. In this paper, we fix λ^t a priori, while leaving the study of adaptive λ^t according to learned information intents [3, 22] for future work.

T indicates how far ahead we consider when ranking documents in the current page. Our experiments using TREC data indicate that setting $T = 2$ (only considering the immediate next page) gives the best results. To simplify our derivations, we consider $T = 2$ for the remaining derivations while the resulting algorithms can be easily generalised to the case where $T > 2$. Thus, our overall objective is to find the optimal rank action $\boldsymbol{s}^*$ such that U is maximised

$$\begin{aligned}\boldsymbol{s}^* = \operatorname*{argmax}_{\boldsymbol{s}} \Bigg\{ \lambda \sum_{j=1}^{M} \frac{\theta_{s_j}^1}{\log_2(j+1)} \\ + (1-\lambda) \sum_{j=1+M}^{2M} \frac{\theta_{s_j}^2}{\log_2(j+1)} \Bigg\}\end{aligned} \tag{3}$$

where we set λ to balance the importance of pages 1 and 2 when $T = 2$. This statistical process can be represented by an Influence Diagram as shown in Figure 2.

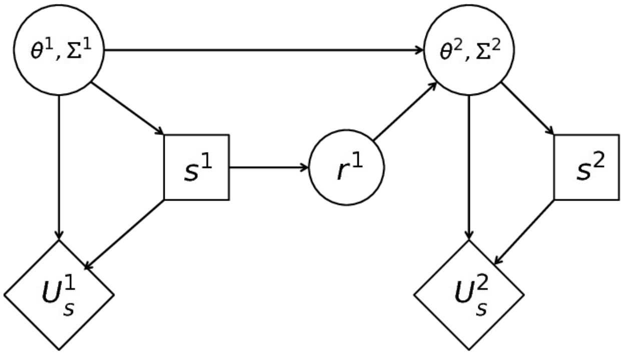


Figure 2: The interactive search process as illustrated by an influence diagram over two stages. In the diagram, the circular nodes represent random variables; the square nodes represent the rank action at each result page. The rhombus node is the instant utility.

3.1 Relevance Belief Update

We start by considering the problem in reverse; given that we've already displayed a ranking to the user, how can we use their feedback to update our ranking in the second result page. After receiving some observation of feedback $\boldsymbol{r}$ on page 1, and with the user requesting more documents, we can set the conditional distribution of the probability of relevance of the remaining documents given the feedback as

$$\boldsymbol{R}^2 = P(\boldsymbol{R}_{\backslash \boldsymbol{s}'} | \boldsymbol{R}_{\boldsymbol{s}'} = \boldsymbol{r}) \tag{4}$$

where we have reordered the vectors $\boldsymbol{R}^1$ and $\boldsymbol{\theta}^1$ as

$$\boldsymbol{R}^1 = \begin{pmatrix} \boldsymbol{R}_{\backslash \boldsymbol{s}'} \\ \boldsymbol{R}_{\boldsymbol{s}'} \end{pmatrix} \quad \boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_{\backslash \boldsymbol{s}'} \\ \boldsymbol{\theta}_{\boldsymbol{s}'} \end{pmatrix} \tag{5}$$

where $\boldsymbol{s}'$ is the vector of documents in rank action $\boldsymbol{s}^1$ that received feedback in $\boldsymbol{r}$, and $\backslash \boldsymbol{s}'$ the remaining $N - K$ documents. We also reorder $\boldsymbol{\Sigma}^1$ to give

$$\boldsymbol{\Sigma}^1 = \begin{pmatrix} \boldsymbol{\Sigma}_{\backslash \boldsymbol{s}'} & \boldsymbol{\Sigma}_{\backslash \boldsymbol{s}' \boldsymbol{s}'} \\ \boldsymbol{\Sigma}_{\boldsymbol{s}' \backslash \boldsymbol{s}'} & \boldsymbol{\Sigma}_{\boldsymbol{s}'} \end{pmatrix} \tag{6}$$

where $\boldsymbol{\Sigma}_{\backslash \boldsymbol{s}'}$ are the variances and covariances of the set $\boldsymbol{s}'$, and $\boldsymbol{\Sigma}_{\backslash \boldsymbol{s}' \boldsymbol{s}'}$ the covariances between $\backslash \boldsymbol{s}'$ and $\boldsymbol{s}'$. Then, we can specify the conditional distribution as

$$\boldsymbol{R}^2 \sim \mathcal{N}(\boldsymbol{\theta}^2, \boldsymbol{\Sigma}^2) \tag{7}$$

$$\boldsymbol{\theta}^2 = \boldsymbol{\theta}_{\backslash \boldsymbol{s}'} + \boldsymbol{\Sigma}_{\backslash \boldsymbol{s}' \boldsymbol{s}'} \boldsymbol{\Sigma}_{\boldsymbol{s}'}^{-1} (\boldsymbol{r} - \boldsymbol{\theta}_{\boldsymbol{s}'}) \tag{8}$$

$$\boldsymbol{\Sigma}^2 = \boldsymbol{\Sigma}_{\backslash \boldsymbol{s}'} - \boldsymbol{\Sigma}_{\backslash \boldsymbol{s}' \boldsymbol{s}'} \boldsymbol{\Sigma}_{\boldsymbol{s}'}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{s}' \backslash \boldsymbol{s}'} \tag{9}$$

These formulas can then be used to generate new, posterior relevance score estimates for the documents, on the client-side, based on the feedback received in result page 1. These score updates can be used to re-rank the documents in result page 2 and display them to the user.

3.2 User Feedback

Before presenting a ranked list at result page 1 and directly observing $\boldsymbol{r}$, it remains an unknown variable and can be predicted using our prior knowledge. In order to perform exploration (active learning), we will need to be able to predict how a user will respond to each potential ranking. Using the model above, we can predict the user's judgements $\boldsymbol{r}$ based on a particular rank action $\boldsymbol{s^1}$. There are generally two simple models we can use when interpreting user feedback: 1) The user gives feedback on all of the top M documents i.e. $K = M$ 2) The number of judgements is variable depending on how many documents the user looks through. The top ranked documents are the most likely to gain feedback [12].

For the first case, $\boldsymbol{r}$ follows the multivariate normal distribution,

$$\boldsymbol{r} \backsim \mathcal{N}(\boldsymbol{\theta_s^1}, \boldsymbol{\Sigma_s^1}) \tag{10}$$

where $\boldsymbol{\theta_s^1} = [\theta^1_{s_1}, \ldots, \theta^1_{s_M}]$. For the second case, the probability of observing a particular feedback vector $\boldsymbol{r}$ depends on the rank bias weight $1/(log_2(j+1))$, which can be easily derived as:

$$P(\boldsymbol{r}) = \frac{1}{(2\pi)^{K/2}|\boldsymbol{\Sigma_{s'}}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\theta_{s'}})^T \boldsymbol{\Sigma_{s'}}^{-1}(\boldsymbol{r}-\boldsymbol{\theta_{s'}})\right\} \times \prod_{j=1}^{M} \frac{(\log_2(j+1)-1)^{1-\epsilon_j}}{\log_2(j+1)} \tag{11}$$

where $\epsilon_j = 1$ if the document at rank j has received a user judgement, otherwise 0. $\boldsymbol{s'}$ refers to the documents in rank action $\boldsymbol{s}^1$ that have feedback in $\boldsymbol{r}$. In this paper, we consider case 2.

4. RETRIEVAL OPTIMISATION

We are now ready to discuss how to maximise the objective function using the Bellman Equation [4] and the feedback distribution in Eq. (11), and we present two toy examples to demonstrate how the model works and its properties.

4.1 Bellman Equation

Let $V(\boldsymbol{\theta}^t, \boldsymbol{\Sigma}^t, t) = \max_{\boldsymbol{s}} U^t_{\boldsymbol{s}}$ be the value function for maximising the user satisfaction of the ranking up until step T. Setting $t = 1$, we can derive the Bellman Equation [4] for this problem, giving us

$$V(\boldsymbol{\theta}^1, \boldsymbol{\Sigma}^1, 1) = \max_{\boldsymbol{s}^1}\left\{\sum_{t=1}^{T} \lambda^t \sum_{j=1+(t-1)M}^{tM} \frac{E(\boldsymbol{R}^t_{\boldsymbol{s}})}{\log_2(j+1)}\right\} \tag{12}$$

$$= \max_{\boldsymbol{s}^1}\left\{\lambda^1 \boldsymbol{\theta}^1_{\boldsymbol{s}} \cdot \boldsymbol{W}_1 + \max_{\boldsymbol{s}^2} E\left(\sum_{t=2}^{T} \lambda^t \boldsymbol{\theta}^t_{\boldsymbol{s}} \cdot \boldsymbol{W}_t \middle| \boldsymbol{r}\right)\right\} \tag{13}$$

$$= \max_{\boldsymbol{s}^1}\left\{\lambda^1 \boldsymbol{\theta}^1_{\boldsymbol{s}} \cdot \boldsymbol{W}_1 + E\left(V(\boldsymbol{\theta}^2, \boldsymbol{\Sigma}^2, 2)\middle| \boldsymbol{r}\right)\right\} \tag{14}$$

where $\boldsymbol{W}_t = [\frac{1}{\log_2(1+(t-1)M)}, \ldots, \frac{1}{\log_2(M+(t-1)M)}]$ is the DCG weight vector. Eq. (14) is the general solution to this problem, by setting $T = 2$ and λ accordingly we can find the solution to our scenario

$$= \max_{\boldsymbol{s}^1}\left\{\lambda \boldsymbol{\theta}^1_{\boldsymbol{s}} \cdot \boldsymbol{W}_1 + \max_{\boldsymbol{s}^2}(1-\lambda)\int_{\boldsymbol{r}} \boldsymbol{\theta}^2_{\boldsymbol{s}} \cdot \boldsymbol{W}_2 P(\boldsymbol{r})\mathrm{d}\boldsymbol{r}\right\} \tag{15}$$

4.2 Examples and Insights

We illustrate the properties of the resulting document ranking using some toy examples. To simplify matters, we observe that if there are only two documents ($N = 2$) and we only need to display one of them at a time ($M = 1$), this allows us to assume that for some h

$$\theta^2_1 > \theta^2_2 \Leftrightarrow r_s > h$$

This indicates that when the observation r_s is bigger than some h, the updated expected relevance score of document 1 will be larger than document 2, and vice versa. This allows us to integrate over regions of possible values for r_s given the optimal ranking decision for s_2, simplifying Eq. (15) into two integrals with exact solution

$$V(\boldsymbol{\theta}^1, \boldsymbol{\Sigma}^1, 1) = \max_{s^1}\left\{\lambda\theta^1_s W_1 + (1-\lambda)W_2\left(\int_{-\infty}^{h} \theta^2_2 P(r_s)\mathrm{d}r_s + \int_{h}^{\infty} \theta^2_1 P(r_s)\mathrm{d}r_s\right)\right\} \tag{16}$$

4.2.1 *Different Variance*

In this example we wish to illustrate how our model is able to use a document's score variance to explore rankings during the first result page; in this case we suppose that there are only two documents and that the search system must choose one at each result page. Our prior knowledge about the score and its covariance matrix is

$$\boldsymbol{\theta}^1 = \begin{pmatrix} 1 \\ 0.9 \end{pmatrix}, \boldsymbol{\Sigma}^1 = \begin{pmatrix} 0.6^2 & 0.2 \\ 0.2 & 0.1^2 \end{pmatrix}$$

i.e. the documents are similarly scored (and have identical covariance), but have different variances.

We first consider ranking document 1 first ($s^1 = 1$) and observe feedback r_1, allowing us to update our expected relevance scores

$$\begin{aligned}
\theta^2_1 &= r_1 \\
\theta^2_2 &= \theta^1_2 + \frac{\sigma_{1,2}}{(\sigma_1)^2}(r_1 - \theta^1_1) \\
&= 0.9 + \frac{0.2}{0.6^2}(r_1 - 1) = 0.344 + 0.556 r_1 \\
\theta^2_1 &> \theta^2_2 \Leftrightarrow r_1 > h = 0.775
\end{aligned}$$

We nominally set $\lambda = 0.5$, then using Eq. (16) we can calculate the value of ranking document 1 first as

$$\begin{aligned}
=&\lambda\theta^1_1 W_1 + (1-\lambda)W_2\left(\int_{-\infty}^{h} \theta^2_2 P(r_1)\mathrm{d}r_1 + \int_{h}^{\infty} \theta^2_1 P(r_1)\mathrm{d}r_1\right) \\
=&0.5 + 0.315\left(\int_{-\infty}^{0.775} 0.344 P(r_1)\mathrm{d}r_1 + 0.556\theta^1_1 + (1-0.556)\int_{0.775}^{\infty} P(r_1) r_1 \mathrm{d}r_1\right) \\
=&0.5 + 0.315\Big(0.344 P(r_1 < 0.775) + 0.556 + 0.444(\theta^1_1 - \sigma^2_1)(1 - P(r_1 < 0.775))\Big) \\
=&0.77169
\end{aligned}$$

Alternatively, if we rank document 2 first ($s^1 = 2$), we have

$$\begin{aligned}
\theta^2_1 &= \theta^1_1 + \frac{\sigma_{1,2}}{(\sigma_2)^2}(r_2 - \theta^1_2) \\
\theta^2_2 &= r_2 \\
\theta^2_1 &< \theta^2_2 \Leftrightarrow r_2 < h = 0.8947
\end{aligned}$$

Giving us a value of 0.78026. Thus, we have greater value when we rank document 2 at result page 1. This is different from the *myopic* policy or the PRP which would simply rank the document with the highest score first, in this case document 1. Here, because the variance of the score of document 2 is lower, it is optimal to display it first, a side effect of which is that we may observe feedback from a user which subsequently reduces its variance and the variance of the other documents, giving more accurate rankings in a later step.

4.2.2 *Different Correlation*

Now, suppose that there are three documents and the search system must choose two documents at each result page. We assume that documents have a graded relevance from 1 to 5 and that the user will give explicit feedback on the top two documents when receiving a ranked list at result page 1. In this example, we wish to demonstrate how the covariance of the documents can affect our optimal ranking, so we fix the variance for all three documents to 1 and set our prior knowledge about their mean and covariance to

$$\boldsymbol{\theta}^1 = \begin{pmatrix} 2.99 \\ 3 \\ 5 \end{pmatrix}, \boldsymbol{\Sigma}^1 = \begin{pmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0.95 \\ 0 & 0.95 & 1 \end{pmatrix}$$

To simplify the problem, we also ignore the weights $\boldsymbol{W}$ and λ. For the first result page, the myopic ranking would be to display documents 3 and 2 first because they have the highest prior scores i.e. $s^1_1 = 3, s^1_2 = 2$. After observing feedback r_3 and r_2, we can update our expected relevance scores

$$\begin{aligned}\theta^2_1 &= \theta^1_1 + \boldsymbol{\Sigma}_{1,\{2,3\}} \boldsymbol{\Sigma}^{-1}_{\{2,3\}} (\boldsymbol{r}_{\{2,3\}} - \boldsymbol{\theta}_{\{2,3\}}) \\ &= \theta^1_1 + \begin{pmatrix} 0.1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix}^{-1} \left[\begin{pmatrix} r_2 \\ r_3 \end{pmatrix} - \begin{pmatrix} \theta^1_2 \\ \theta^1_3 \end{pmatrix} \right] \\ &= 2.99 + 1.026(r_2 - 3) - 0.974(r_3 - 5) \\ \theta^2_2 &= r_2 \\ \theta^2_3 &= r_3\end{aligned}$$

Thus, the value for this particular ranking becomes

$$\begin{aligned} &= \theta^1_3 + \theta^1_2 + \int_A P(r_2, r_3) \cdot (r_2 + r_3) \mathrm{d}r_2 \mathrm{d}r_3 \\ &\quad + \int_B P(r_2, r_3) \cdot (\theta^2_1 + r_3) \mathrm{d}r_2 \mathrm{d}r_3 \\ &\quad + \int_C P(r_2, r_3) \cdot (\theta^2_1 + r_2) \mathrm{d}r_2 \mathrm{d}r_3 = 16.380\end{aligned}$$

where

$$\begin{aligned} A &= \{r_2, r_3 | \theta^2_1 < \min(r_2, r_3)\} \\ B &= \{r_2, r_3 | r_2 < \min(\theta^2_1, r_3)\} \\ C &= \{r_2, r_3 | r_3 < \min(\theta^2_1, r_2)\}\end{aligned}$$

Alternatively, if we choose the ranking $s^1_1 = 3, s^1_2 = 1$, then we can calculate the value of the ranking as 16.528.

We see that the myopic policy (e.g. the PRP) is once again not the optimal policy, in this case it is more optimal to rank $s^1_1 = 3, s^1_2 = 1$ at result page 1. This is because documents 2 and 3 are highly correlated, unlike documents 1 and 3, and so we can learn more information from ranking documents 1 and 3 together. The toy example reveals that an optimal policy tends to rank higher the documents that are either negatively or positively correlated with the remaining documents. Encouraging the negative correlation would guarantee a diversified rank list [37] whilst the positive correlation could result in the documents in the center of a document cluster being driven to the higher ranks.

5. SOLUTION APPROXIMATION

In the previous section we outlined and demonstrated the formulae that can be used to calculate the optimal solution to the interactive, multi page exploratory search problem, making use of a Bayesian update and the Bellman equation. Unfortunately, as is often the case with dynamic programming solutions the optimal solution does not reasonably scale, and for more than three documents, the integral in Eq. (15) cannot be directly solved.

5.1 Monte Carlo Sampling

Instead, we can find an approximate solution by making use of Monte Carlo sampling. The optimal value function $V(\boldsymbol{\theta}^1, \boldsymbol{\Sigma}^1, 1)$ can be approximated by

$$\approx \max_{\boldsymbol{s}^1} \left\{ \lambda \boldsymbol{\theta}^1_{\boldsymbol{s}} \cdot \boldsymbol{W}_1 + \max_{\boldsymbol{s}^2} (1 - \lambda) \frac{1}{S} \sum_{\hat{\boldsymbol{r}} \in O} \boldsymbol{\theta}^2_{\boldsymbol{s}} \cdot \boldsymbol{W}_2 P(\hat{\boldsymbol{r}}) \right\} \tag{17}$$

where O is the sample set of possible feedback vectors $\boldsymbol{r}$ and S is the sample size. We can use Eq. (17) to determine an estimate of the optimal ranking decision for the first page at $t = 1$.

5.2 Sequential Ranking Decision

Eq. (17) does not completely solve the problem of intractability as there are still an insurmountable number of possible ranking actions possible, even if we restrict our ranking to only select documents at the top M ranking positions. To counter this, we use a sequential ranking decision that lowers the computational complexity of generating rankings. We first consider selecting only one document to display at rank position 1, which has only N possible actions. We use Eq. (17) to find the optimal rank action, then repeat the procedure for the remaining ranking positions and documents, sequentially selecting a document from rank position 1 to position M. Although this method does not give a global optimal solution, it does provide an excellent balance between accuracy and efficiency.

Algorithm 1 illustrates our approximated approach. Here, parameter K is assigned a priori, which depends on our assumptions about the behavior of user judgements. If we believe a user only provides judgements on the top M documents, it is reasonable to assign $M = K$, as only the selection for the top M rank positions will influence the updated model at result page 2. When we find the optimal ranking from ranks 1 to K, the rest of the documents in page 1 can be ranked following the traditional IR scores. Because $T = 2$, the maximum utility for the second page is obtained when we rank in decreasing order of $\boldsymbol{\theta}^2_{\setminus \boldsymbol{s}}$ i.e. the updated probabilities of relevance for the remaining documents given $\hat{\boldsymbol{r}}$.

6. EXPERIMENT

We consider the scenario where a user is presented with a list of M documents on the first page, where we set $M = 10$

Algorithm 1 The IES (Interactive Exploratory Search) algorithm

```
function IES($\boldsymbol{\theta}, \boldsymbol{\Sigma}, K$)
    $\boldsymbol{s}$ = ARRAY($M$)
    loop $j \leftarrow 1$ to $K$                         ▷ For each rank position
        value = ARRAY($N$)
        loop $i \leftarrow 1$ to $N$                     ▷ For each document
            if $i \subseteq \boldsymbol{s}$ then continue
            end if                  ▷ Ignore if document already ranked
            $\boldsymbol{s}(j) = i$
            $\boldsymbol{O}$ = ARRAY($S$)
            $\boldsymbol{O}$ = CREATESAMPLE($\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{s}$)
            sampleAvg = 0
            for all $\hat{\boldsymbol{r}} \in \boldsymbol{O}$ do            ▷ For each sample
                $\boldsymbol{\theta}^2_{\backslash \boldsymbol{s}} = \boldsymbol{\theta}^1_{\backslash \boldsymbol{s}} + \boldsymbol{\Sigma}_{\backslash \boldsymbol{s}\boldsymbol{s}} \boldsymbol{\Sigma}_{\boldsymbol{s}}^{-1} (\hat{\boldsymbol{r}} - \boldsymbol{\theta}^1_{\boldsymbol{s}})$
                $\boldsymbol{s}_2$ = SORT($\boldsymbol{\theta}^2_{\backslash \boldsymbol{s}}$, descend)$[1 \rightarrow M]$
                sampleAvg$+= \boldsymbol{\theta}^2_{\boldsymbol{s}_2} \cdot \boldsymbol{W}_2 \times P(\hat{\boldsymbol{r}})$
            end for
            sampleAvg$\backslash = S$              ▷ Average sample value
            value($i$) $= \lambda \boldsymbol{\theta}^1_{\boldsymbol{s}} \cdot \boldsymbol{W}_1 + (1 - \lambda) \times$ sampleAvg
        end loop
        $\boldsymbol{s}(j)$ = INDEX(MAX(value))
    end loop
    $\boldsymbol{s}(K+1 \rightarrow M)$ = SORT($\boldsymbol{\theta}^1_{\backslash \boldsymbol{s}}$, descend)$[1 \rightarrow M - K + 1]$
    return $\boldsymbol{s}$
end function
```

as it provided reasonable and meaningful results within the scope of our test data set and is reflective of actual search systems. The user then provides explicit or implicit judgements and then clicks the 'Next Page' button, whereby they are presented with a second (and possibly third) page of M documents on which they also provide their judgements. We test our Interactive Exploratory Search (IES) technique which uses dynamic programming to select a ranking for the first page, then using the judgements from the first page generates a ranking for the remaining pages using the conditional model update from Section 3.1.

We compare the IES algorithm with a number of methods representing the different facets of our technique: a baseline which is also the underlying primary ranking function, the BM25 ranking algorithm[34]; a variant of the BM25 that uses our conditional model update for the second page ranking which we denote BM25-U; the Rocchio algorithm[26], which uses the baseline ranking for the first page and performs relevance feedback to generate a new ranking for the second page; and the Maximal Marginal Relevance (MMR) method[8] and variant MMR-U which diversifies the first page using the baseline ranking and our covariance matrix, and ranks the second page according to the baseline or the conditional model update respectively.

We tested our algorithm on three TREC collections; we used the WT10G collection because it uses graded relevance judgements which are suited to the multivariate normal distribution we use, allowing us to test the explicit feedback scenario. To test our approach on binary judgements representing implicit clickthroughs, we used the TREC8 data set. Additionally, we used the 2004 Robust track which focused on poorly performing topics, which allowed us to test our algorithm on difficult queries which are likely to require recall-oriented, exploratory search behaviour.

The documents were ranked according to BM25 scores for each topic, and the top 200 used for further re-ranking using the IES algorithm and baselines. Each dataset contained 50 topics, and relevance judgements for those topics were used to evaluate the performance of each algorithm. The judgements were also used to simulate the feedback of users after viewing the first page, allowing for the conditional model update to generate a ranking for the second page.

We used four metrics to measure performance; precision@M, recall@M, nDCG@M and MRR, and measured the metrics at both M (for first page performance) and $2M$ (for overall performance over the 1st and 2nd page). Precision, recall and nDCG are commonly used IR metrics that allow us to compare the effectiveness of different rankings. Because our investigation is focused on the exploratory behaviour of our approach, we did not explicitly use any sub-topic related diversity measure in our experiments. However, we used the MRR as a risk-averse measure, where a diverse ranking should typically yield higher scores [37, 10].

6.1 Mean Vector and Covariance Matrix

We applied the widely used BM25 [34] model to define the prior mean score vector $\boldsymbol{\theta}^1$ for result page 1. Eq. 18 below was used to scale the BM25 scores into the range of relevance judgements required for our framework

$$\theta = \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \cdot b \tag{18}$$

where $\boldsymbol{x}$ is the vector of BM25 scores and b is the range of score values i.e. $b = 1$ for binary clickthrough feedback, $b > 1$ when explicit ratings are given by the user.

The covariance matrix $\boldsymbol{\Sigma}^1$ represents the uncertainty and correlation associated with the estimations of the document scores. It is reasonable to assume that the covariance between two relevant scores can be approximated by the document similarities. We investigated two popular similarity measures, Jaccard Similarity and Cosine Similarity, and our experiments showed that the latter had a much better performance and is used in the remainder of our experiments. The variance of each document's relevance score is set to be a constant in this experiment as we wish to demonstrate the effect of document dependence on search results, and it is more difficult to model score variance than covariance.

6.2 The Effect of λ on Performance

In this experiment, we measured the recall@M, precision@M, nDCG@M and MRR for $M = 10$ and $M = 20$ i.e. the scores for the first page, and for the first and second page combined respectively. We wished to explore how the parameter λ affected the performance of the IES algorithm, in particular, the average gains we could expect to achieve. To calculate the overall average gain, we calculated the average loss $(\text{metric(IES)} - \text{metric(baseline)})/M$ at $M = 10$ and $M = 20$, and added the two values. This value tells us the difference in a metric on average we'd expect to gain/lose by using our technique over multiple pages.

In Figure 3, we see how the average gain varies for different values of λ across the different data sets against the BM25 baseline. We note that the setting of λ has a definite effect on performance; we observe that for smaller values ($\lambda < 0.5$), where the algorithm places more emphasis on displaying an exploratory, diverse ranking on the first page, we see a marked drop in performance across all metrics. As

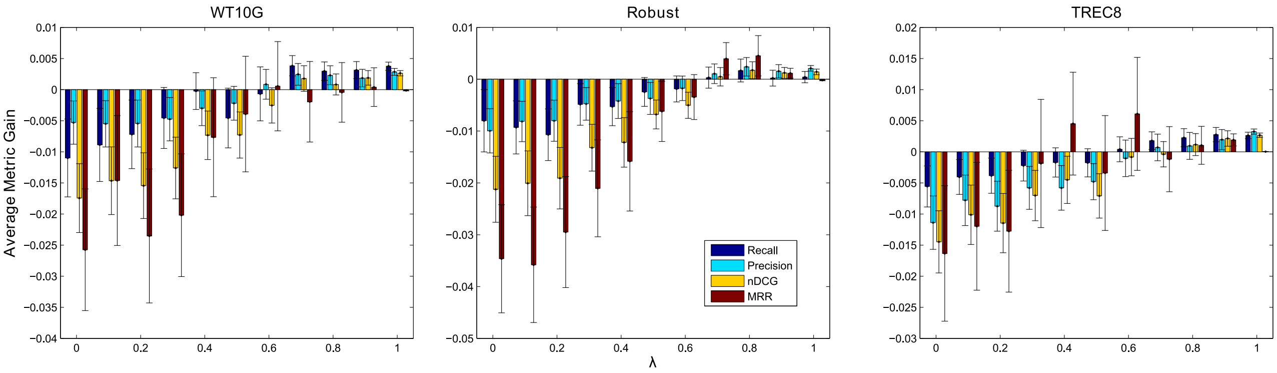


Figure 3: The average metric gain of the IES algorithm over the BM25 baseline when $M = 10$ with 95% confidence intervals. Each bar represents the average gain in a particular metric for a given value of λ, and each chart gives results for a different TREC data set. Positive values indicate gains made over the baseline algorithm, and negative values losses.

			Recall@		Precision@		nDCG@		MRR@	
			10	20	10	20	10	20	10	20
WT10G	#	IES	0.2287	0.4540^{13}	0.3560	0.3940^{135}	0.3823	0.4755^{135}	0.5900	0.5912
	1	BM25	0.2334	0.3651	0.3680	0.3210	0.3935	0.4171	0.6025	0.6053
	2	BM25-U	0.2334	0.4438	0.3680	0.3810	0.3935	0.4715	0.6025	0.6025
	3	MMR	0.1937	0.3651	0.3260	0.3210	0.3424	0.4171	0.5819	0.6053
	4	MMR-U	0.1937	0.4102	0.3260	0.3780	0.3424	0.4407	0.5819	0.5832
	5	Rocchio	0.2334	0.3999	0.3680	0.3320	0.3935	0.4392	0.6025	0.6053
Robust	#	IES	0.2308	0.3898^{13}	0.3851	0.3968^{123}	0.4594	0.5612^{135}	0.6652	0.6670
	1	BM25	0.2276	0.3607	0.3915	0.3330	0.4636	0.4706	0.6317	0.6377
	2	BM25-U	0.2276	0.3694	0.3915	0.3777	0.4636	0.5008	0.6317	0.6335
	3	MMR	0.2179	0.3607	0.3766	0.3330	0.4429	0.4706	0.6246	0.6377
	4	MMR-U	0.2179	0.3785	0.3766	0.3883	0.4429	0.5010	0.6246	0.6264
	5	Rocchio	0.2276	0.3582	0.3915	0.3638	0.4636	0.4725	0.6317	0.6377
TREC8	#	IES	0.1803	0.3768	0.4388	0.4704	0.4644	0.5228	0.6533	0.6583
	1	BM25	0.1851	0.3106	0.4551	0.3980	0.4728	0.4624	0.6404	0.6450
	2	BM25-U	0.1851	0.3644	0.4551	0.4643	0.4728	0.5168	0.6404	0.6455
	3	MMR	0.1812	0.3106	0.4388	0.3980	0.4616	0.4624	0.6472	0.6450
	4	MMR-U	0.1812	0.3640	0.4388	0.4571	0.4616	0.5106	0.6472	0.6523
	5	Rocchio	0.1851	0.3600	0.4551	0.4038	0.4728	0.4892	0.6404	0.6450

Table 1: Table of metrics at $M = 10$ (first page) and $M = 20$ (first and re-ranked second page) for each algorithm and for each data set. A superscript number refers to a metric value significantly above the value of the correspondingly numbered baseline in the table (using the Wilcoxon Signed Rank Test with p = 0.05).

λ increases, performance improves until we start to see gains for the higher values. This highlights that too much exploration can lead to detrimental performance in the first page that isn't recovered by the ranking in the second page, indicating the importance of tuning the parameter correctly. On the other hand, we do see that the optimal setting for λ is typically < 1, indicating that some exploration is beneficial and leads to improved performance across all metrics.

We also observe that the different datasets display different characteristics with regard to the effect that λ has on performance. For instance, for the difficult to rank Robust data set the optimal setting is $\lambda = 0.8$, indicating that exploration is not so beneficial in this case possibly due to the lack of relevant documents in the data set for each topic. Likewise, we find that a setting of $\lambda = 0.9$ is optimal for the TREC8 data set, although there is greater variation possibly owing to the easier to rank data. Finally, the WT10G data set also showed variation, which could be a result of the explicit, graded feedback available during re-ranking. For this data set we chose $\lambda = 0.7$. The differing data sets represent different types of query and search behaviour, and illustrate how λ can be tuned to improve performance over different types of search.

6.3 Comparison with Baselines

After setting the value of λ for each data set according to the optimal values discovered in the previous section (and also setting for the MMR variants $\lambda = 0.8$ for WT10G, $\lambda = 0.8$ for Robust and $\lambda = 0.9$ for TREC8 after some initial investigation), we sought to investigate how the algorithm compared with several baseline algorithms, each of which shares a feature with the IES algorithm; the Rocchio algorithm also performs relevance feedback; the MMR diversifies results; and also some variants that use our conditional

	Recall@		Precision@		nDCG@		MRR@	
	5	10	5	10	5	10	5	10
WT10G	0.2287	0.4540^{134}	0.3560	0.3940^{1345}	0.3823	0.4755^{13}	0.5900	0.5912
Robust	0.2308	0.3898^{13}	0.3851	0.3968^{135}	0.4594	0.5612^{135}	0.6652	0.6670^{1345}
TREC8	0.1803	0.3768^{134}	0.4388	0.4704^{1345}	0.4644	0.5228^{135}	0.6533	0.6583

Table 2: Table of metrics at $M = 5$ (first page) and $M = 10$ for the IES algorithm on each data set. The superscript numbers refer to significantly improved values over the baselines, as indicated in Table 1.

model update. We repeated the experiment as before over two pages with $M = 10$, and measured the recall, precision, nDCG and MRR in order to evaluate the overall search experience of the user who engaged with multiple pages (note that we find different values for MRR@10 and MRR@20 owing to the occasions where a relevant document wasn't located in the first M documents). The results can be seen in Table 1.

We first observe that the scores for the first page ranking are generally lower than that of the baselines, which is to be expected as we sacrifice immediate payoff by choosing to explore and diversify our initial ranking. We still find that the IES algorithm generally outperforms the MMR variants on the first step, particularly for the MRR metric, indicating improved diversification. In the second page ranking we find significant gains over the non-Bayesian update baselines, and some of the baselines using the conditional model update. It is worth noting that the BM25-U variant is simply the case of the IES algorithm with $\lambda = 1$. This demonstrates that our formulation is able to correctly respond to user feedback and generate a superior ranking on the second page that is tuned to the user's information need. We refer back to Figure 3 to show that the second page gains outweigh the first page losses for the values of λ that we have chosen.

Finally, in Table 2 we see a summary of results for the same experiment where we set $M = 5$, so as to demonstrate the IES algorithm's ability to accommodate different page sizes. We observe similar behaviour to before, with the IES algorithm significantly outperforming the baselines across data sets, indicating that even with less scope to optimise the first page, and less feedback to improve the second, the algorithm can perform well.

6.4 Number of Optimised Search Pages

Throughout this paper we have simplified our formulation by setting the number of pages to $T = 2$, so for this experiment, we observe the effect of setting $T = 3$. We performed the experiment as before where we display $M = 5$ documents on each page to the user and receive feedback, which is used to inform the next page's ranking. For the $T = 3$ variant, dynamic programming is used to generate the rankings for both pages 1 and 2 (with λ nominally set to 0.5), and the document's relevancy scores are updated after receiving feedback from pages 1 and 2. We compare against the IES algorithm with $T = 2$, where after page 1 we create a ranking of $2M$ documents, split between pages 2 and 3. Finally, we compare against the baseline ranking of $3M$ documents. The results can be seen in Table 3.

Algorithm	Recall@15	Prec@15	nDCG@15	MRR
$T = 3$	0.3249	0.3605	0.3638	0.4194
$T = 2$	0.3584	0.3905	0.4469	0.6371
Baseline	0.2955	0.3293	0.3931	0.5961

Table 3: Table showing metric scores at rank 15 for the baseline and two variants of the IES algorithm, where T is set to 3 and 2.

We can see from the results that whilst the $T = 3$ variant still offers improved results over the baseline (except in the case of MRR), the performance is also worse than the $T = 2$ case. This is an example of too much exploration negatively affecting the results; when T is set to 3, the algorithm creates exploratory rankings in both the first and second pages, before it settles on an optimal ranking for the user on the third page. Except for on particular queries, a user engaging in exploratory search should be able to provide sufficient feedback on the first page in order to optimise the remaining pages, and so setting $T = 2$ is ideal in this situation.

7. DISCUSSION AND CONCLUSION

In this paper we have considered how to optimally solve the problem of utilising relevance feedback to improve a search ranking over two or more result pages. Our solution differs from other research in that we consider document similarity and document dependence when optimally choosing rankings. By doing this, we are able to choose more effective, exploratory rankings in the first results page that explore dissimilar documents, maximizing what can be learned from relevance feedback. This feedback can then be exploited to provide a much improved secondary ranking, in a way that is unobtrusive and intuitive to the user. We have demonstrated how this works in some toy examples, and formulated a tractable approximation that is able to practically rank documents. Using appropriate text collections, we have shown demonstrable improvements over a number of similar baselines. In addition, we have shown that exploration of documents does occur during the first results page and that this leads to improved performance in the second results page over a number of IR metrics.

Some issues that arise in the use of this algorithm include trying to determine the optimal setting for λ, which may be non-trivial, although it could be learned from search click log data by classifying the query intent [3] and associating intent with values for λ. Also, in our experiments we used BM25 scores as our underlying ranking mechanism and cosine similarity to generate the covariance matrix, it is for future work to investigate combining IES with different ranking algorithms and similarity metrics. We would also like to further investigate the effect that M plays on the optimal ranking and setting for λ; larger values of M should discourage exploration as first page rankings contain more relevant documents. Also, whilst our experiments were able to handle the binary case, our multivariate distribution assumption is a better model for graded relevance, and our approximation methods and the sequential ranking decision can generate non optimal rankings that may cause detrimental performance.

In the future, we intend to continue developing this technique, including attempting to implement it into a live system or in the form of a browser extension. We also note that multi-page interactive retrieval is a sub-problem in the larger issue of exploratory search, in particular, designing unobtrusive user interfaces to search systems that can adapt to user's information needs. Furthermore, as previously stated we can consider our formulation in the context of POMDP and MAB research [9, 20], which may lead to a general solution that is applicable to other applications, or instead may grant access to different solution and approximation techniques.

8. REFERENCES

[1] Agichtein, E., Brill, E., and Dumais, S. Improving web search ranking by incorporating user behavior information. SIGIR '06, ACM, pp. 19–26.

[2] Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. Diversifying search results. WSDM '09, ACM, pp. 5–14.

[3] Ashkan, A., Clarke, C. L., Agichtein, E., and Guo, Q. Classifying and characterizing query intent. ECIR '09, Springer-Verlag, pp. 578–586.

[4] Bellman, R. E. *Dynamic Programming.* Dover Publications, Incorporated, 2003.

[5] Billerbeck, B., and Zobel, J. Questioning query expansion: an examination of behaviour and parameters. ADC '04, Australian Computer Society, Inc., pp. 69–76.

[6] Brandt, C., Joachims, T., Yue, Y., and Bank, J. Dynamic ranked retrieval. WSDM '11, ACM, pp. 247–256.

[7] Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. Selecting good expansion terms for pseudo-relevance feedback. SIGIR '08, pp. 243–250.

[8] Carbonell, J., and Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *In Research and Development in Information Retrieval* (1998), pp. 335–336.

[9] Cassandra, A. A survey of POMDP applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes* (1998), pp. 17–24.

[10] Chen, H., and Karger, D. R. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR* (2006), pp. 429–436.

[11] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. Novelty and diversity in information retrieval evaluation. SIGIR '08, ACM, pp. 659–666.

[12] Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. An experimental comparison of click position-bias models. WSDM '08, ACM, pp. 87–94.

[13] Fox, S., Karnawat, K., Mydland, M., Dumais, S., and White, T. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst. 23* (April 2005), 147–168.

[14] Hemmje, M. A 3d based user interface for information retrieval systems. In *Proceedings of the IEEE Visualization '93 Workshop on Database Issues for Data Visualization* (1993), Springer-Verlag, pp. 194–209.

[15] Hofmann, K., Whiteson, S., and de Rijke, M. Balancing exploration and exploitation in learning to rank online. In *Proceedings of the 33rd European conference on Advances in information retrieval* (2011), ECIR'11, pp. 251–263.

[16] Joachims, T. Optimizing search engines using clickthrough data. KDD '02, ACM, pp. 133–142.

[17] Koenemann, J., and Belkin, N. J. A case for interaction: a study of interactive information retrieval behavior and effectiveness. CHI '96, ACM, pp. 205–212.

[18] Kules, B., and Capra, R. Visualizing stages during an exploratory search session. HCIR '11.

[19] Morita, M., and Shinoda, Y. Information filtering based on user behavior analysis and best match text retrieval. SIGIR '94, Springer-Verlag New York, Inc., pp. 272–281.

[20] Pandey, S., Chakrabarti, D., and Agarwal, D. Multi-armed bandit problems with dependent arms. ICML '07, ACM, pp. 721–728.

[21] Radlinski, F., Kleinberg, R., and Joachims, T. Learning diverse rankings with multi-armed bandits. ICML '08, ACM, pp. 784–791.

[22] Radlinski, F., Szummer, M., and Craswell, N. Inferring query intent from reformulations and clicks. WWW '10, pp. 1171–1172.

[23] Raman, K., Joachims, T., and Shivaswamy, P. Structured learning of two-level dynamic rankings. CIKM '11, ACM, pp. 291–296.

[24] Robertson, S., and Hull, D. A. The trec-9 filtering track final report. In *TREC* (2001), pp. 25–40.

[25] Robertson, S. E. The Probability Ranking Principle in IR. *Journal of Documentation 33*, 4 (1977), 294–304.

[26] Rocchio, J. *Relevance Feedback in Information Retrieval.* 1971, pp. 313–323.

[27] Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. Circuits Syst. Video Techn. 8*, 5 (1998), 644–655.

[28] Ruthven, I. Re-examining the potential effectiveness of interactive query expansion. SIGIR '03, ACM, pp. 213–220.

[29] Shen, X., Tan, B., and Zhai, C. Context-sensitive information retrieval using implicit feedback. SIGIR '05, ACM, pp. 43–50.

[30] Shen, X., Tan, B., and Zhai, C. Implicit user modeling for personalized search. CIKM '05, ACM, pp. 824–831.

[31] Shen, X., and Zhai, C. Active feedback in ad hoc information retrieval. SIGIR '05, ACM, pp. 59–66.

[32] Sloan, M., and Wang, J. Iterative expectation for multi period information retrieval. In *Workshop on Web Search Click Data* (2013), WSCD'13.

[33] Spink, A., Jansen, B. J., and Ozmultu, C. H. Use of query reformulation and relevance feedback by Excite users. *Internet Research: Electronic Networking Applications and Policy 10*, 4 (2000), 317–328.

[34] Stephen, R., and Hugo, Z. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval 3*, 4 (2009).

[35] Sugiyama, K., Hatano, K., and Yoshikawa, M. Adaptive web search based on user profile constructed without any effort from users. WWW '04, ACM, pp. 675–684.

[36] Voorhees, E. M. The cluster hypothesis revisited. SIGIR '85, pp. 188–196.

[37] Wang, J., and Zhu, J. Portfolio theory of information retrieval. SIGIR' 09, ACM, pp. 115–122.

[38] Wang, J., and Zhu, J. On statistical analysis and optimization of information retrieval effectiveness metrics. SIGIR '10, pp. 226–233.

[39] White, R. W., and Ruthven, I. A study of interface support mechanisms for interactive information retrieval. *J. Am. Soc. Inf. Sci. Technol. 57*, 7 (May 2006), 933–948.

[40] White, R. W., Ruthven, I., and Jose, J. M. A study of factors affecting the utility of implicit relevance feedback. ACM Press, pp. 35–42.

[41] Xu, Z., and Akella, R. Active relevance feedback for difficult queries. CIKM '08, ACM, pp. 459–468.

[42] Yan, R., Hauptmann, A. G., and Jin, R. Negative pseudo-relevance feedback in content-based video retrieval. MULTIMEDIA '03, ACM, pp. 343–346.

[43] Zamir, O., and Etzioni, O. Grouper: a dynamic clustering interface to web search results. WWW '99, Elsevier North-Holland, Inc., pp. 1361–1374.

[44] Zhang, L., and Zhang, Y. Interactive retrieval based on faceted feedback. SIGIR '10, ACM, pp. 363–370.