A Portfolio-Armed Bandit Machine Approach to Multi-Period Information Retrieval Modelling

Marc Sloan

Department of Computer Science

University College London
marc.sloan.10@ucl.ac.uk

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Master of Research

of the

University of London.

Department of Computer Science

University College London

August 23, 2011

Abstract

This dissertation investigates how to devise optimal ranking strategies over time with which to dynamically improve a ranking algorithm. Many Information Retrieval (IR) models make an assumption that the scores (relevance) for documents given an information need are static and that the goal is to estimate them as accurately as possible over one period. This means that the dynamic nature of document relevance and thus the underlying retrieval system has been largely ignored. In this paper, we explicitly formulate a general Multi-Period Information Retrieval problem, where we consider retrieval as a stochastic yet controllable process. The rank actions during the process continuously control the retrieval system's dynamics, and an optimal ranking policy is found in order to maximise the overall users' satisfaction at the end of the process as much as possible.

Our derivations show interesting properties about how the posterior probability of the documents relevancy evolves from users' feedbacks through clicks. Based on the Multi-armed Bandit algorithm and Portfolio Theory of IR, we propose a simple dynamic ranking rule that takes both rank bias and the dependency of clicks into account. We verify the versatility and robustness of our algorithms in a number of experiments written using Matlab, and also demonstrate improved performance over several strong baselines. The experiments demonstrate that our approaches can handle the rank bias and the dependency of users' feedback on ranked documents, and as a result significant performance gains have been achieved.

Acknowledgements

I would first like to offer my sincere gratitude to Dr. Jun Wang for his supervision, comments, ideas and input into this work and for continually challenging me to develop further the ideas in this dissertation, and I look forward to starting my PhD under his guidance.

Secondly I would like to thank the UK PhD Centre for Financial Computing for their resources and funding, and Prof. Philip Treleaven and Yonita Carter for their support and help during my studies.

I would also like to thank Prof. John Shawe-Taylor for his insights into how this work could be developed in the future.

Finally, I wish to thank my mother for her support and also Steph for enduring my nocturnal work sessions and looking after and encouraging me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we provide a brief overview of the work contained in this dissertation. We first define the overall objective of this work and why it is of interest, and then briefly outline some of the necessary background material to our formulation, indicating where it unites the fields of finance and IR. We then summarise our contribution and results, and finish by describing the structure of the paper.

1.1 Problem Overview and Dissertation Objective

In Information Retrieval (IR) research, we study mathematical models of IR systems because they provide formal and quantitative tools for us to understand the underlying retrieval mechanisms, and at the same time, lead to the development of practical retrieval algorithms and systems. Yet, the mainstream IR models and theories have been largely devoted to maximising performance over one period. This means that the dynamic nature of document relevance and thus the retrieval system has been largely ignored. For instance, to estimate the relevancy, statistical language models and relevance models assign a static score to each document given a query by analyzing term statistics [RZ09, Zha08], whereas link analysis such as PageRank approaches this by looking at the long term stabilised visit rates of web documents [Fra11]. A key objective in this paper is how to devise optimal ranking strategies over time with which to dynamically improve a ranking algorithm, and in the end maximise a certain expected utility.

We formulate a control theory based optimisation framework, modelling the problem as a dynamic system requiring a control signal (the rank action) and a feedback mechanism (the clickthroughs), where the resulting framework is flexible and many click models can be naturally integrated. Interestingly, similar to the portfolio theory of IR [WZ09], it also shows that the objective function can be conveniently decomposed into two parts, the mean and the variance of the users satisfactions (e.g., clicks), where the mean deals with rank bias and the variance tackles the click dependency. Traditionally, IR systems work under the Probability Ranking Principle (PRP), which states that documents should be ranked according to their probability of relevance with an independence assumption [Rob77], whereas the portfolio theory of IR extends the ranking principle to address the dependency issue and promote diversity. This is one instance of a number of recent developments in combining the fields of IR and finance, which has been further added to by our interdisciplinary work.

Considering the problem as a multi-armed bandit (MAB) provides a simplified iterative solution,

and so we focus was on how a multi-armed bandit could be used to learn a correct ranking over time. There have been a few attempts at using MABs in a ranking context, most notably [RKJ08], but the multi-play MAB algorithm was largely ignored and so we have developed this into a working ranking algorithm. We adopt the UCB1 algorithm [ACBF02], but provide a rather general treatment of the multi-period IR problem where other algorithms such as dynamic programming and Gittins index algorithms can be plugged in. The UCB1 based multi-play MAB algorithm isn't itself novel [LGJP07], but applying it in the context of our multi-period IR problem is, and motivates the development of our multi-period framework so as to incorporate different click models and document dependency.

Thus, we subsequently formulate two multi-armed bandit based algorithms that respectively, take into account rank bias and maximize the expected utility of a ranking, and minimise the variance of a ranking thereby introducing topic diversity. We confirm the theoretical insights using simulated experiments and observe improved performance in addressing rank bias and dependency of user feedbacks on retrieved documents.

The writer of this dissertation has a background in Computer Science and IR and has only recently become involved in the study of finance and economics, but feels that there is much to be gained in the crossing over of the two fields. There is already evidence of other financial models being used in IR, such as the use of utility functions and hedonic regression in sentiment analysis [GIS07, AGI07] and production theory in interactive IR [Azz11]. In addition, MABs are also finding use in finance and it is conceivable that the work outlined in this dissertation could be used in asset allocation.

1.2 Dissertation Structure

We start with an introduction into the field of information retrieval for the more financially minded reader, and emphasise the aspects that are relevant to the work in this dissertation, outlining probability of relevance and click models. We then present the portfolio theory of IR and define topic diversity, with a look at how other financial models are currently being used in IR. We continue with learning using clickthroughs and how this can be achieved dynamically, first by making use of control theory, and then simplified using a multi-armed bandit algorithm.

We then define our problem statement and formulate our multi-period IR framework in detail and derive a solution that allows different click models. After presenting an appropriate mixed-click model, we introduce our two multi-armed bandit algorithms, which we then test in number of experimental simulations. Finally, we describe our results and present our conclusions and thoughts on improvements and future work.

Chapter 1: Introduction

In this chapter we provide a brief overview of the work contained in this dissertation. We first define the overall objective of this work and why it is of interest, and then briefly outline some of the necessary background material to our formulation, indicating where it unites the fields of finance and IR. We then summarise our contribution and results, and finish by describing the structure of the paper.

Chapter 2: Background

In this chapter we will present all of the necessary background material that underlies the work in this dissertation. We will start by defining the field of Information Retrieval and how our work in ranking and feedback fits into it, and then we will explore how clickthroughs can be used as a form of implicit feedback, and why these require a click model to be interpreted properly. We then introduce the idea of diversity in ranking and the portfolio theory of IR, and consider other uses of economics in IR. We then move onto dynamic ranking and how control theory fits into our formulation, before finishing with multi-armed bandit algorithms and a review of the related literature.

Chapter 3: Multi-period IR Modelling

In this chapter we explicitly derive our optimal control formulation for the dynamic ranking scenario, and show that it naturally allows us to the study the expectation and variance as separate cases. We study the first case, where we introduce our mixed-clicks model and combine this with an expectation maximisation update rule to arrive at an iterative formula for maximising the expectation, which is then shown as part of a multi-armed bandit algorithm. We then consider the variance and introduce portfolio theory and diversity into the problem and finally derive a similar update rule and apply it to another multi-armed bandit algorithm.

Chapter 4: Experiments

In the previous chapter we studied the properties of document ranking over time, under the assumption of certain click models. This chapter continues the study by evaluating the resulting practical ranking strategies through simulations. We mainly intend to 1) understand the optimality of the proposed algorithms; 2) study the impact of parameters; and 3) analyse the robustness against noise and changing environments. Three resulting ranking strategies were evaluated. Namely, we have the UCB algorithm with Interactive Expectation from Mixed-Click model (denoted as UCB-IE-MC) and that from Examination Hypothesis (denoted as UCB-IE-EH), which are intended to deal with rank bias. To address the diversity and dependency of the clicks, we have the Portfolio-Armed Bandit algorithm (denoted as PAB). Note that they can be naturally combined in practice, however, to make our evaluation targeted, we do not combine them here and have separated the experiments into two scenarios: one for rank bias and the other for diversity.

Chapter 5: Conclusion and Discussion

In this chapter, we summarize our work and the results obtained in our experiments, and show how they meet the objectives of this dissertation. We also make note of the limitations of our theoretical model with suggestions for improvements. Then we discuss the future directions that this research could take, including the combination of the two algorithms and the need to continue the experiments using real data. Furthermore, variations of the formulation are considered and applied to a broader research topic. We end by examining how our work could be used in the field of finance.

Chapter 2

Background

In this chapter we will present all of the necessary background material that underlies the work in this dissertation. We will start by defining the field of Information Retrieval and how our work in ranking and feedback fits into it, and then we will explore how clickthroughs can be used as a form of implicit feedback, and why these require a click model to be interpreted properly. We then introduce the idea of diversity in ranking and the portfolio theory of IR, and consider other uses of economics in IR. We then move onto dynamic ranking and how control theory fits into our formulation, before finishing with multi-armed bandit algorithms and a review of the related literature.

2.1 Information Retrieval

Information Retrieval (IR) is concerned with the storage, indexing and retrieval of information describing or contained in documents, whereby the documents can be of any type of media. Historically, the field has its roots in library sciences and record keeping in the 1950's, when computers began to be considered a viable solution to the increasing quantity of information that needed to be stored and referenced. IR techniques have since improved in sophistication and reach, keeping abreast of complex new document types such as digital images and video and encompassing huge information resources such as the World Wide Web, best typified by modern day Internet search engines.

The process of retrieving information can be broken down into a number of discrete stages, represented by Figure 2.1 and described below:

Information Need The underlying information that a user is trying to acquire, though they themselves may not know exactly what they're looking for until they've already started the IR process. This need is typically represented as some form of **query**.

Information Items The documents that will be searched in order to fulfil the information need. Before retrieval can occur the items will need to be **indexed** and stored in an appropriate way.

Retrieved Items The items that the IR system considers most relevant to the information need of the user, typically these will be ranked in order of decreasing relevancy. Often an IR system will receive some form of feedback from the user indicating the success of the search.

Additionally, a user will often go through multiple iterations of this process, further refining their query

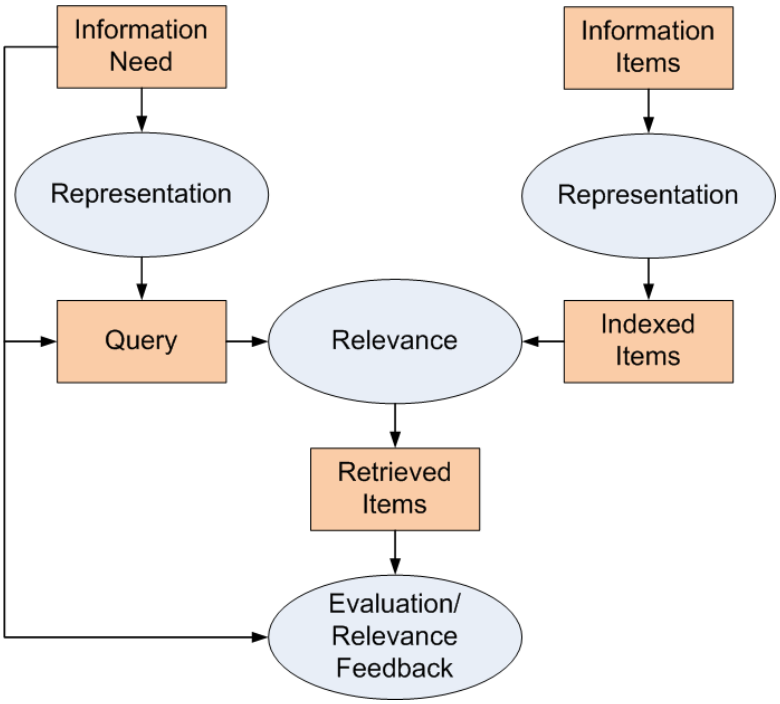

Figure 2.1: Representation of an IR system, where ellipses denote IR tasks and rectangles the outcomes

and narrowing down their information needs until finally satisfied. The research in this dissertation concerns the final two stages of the IR process, that of displaying a ranked list of documents to a user, and then eliciting feedback to further improve future rankings.

2.1.1 Probability of Relevance

The most common way of determining if a document is relevant to an information need is to calculate its *probability of relevance*, the posterior probability that, given the query issued by the user, the document is relevant. How to calculate this is a fundamental feature of IR and over the years numerous theorems and models have been proposed and evaluated.

Initially, IR systems were satisfied purely with returning results with a binary relevance of either non-relevant or relevant. An early example of such a system is the *Boolean Model*, where words from queries are matched directly to the words found in documents and combinations of words can be searched for by including appropriate AND and OR operators. Due to the unintuitive query format this model has been superseded by newer models that allow natural language queries, although Boolean search is still popular amongst professionals due to its preciseness (for instance the Westlaw legal search engine[1]).

More complicated models have since been developed, including the Robertson-Spärck Jones probabilistic model and the celebrated BM25 weighting [RZ09], the language modelling approaches [Zha08] and the latest developments on Learning to Rank [GLW09, Joa02]. Each of these methods attempts to discern a continuous probability value or a relative relevance score that compares the suitability of displaying documents for a given query. In the case of the first two examples, the content of the documents (i.e. the terms contained in the document) is compared to the query to determine the best match; in the

[1] http://www.westlaw.com

latter case, correct classifications are learnt from historical data.

2.1.1.1 Probability Ranking Principle

A drawback of the Boolean model is that it returns all documents it considers relevant with no particular ordering. Given that most modern IR systems can return thousands or even millions of documents, it is essential that such a system be able to prioritise displaying those documents it deems most suitable. In 1979, [Rob77] defined the *Probability Ranking Principle* (PRP) which states that the effectiveness of a retrieval system is maximised when displaying documents in decreasing order of estimated probability of relevance; it is most worthwhile to all users to display the document that has the highest probability of relevance first, followed by the second highest and so forth.

This intuitive result formalised the optimal strategy for displaying ranked documents to users and underlies most modern IR models. Nonetheless, this principle also makes the assumption that we are able to accurately estimate probability of relevance and also that the relevancies are independent of one another. We shall see in Section 2.2 that this is indeed not always the case, and we will develop further an alternative that incorporates portfolio theory from finance.

2.1.2 Clickthroughs

Before continuing further with ranking documents, we will next consider methods for gaining feedback from users. Early research in relevance feedback was personalized to each user and dealt with the difficulty in coming up with an appropriate query for a given information need. After an initial, tentative search, a user could select those documents most relevant to their information need, possibly using a grading schema indicating the level of relevance, resulting in a reformulation of their query into a more relevant one [RL03]. Unfortunately, such explicit methods of obtaining feedback were often met with unenthusiasm by users, and are little used in modern IR systems.

Instead, [FKM$^+$05] evaluated a number of implicit measures against explicit user satisfaction measurements to determine those most effective for representing user feedback. The implicit measures covered a range of observable user behaviours including scrolling, number of results viewed, time spent on search pages and the results that were clicked, with the latter two correlating well with user satisfaction.

Clicked results (or *clickthroughs*) in particular have established themselves as a common way to infer user satisfaction; the act of clicking on a returned document is an indication from the user that that document is probably relevant to the query. In addition, recording clickthroughs is unintrusive and cheap and they can be recorded easily by the server hosting the search engine. Their abundance allows them to be readily used to evaluate ranking algorithms and allow them to learn a correct ranking using machine learning techniques such as SVMs [Joa02, ABD06].

Nonetheless, they are not a perfect representation of user interest and are subject to noise and bias when not interpreted correctly. [SSBT08] found that the clickthrough behaviour didn't vary systematically with the quality of a search engine, and that there were large differences in observed behaviour between different topics and different users. Furthermore, a well studied bias is the *rank bias* of a search engine, confirmed by [JGP$^+$05] as part of an eye tracking study. The bias is a consequence of the *trust* that a user places in the ability of a search engine to correctly find and rank relevant documents. A user

typically reads a list of search results from top to bottom with decreasing interest, making them considerably more likely to view and click on a top ranked document, even if lower ranked documents are actually more relevant.

2.1.2.1 Click Models

To account for rank bias when considering clickthroughs, a number of *click models* have been proposed to correctly calculate the probability of relevance for clicked documents, these being Bayesian probabilistic models representing a users typical behaviour. Let q be a given query, d_i a document displayed at position i where $i = 1$ is the highest rank, r_{d_i} the relevance of d_i to the query, C the binary click event and E a hidden random binary variable indicating whether a user has examined a document; then an early example of a click model is the Examination Hypothesis given below:

$$P(C = 1|q, d_i) = \underbrace{P(C = 1|d_i, q, E = 1)}_{r_{d_i}} P(E = 1|i) \tag{2.1}$$

This model assumes that if a clickthrough occurs, then the document must have been examined and that the probability of the document being examined is dependent on the position of the document. This model makes the strong assumption that users examine documents indiscriminately and that a user will examine all documents in a ranking.

The Cascade Model removed this assumption and was shown by [CZTR08] to be an improvement over the examination model. It introduced a dependency on the documents ranked before a clicked document, whereby the first document is always examined; documents are examined strictly from top to bottom and if a user clicks on a document, then the search session is terminated. Whilst effective, this model too has been superseded by improved models, including the Click Chain Model [GLK$^+$09], which removes the assumption that the user stops examining after clicking a document, and the General Click Model [ZCM$^+$10], which generalises the previous models by using three continuous random variables R_i, A_i and B_i to capture the tendency of the user to respectively, click on a document, examine the next document if a click didn't occur and examine the next document if a click did occur, as shown more clearly in Figure 2.2.

In this dissertation we will be formulating a new click model and also creating a framework in which other click models can be incorporated, and we will demonstrate this by implementing the simple Examination Hypothesis model and performing experiments using it.

2.2 Diverse Ranking

Before progressing further with click models and using clickthroughs to rank, we will first consider another increasingly important aspect of document ranking, that of topic diversity. We have already encountered the probability ranking principle and alluded to its shortcomings and in this section we will develop this idea further.

Queries expressed in natural language can be ambiguous, particularly when expressed using words that contain numerous homonyms. Consider the query term 'Jaguar'; does the users information need concern animals or the car manufacturer? In this case the query can fall into two possible topics, and it

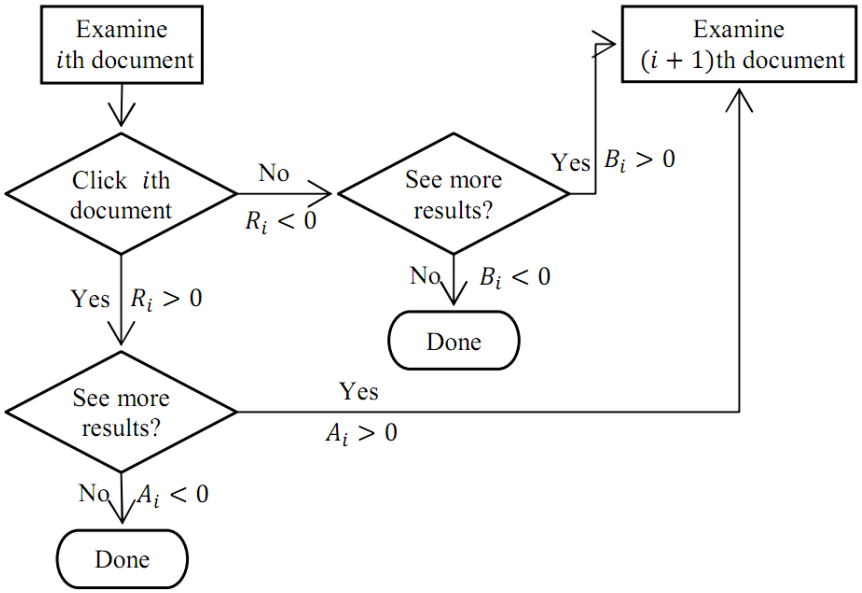

Figure 2.2: Flow chart for the General Click Model, where A_i, B_i and R_i are hidden variables [ZCM+10]

is not possible to discern which of the two the user is interested based on the query alone. A traditional ranking algorithm that makes use of the PRP would perhaps display documents that contained more instances of the term 'Jaguar', and as such, may bias the results in favour of only one of the topics. Nonetheless, it seems intuitive that to give good performance for all users overall, perhaps documents from other topics, despite being deemed less relevant, should also be displayed. The PRP no longer applies in this case because we are now introducing *dependencies* between displayed documents and those ranked before them. Continuing with our example, given that the first two ranked documents concern the topic cars, then the probability that the next ranked document belongs to the animals topic should be higher.

Diversity in ranking is currently an active area of research, an early attempt of which is the Maximal Marginal Relevance [CG98], whereby an already ranked list of documents is diversified by displaying documents that are dissimilar from previously displayed documents (using some form of similarity metric), an approach similar to the more recent [ZGVA07]. [CK06] introduced diversity by generalizing the PRP and purposefully displaying less documents to a user whilst also explicitly calculating the document dependency, and [RKJ08] implicitly learns a diverse ranking by dynamically monitoring clickthroughs for each displayed document.

2.2.1 Portfolio Theory

More recently, [WZ09] proposes an alternative method of diversifying search results that makes use of modern portfolio theory taken from finance. Portfolio theory was devised in the 50's by [Mar57] as a way of choosing a portfolio of investments so as to reduce the overall risk (here given by the variance of the portfolio). This is achieved by investing in assets that are negatively correlated with one another i.e. diverse from one another, so that a loss of value in one of the assets is balanced by an increase in value

in another. Additionally, it is possible to pre-define an acceptable level of risk and construct a portfolio that gives the maximum expected return for that level of risk.

Despite the method being challenged by modern financial theorems and used less in portfolio management, it can still be made applicable to the field of IR where, instead of diversifying a portfolio of assets we instead diversify a ranking of documents. If each document has a relevancy r_i at position i (where $1 \leq i \leq M$), then the overall expected relevancy of a ranking is given by

$$E[R_M] = \sum_{i=1}^{M} w_i E[r_i] \tag{2.2}$$

where w_i is a weight value representing the rank bias at position i and R_M is the relevance of the overall ranking, a value we wish to maximise. The variance of the ranking can thus be calculated as

$$Var[R_M] = \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j c_{i,j} \tag{2.3}$$

$$= \sum_{i=1}^{M} w_i^2 c_{i,i} + 2\sum_{i=1}^{M}\sum_{j=i+1}^{M} w_i w_j c_{i,j} \tag{2.4}$$

$$= \sum_{i=1}^{M} w_i^2 \sigma_i^2 + 2\sum_{i=1}^{M}\sum_{j=i+1}^{M} w_i w_j \sigma_i \sigma_j \rho_{i,j} \tag{2.5}$$

where $c_{i,j}$ is the covariance between documents i and j, σ_i the variance of document i and $\rho_{i,j}$ the *correlation coefficient*. Using this formulation, it was found that a non-optimal but efficient way of maximising $E[R_M]$ whilst minimising $Var[R_M]$ in a ranked list is to first choose the highest ranking document, then for each subsequent rank k choose the document that maximises

$$E[r_k] - \lambda w_k \sigma_k^2 - 2\lambda \sum_{i=1}^{k-1} w_i \sigma_i \sigma_k \rho_{i,k} \tag{2.6}$$

where λ is a tuneable parameter.

Later in this dissertation we will be making use of this theorem and the ranking rule given by Eq. (2.6) in our own diversifying algorithm.

2.2.1.1 Finance in IR

The use of portfolio theory in IR is not restricted only to the case of diversifying search results, but also has been used to trade off the risk of incorrectly expanding a query during query reformulation [CT08], and is one of a number of financial models that are progressively appearing in IR research.

An early example is the use of economic utility functions and hedonic regression in the sentiment analysis of customer product reviews (such as on Amazon.com), in an attempt to assign a dollar value to such comments and infer if such reviews harmed or promoted products [GIS07, AGI07]. Another use of utility functions is to learn which attributes of a document (in this case, hotel metadata) were considered more valuable to a user, so that, in this case, the utility of a hotel booking could be compared to its price and ranked according to value for money for the user [LGI11].

A recent example made use of production theory to model an interactive IR system [Azz11], where the number of queries issued and documents read by a user in an IR session are considered the raw

materials (the input), the technology used is the search engine, and the output is the cumulative gain of satisfying the users information need. This model allowed the researcher to find the optimal balance between queries and documents for different types of information search, which they confirmed using empirical data.

2.3 Learning Using Clickthroughs

After choosing an appropriate click model, clickthroughs can start to be used to reliably train ranking algorithms. Typically, this is achieved by using clickthrough logs as a training set for supervised learning algorithms, where the click model is used to interpret the results from the logs and multiple IR scoring metrics (such as BM25 and PageRank scores) are used as document features. This training is usually carried out before the ranking algorithm is used on a live search engine (see [Joa02, ZJ07, ABD06] for examples), and periodically updated so that the algorithm is able to incorporate new documents and remove spam, allowing the search engine to remain competitive and relevant.

A drawback of these static, offline techniques is that after the training phase they are vulnerable to changes in user satisfaction and may no longer offer suitable results. The information need underlying a query can be subject to the context of changing world events, or depend on seasonal and calendar variation [ZHL+06], and as such the relevance of documents can change over time [YL11]. Training offline regularly can be expensive and time-consuming and so an online algorithm that adapts to changes over time is ideal in this situation.

2.3.1 Control Theory

Control theory is a mainstay of mechanical engineering and is used in the modelling of dynamical behaviour over time. Such dynamic systems usually consist of some form of control signal that alters the state of the system, which is detected by a sensor and this information is fed back to the control signal so that it can correct any mistakes and keep the system in some form of equilibrium [Oga01]. As discussed in the previous section, the relevance of a document is in fact stochastic and liable to change over time, and in this dissertation we will model this dynamic system by making use of control theory.

Not only is the document relevance stochastic, but also the relevance estimation in many cases is multi-period, and the initial guess of a relevance model must be tested over time so that incorrectly ranked documents can be removed and newly relevant documents (or new ones) can be tried and accordingly ranked. Thus, our dynamic formulation will take the form of a multi-period IR problem, which we will explicitly define later.

2.4 Multi-Armed Bandits

A well known type of online learning problem is the *multi-armed bandit* (MAB) problem, which will be motivated in this section and used extensively later in this dissertation. The multi-armed bandit problem is a classic statistical resource allocation problem, usually described using the analogy of a casino with multiple one-armed bandit slot machines. Each of the slot machines has a different probability distribution of rewards, and the problem is to find the optimal strategy for playing them so as to maximise

the overall rewards over some time horizon. A key characteristic of the solutions to the problem is the trade-off between the exploration of the arms, where we learn the distributions of each of the arms by playing them and sampling their rewards, and the exploitation of the arms that have been found to have the highest probability of reward.

[Git79] formulated an optimal solution, which calculates for each arm a scalar value known as the *Gittens index*, and at each time step the arm with the highest Gittens index value is the arm chosen to play. This work was further refined by [LR85] where less computationally expensive allocation rules were defined that weren't necessarily optimal, but guaranteed asymptotic performance. Following on from this work came the practical algorithms devised by [ACBF02], including the popular UCB1 algorithm used in this dissertation and given below:

1. Play each arm once
2. Play arm i that maximizes
$$\underset{i}{\operatorname{argmax}} \frac{X_i}{Y_i} + \sqrt{\frac{2 \ln t}{Y_i(t)}} \tag{2.7}$$
3. Update
$$X_i = X_i + \text{reward} \tag{2.8}$$
$$Y_i = Y_i + 1 \tag{2.9}$$
4. Repeat steps 3 to 6

Here, X_i is used to keep track of the rewards received from arm i (each reward is a continuous value between 0 and 1) and Y_i records the number of times we have played arm i. The idea behind this algorithm is that the square root term in Eq. (2.7) is the variance of the estimated reward $\frac{X_i}{Y_i}$, and so Eq. (2.7) is our *upper confidence bound* of the expected reward. As such, those arms that have been played less will exhibit a larger variance over time until eventually they are played, and their expected reward updated with the new reward information learned. Thus, exploration is conducted throughout the lifetime of this algorithm, although less so as the true parameters for the probability distributions are discovered and exploited.

2.4.1 Multi-Armed Bandits Research

2.4.1.1 Multi-Armed Bandits in IR

The exploration/exploitation and online nature of MAB algorithms make them ideal candidates for employing in IR systems that aim to be responsive to changing relevancies, and are seeing increasing use in the field. They have been used to learn when to display particular web advertisements [CKRU08] or which types of news stories to display to different users [KRS10], and even to learn a diverse ranking of documents [RKJ08], the subject of this dissertation.

In addition, the MAB problem itself is an active area of research and new variations and algorithms, operating under different assumptions have recently been formulated, making them more suitable for use

in future IR research. Some examples of the differing MAB problems include restless bandits [GMS07], where the probability distributions are allowed to change over time, adversarial bandits [ACBFS95] where an adversary controls the rewards for each arm in a worst case scenario, and multi-play MABs [AVW86] where multiple arms can be played at each time step. In this dissertation we develop further the multi-play MAB and combine it with the UCB1 algorithm and a click model to learn a correct ranking of documents in an online fashion.

2.4.1.2 Multi-Armed Bandits in Finance

The multi-armed bandit problem has also been used in finance, beginning with [Rot74]'s work in the 70's using a MAB to model how a business should exploratively price their product so as to learn the market and find an optimal value. Similar work was carried out by [BV96], this time using a MAB to model how a consumer should compare the products of competing businesses, and how this may affect the overall market if it is known to the businesses the strategy of the consumer. Another early paper concerned the modelling of job turnover as an MAB, whereby employers were considered as the arms and employees learnt their productivity at each business by being employed, and either being laid off or continuing with work [Jov79].

Furthermore, modern uses of MABs in finance have been the modelling of venture capitalist investments; whether capturing how successful VC's tend to be the ones who make explorative investments in the hope of learning for future reference [Sor07], or by providing strategies to aid in choosing when to continue funding research oriented projects or withdraw investment [BH05]. [BBF$^+$03] gives a good summary of how many optimization problems in option theory, dynamic allocation problems and microeconomic theory of intertemporal consumption can also be reduced to the multi-armed bandit problem. We shall later investigate how the MAB algorithms presented in this dissertation may be used in the financial sector.

2.5 Related Work

A relevant theoretical development similar to our multi-period IR framework is found in [Fuh08], which extends the PRP to cover interactive retrieval, the goal being to design interactive strategies within a single search session for a given user. By contrast, we consider a rather different scenario where the same information need is repeated over time for a population of users. Furthermore, our mixed-click model is an extension of the Examination Hypothesis model and a more general variation of the mixture model found in [CZTR08].

Our UCB formulations are an extension of the multi-play UCB based MAB found in [LGJP07], which is itself developed from the UCB1 algorithm given by [ACBF02]. Similar multi-play MABs for use in IR have been developed by [KRS10, UNK10], although instead of considering the more practical UCB1 algorithms they instead develop further the stochastic adversarial Exp3 MAB algorithm, which optimises for the worst case scenario and is useful primarily for determining theoretical limitations.

Relevant to our work on the portfolio armed bandit is [RKJ08]'s multi-armed bandit algorithm that is able to dynamically learn a diverse ranking of documents over time, and whose effectiveness is con-

firmed using simulations that also come with theoretical guarantees. On the other hand, our algorithm was designed to improve on the limitations of their work, namely that diversity isn't explicitly incorporated into their algorithm but rather learnt as a by-product of recording clickthroughs, and we provide a comparison of the two algorithms. The UCB formulation of our algorithm is also closely related to the recent efforts for introducing dependencies between MAB arms, such as the method in [DGST09], although their aim was to reduce the learning time of such an algorithm rather than diversification.

Chapter 3

Multi-period IR Modelling

In this chapter we explicitly derive our optimal control formulation for the dynamic ranking scenario, and show that it naturally allows us to the study the expectation and variance as separate cases. We study the first case, where we introduce our mixed-clicks model and combine this with an expectation maximisation update rule to arrive at an iterative formula for maximising the expectation, which is then shown as part of a multi-armed bandit algorithm. We then consider the variance and introduce portfolio theory and diversity into the problem and finally derive a similar update rule and apply it to another multi-armed bandit algorithm.

3.1 An Optimal Control Formulation

The challenging *Multi-Period Information Retrieval problem* is systematically expressed as follows:

"Suppose we have an information retrieval system receiving requests over a period of time from $t = 1$ *to* $t = T$*. For the same information need, there are many requests (impressions) over that period. Without loss of generality, we assume a request happens at time* $t \in \{1, ..., T\}$*, and the system responds with a ranked list of documents. The requester (user) then goes through the list and clicks documents which he or she thinks relevant. The problem is for the given information need how to determine an optimal ranking policy over time so as to maximize a utility (e.g. users satisfaction) of running the system over that period".*

Our study in this dissertation is a theoretical one; we offer a general solution framework by following an optimal control formulation [AF06]. As with any type of dynamically controlled system, in the case of our IR framework the rank action at time t is the control signal injected into the system [Oga01], where its state, in our case, denotes the system's belief about the documents' relevancy. Some form of sensor or user feedback (in our case clickthroughs) is required so that the system can adjust its state (its belief about the relevancy) to its changing environment. The rank action at a given time alters the system states by considering the system's dynamics (in this case the time evolution of the belief about the documents relevancy), the current utility and future potential utility.

Under this framework, three critical issues are addressed. Firstly, the system's state is represented as the posterior probability of a document being relevant, and we propose an iterative update mechanism

to update the posterior probability about the documents relevancy from users' feedbacks. Secondly, the system's dynamic function is derived, which shows how the belief (thus the posterior probability) evolves according to the rank decision made in the past and the user feedback (e.g. clicks) received so far. Lastly, equipped with the dynamic function, we develop simple ranking rules by integrating portfolio ranking with the recent results from multi-armed bandit machine research.

3.1.1 Formulation

We investigate a scenario for a given information need where there are $1 \leq i \leq N$ documents that need to be ranked. At each time t (or for each search of the given query), a fixed number M documents (where $0 \leq M \leq N$) are displayed to the user. The rank decision (control rule) is denoted as vector

$$\mathbf{a}(t) = \{a_1(t), ..., a_j(t), ..., a_M(t)\} \in \{1, ..., M\}^N \tag{3.1}$$

where $a_j(t) = i$ is the document displayed at rank j at time t. Those documents that are clicked are given a reward $x_j(t) = 1$ and 0 otherwise. Similarly, the user's responses over a ranked list is represented by vector

$$\mathbf{x}(t) = (x_1(t), ..., x_j(t), ..., x_M(t)) \in \{0, 1\}^M \tag{3.2}$$

where j is the index of the rank positions.

We further specify $R\big(\mathbf{x}(t), \mathbf{a}(t)\big)$ as our reward function at time t for a ranked list, which one can think of as the total number of clicks received from the ranked list.

Rather than directly using clicks to measure utility, we will introduce a utility function U, where the expected utility over a period of time is given as

$$o_T = E[U(\sum_{t=1}^{T} R(\mathbf{x}(t), \mathbf{a}(t)))] \tag{3.3}$$

and it should be emphasized that the expectation is over the clicks $\mathbf{x}(t)$ and the probability of a click, where $t \in [1, T]$, which will become clear in the next section. The optimal ranking is thus given by

$$(\mathbf{a}(1), ..., \mathbf{a}(T)^*) = \underset{\mathbf{a}(1),...,\mathbf{a}(T)}{\operatorname{argmax}} E[U(\sum_{t=1}^{T} R(\mathbf{x}(t), \mathbf{a}(t)))] \tag{3.4}$$

In this paper we adopt the exponential utility, one of the commonly used risk averse utilities, and assume the overall clicks follow a Gaussian distribution:

$$o_T = E[U(\sum_{t=1}^{T} R(\mathbf{x}(t), \mathbf{a}(t)))] \tag{3.5}$$

$$= E[1 - EXP(-\lambda \sum_{t=1}^{T} R(\mathbf{x}(t), \mathbf{a}(t)))] \tag{3.6}$$

$$= 1 - EXP(-\lambda E[\sum_{t=1}^{T} R(\mathbf{x}(t), \mathbf{a}(t))] + \frac{\lambda^2}{2} Var[\sum_{t=1}^{T} R(\mathbf{x}(t), \mathbf{a}(t))]) \tag{3.7}$$

where λ is a predefined utility parameter, used to adjust the risk preference. Thus, we reach our final

objective function:

$$(\mathbf{a}(1),...,\mathbf{a}(T)^*) = \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} E[U(\sum_{t=1}^{T} R(\mathbf{x}(t),\mathbf{a}(t)))] \tag{3.8}$$

$$= \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} (1 - EXP(-\lambda E[\sum_{t=1}^{T} R(\mathbf{x}(t),\mathbf{a}(t))] + \frac{\lambda^2}{2} Var[\sum_{t=1}^{T} R(\mathbf{x}(t),\mathbf{a}(t))])) \tag{3.9}$$

$$= \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} E[\sum_{t=1}^{T} R(\mathbf{x}(t),\mathbf{a}(t))] - \frac{\lambda}{2} Var[\sum_{t=1}^{T} R(\mathbf{x}(t),\mathbf{a}(t))] \tag{3.10}$$

$$= \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} \sum_{t=1}^{T} (E[R(\mathbf{x}(t),\mathbf{a}(t))] - \frac{\lambda}{2} Var[R(\mathbf{x}(t),\mathbf{a}(t))] \tag{3.11}$$

where we can see that the objective function has been decomposed into two parts; one for the expectation and the other for the variance. The last equation is due to the independent and identically distributed (i.i.d.) assumption about $\mathbf{x}(t)$, which is reasonable as for each t the system might receive independent clicks (note that the clicks over a ranked list may not be independent) from a random user. In the following two sections, we describe an algorithm to maximise the expectation assuming $\lambda = 0$. In Section 3.4, we extend it to consider the variance and thus the dependency of the clicks over different documents across the ranked list.

3.2 Iterative Expectation

$E\left[R\big(\mathbf{x}(t),\mathbf{a}(t)\big)\right]$ is the expected number of clicks for a given ranking $\mathbf{a}(t)$ at time t, which can be modelled and calculated using a *click model*. As outlined in Section 2.1.2.1, a click model describes the probability distribution associated with how a user typically browses a webpage and navigates search results, and can be used to infer document relevancies and predict clickthroughs. For a given document $D = d_{a_j(t)}$ we have observations over time about its clicks in different rank positions $\{C = x_{a_j(t)}, J = j\}_{k=1}^{t}$, where C is a binary variable representing a click, and J is the rank. The expected clicks over time can be obtained by:

$$\sum_{t=1}^{T} E[R(\mathbf{x}(t),\mathbf{a}(t))] = \sum_{t=1}^{T} \sum_{j=1}^{M} \sum_{x_j(t)} x_j(t) p(C = x_j(t) | D = d_{a_j(t)}, J = j) \tag{3.12}$$

$$= \sum_{t=1}^{T} \sum_{j=1}^{M} p(C = 1 | D = d_{a_j(t)}, J = j) \tag{3.13}$$

We, however, do not know whether a user clicking on a document was due to the fact it was relevant or simply because it has a high ranking position. Mathematically, we assume that the click is generated by a mixture of two binomial distributions, where a hidden binary variable is used to represent the membership, i.e. $S = 0$ denotes it is due to the rank bias and $S = 1$ due the document relevance. This

results in the following conditional probability of the click:

$$p(C = x_j(t)|D = d_{a_j(t)}, J = j) = p(x_j(t)|d_{a_j(t)}, S = 1, j)p(S = 1|j) + p(x_j(t)|j, S = 0)p(S = 0|j) \quad (3.14)$$

$$= p(x_j(t)|d_{a_j(t)})p(S_j = 1) + p(x_j(t)|S_j = 0)p(S_j = 0) \quad (3.15)$$

$$= (\mu_{a_j(t)})^{x_j(t)}(1 - \mu_{a_j(t)})^{1-x_j(t)}\pi_j + (\gamma_j)^{x_j(t)}(1 - \gamma_j)^{1-x_j(t)}(1 - \pi_j) \quad (3.16)$$

where j is the rank index and for simplicity we have defined $p(S|j) \equiv p(S_j)$. Here, three parameters have been defined:

$$p(C = 1|d_i) = \mu_i, \quad (3.17)$$

$$p(C = 1|S_j = 0) = \gamma_j \quad (3.18)$$

$$p(S_j = 1) = \pi_j \quad (3.19)$$

where i indicates the document index. A mixture click model has been evaluated against real click-through data in [CZTR08], and it should be emphasized that the mixture model presented here is a more general one. A heuristic background click model in [ABDR06] is in fact the case where $\pi_j = 1$ and γ_j is considered as the background click rate, and we shall see in Appendix B that other click models such as the Cascade model [Joa02] and the Examination Hypothesis model [CZTR08] are also special cases in the formulation by setting different types of parameters in Eq. (3.17, 3.18 & 3.19).

3.2.1 Expectation Maximization

We have the following Expectation Maximization (EM) algorithm to estimate the parameters [Bil97]:
E Step:

$$p(S_j|C) = \frac{p(C|S_j)p(S_j)}{p(C|S_j = 1)p(S_j = 1) + p(C|S_j = 0)p(S_j = 0)} \quad (3.20)$$

M Step:

$$\mu_{a_j(t)} = p(C = 1|d_{a_j(t)}) = \frac{\sum_{k=1}^{t} p(S_j = 1|C = x_j(k))x_j(k)}{\sum_{k=1}^{t} p(S_j = 1|C = x_j(k))} \quad (3.21)$$

$$\gamma_j = p(C = 1|S_j = 0) = \frac{\sum_{k=1}^{t} p(S_j = 0|C = x_j(k))x_j(k)}{\sum_{k=1}^{t} p(S_j = 0|C = x_j(k))} \quad (3.22)$$

$$\pi_j = p(S_j = 1) = \frac{1}{t}\sum_{k=1}^{t} p(S_j = 1|C = x_j(k)) \quad (3.23)$$

where the E step is obtained by applying Bayes' Rule.

3.2.1.1 The Calculation of the M Step

For a given document $D = d_{a_j(t)}$ we have observations over time about its clicks in different rank positions $\{C = x_j(t), J = j\}_{k=1}^{t}$. The M step can be obtained by maximizing the lower bound of the following likelihood function:

$$
\begin{aligned}
& L(\{\mu_{a_j(t)}\}, \{\pi_j\}, \{\gamma_j\}) \\
=& \prod_{t=1}^{T} p(C = x_j(t), J = j | D = d_{a_j(t)}) && (3.24)\\
=& \prod_{t=1}^{T} p(C = x_j(t) | D = d_{a_j(t)}, J = j) p(J = j | D = d_{a_j(t)}) && (3.25)\\
\propto& \prod_{t=1}^{T} p(C = x_j(t) | D = d_{a_j(t)}, J = j) && (3.26)\\
\propto& \sum_{t=1}^{T} \log p(C = x_j(t) | D = d_{a_j(t)}, J = j) && (3.27)\\
=& \sum_{t=1}^{T} \log \Big((\mu_{a_j(t)})^{x_j(t)} (1 - \mu_{a_j(t)})^{1 - x_j(t)} \pi_j + (\gamma_j)^{x_j(t)} (1 - \gamma_j)^{1 - x_j(t)} (1 - \pi_j) \Big) && (3.28)\\
\geq& \sum_{t=1}^{T} \Big(p(S_j = 1 | x_j(t)) \log \frac{(\mu_{a_j(t)})^{x_j(t)} (1 - \mu_{a_j(t)})^{1 - x_j(t)} \pi_j}{p(S_j = 1 | x_j(t))} \\
& \quad + p(S_j = 0 | x_j(t)) \log \frac{(\gamma_j)^{x_j(t)} (1 - \gamma_j)^{1 - x_j(t)} (1 - \pi_j)}{p(S_j = 0 | x_j(t))} \Big) && (3.29)\\
\propto& \sum_{t=1}^{T} \Big(p(S_j = 1 | x_j(t)) \log (\mu_{a_j(t)})^{x_j(t)} (1 - \mu_{a_j(t)})^{1 - x_j(t)} \pi_j \\
& \quad + p(S_j = 0 | x_j(t)) \log (\gamma_j)^{x_j(t)} (1 - \gamma_j)^{1 - x_j(t)} (1 - \pi_j) \Big) && (3.30)
\end{aligned}
$$

where the lower bound holds due to Jensen's inequality [Bil97]. Maximizing Eq. (3.30) with respect to the three parameters respectively obtains the M step. Furthermore, this algorithm requires iterative steps for each time t and we shall explore a simpler update procedure below.

3.2.1.2 Iterative EM

The EM algorithm can be further simplified by noticing that Eq. (3.22) and Eq. (3.23) are document independent and can be learned from observations of other documents as well. We thus define

$$\pi_j \equiv \pi, \ \gamma_j \equiv \eta^{j-1} \tag{3.31}$$

where $\eta \in (0, 1)$ and $\pi \in [0, 1]$ are predefined parameters. π is the prior information about the mixture, specifying the *trust* a user is giving to the search engine and is used to balance the influence of the rank bias and document relevancy. η is the probability that a user is going to examine the next document and click it regardless of its relevancy and represents the *rank bias*. To obtain Eq. (3.21), we fix the E step as:

$$
\begin{aligned}
\alpha_j(t) \equiv p(S_j = 1 | C = 1) =& \frac{p(C = 1 | S_j = 1) p(S_j = 1)}{p(C = 1 | S_j = 1) p(S_j = 1) + p(C = 1 | S_j = 0) p(S_j = 0)} && (3.32)\\
=& \frac{\hat{\mu}_{a_j(t)} \pi}{\hat{\mu}_{a_j(t)} \pi + \eta^{j-1} (1 - \pi)} && (3.33)
\end{aligned}
$$

and

$$\beta_j(t) \equiv p(S_j = 1|C = 0) = \frac{p(C = 0|S_j = 1)p(S_j = 1)}{p(C = 0|S_j = 1)p(S_j = 1) + p(C = 0|S_j = 0)p(S_j = 0)} \tag{3.34}$$

$$= \frac{(1 - \hat{\mu}_{a_j(t)})\pi}{(1 - \hat{\mu}_{a_j(t)})\pi + (1 - \eta^{j-1})(1 - \pi)} \tag{3.35}$$

where $\hat{\mu}_{a_j(t)}$ is the probability of click for document $a_j(t)$ estimated from past observations. $\alpha_j(t)$ and $\beta_j(t)$ are considered as "effective counts" and differentiate rank positions when receiving a click and non-click, respectively. α_j is an increasing function of j and is larger if receiving a click on a lower ranked document, rewarding the fact that a user makes efforts to reach that rank because it is unlikely that the click is due to the rank bias.

By contrast, if a document is clicked in the top rank, the effective count is rather small as the click might just be due to the rank bias, so by the same token we also notice that β_j is a decreasing function with respect to j i.e. for a higher ranked document, j is small and thus β_j is large. If such a high ranking document is not clicked, we effectively penalize the document more by adding large β_j in the denominator, and as the rank position increases, the penalty becomes small as well. In summary, a non-click on the document at high rank or a click on a low ranked document is an important observation and should give a large effective count when updating our belief about the probability of a click.

The probability of click is subsequently updated by:

$$\hat{\mu}^{new}_{a_j(t)} = p(C = 1|D = d_{a_j(t)}) = \frac{\sum_{k=1}^{t} (\alpha_j(k))^{x_j(k)} (\beta_j(k))^{1-x_j(k)} x_j(k)}{\sum_{k=1}^{t} (\alpha_j(k))^{x_j(k)} (\beta_j(k))^{1-x_j(k)}} \tag{3.36}$$

Which are iteratively obtained over time, giving us

$$\hat{\mu}_{a_j(t)} = \hat{\mu}_{a_j(t-1)} A_{a_j(t)}(t) + x_j(t)(1 - A_{a_j(t)}(t)) \tag{3.37}$$

where

$$A_{a_j(t)}(t) \equiv \frac{\sum_{k=1}^{t-1} (\alpha_j(k))^{x_j(k)} (\beta_j(k))^{1-x_j(k)}}{\Big(\sum_{k=1}^{t-1} (\alpha_j(k))^{x_j(k)} (\beta_j(k))^{1-x_j(k)}\Big) + (\alpha_j(k))^{x_j(t)} (\beta_j(k))^{1-x_j(t)}} \tag{3.38}$$

serves as an importance weight, balancing the influence between the previous observations and the current one. Eq. (3.37) provides the dynamical function of the probability of relevance, the dynamics of which are not only relevant to the control signal a, but also relevant to the random variable x.

To illustrate the flexibility of the proposed iterative update framework, we also provide an alternative scheme by plugging in the Examination Hypothesis click model introduced in [CZTR08] in the next section.

3.2.2 Plug in Examination Hypothesis

Another simplified model makes use of the *examination hypothesis* [CZTR08], whereby users not only have a certain probability of clicking on a document (based on its relevance), but also there is an independent probability associated with clicking on any document depending on its rank. Formally, we

would have the following assumption (instead of Eq. (3.31)):

$$\pi_j \equiv \eta^{j-1}, \qquad \gamma_j \equiv 1 \tag{3.39}$$

where $\eta \in [0, 1]$ indicates the rank bias (a cascade model assuming a linear traversal through the list) [Joa02], and $p(C = 1|S_j = 0) \equiv 1$ means that if the user decides not to examine the next document, there is no chance the document could be clicked. This model is considered as a partial mixture because the click event only occurs when event $S_j = 1$ happens. Combining the above parameters with Eq. (3.14) gives

$$p(C = 1|d_{a_j(t)}, j) = \mu_{a_j(t)}\eta^{j-1} \tag{3.40}$$

$$p(C = 0|d_{a_j(t)}, j) = (1 - \mu_{a_j(t)})\eta^{j-1} + (1 - \eta^{j-1}) \tag{3.41}$$

Subsequently, we obtain the simpler updates for α_j and β_j

$$\alpha_j(t) \equiv p(S_j = 1|C = 1) = \frac{p(C = 1|S_j = 1)p(S_j = 1)}{p(C = 1|S_j = 1)p(S_j = 1) + p(C = 1|S_j = 0)p(S_j = 0)} \tag{3.42}$$

$$= \frac{\mu_{a_j(t)}\eta^{j-1}}{\mu_{a_j(t)}\eta^{j-1} + 0(1 - \eta^{j-1})} \tag{3.43}$$

$$= 1$$

and

$$\beta_j(t) \equiv p(S_j = 1|C = 0) = \frac{p(C = 0|S_j = 1)p(S_j = 1)}{p(C = 0|S_j = 1)p(S_j = 1) + p(C = 0|S_j = 0)p(S_j = 0)} \tag{3.44}$$

$$= \frac{(1 - \mu_{a_j(t)})\eta^{j-1}}{(1 - \mu_{a_j(t)})\eta^{j-1} + (1 - 0)(1 - \eta^{j-1})} = \tag{3.45}$$

$$= \frac{(1 - \mu_{a_j(t)})}{(1/\eta^{j-1}) - \mu_{a_j(t)}} \tag{3.46}$$

Unlike the mixed click model, in this examination hypothesis model the effective count α_j for a click is 1, suggesting that the rank bias does not have any effect on a click update. It may seem counter-intuitive, but in fact it is true, as the click only happens when an examination event occurs (Eq. (3.40)). In other words, when a click is observed, the examine event is also observed. By contrast, the effective count for a non-click β_j is a decreasing functions of j just as before, taking into account the rank bias and penalizing less for a non-click of the later retrieved document.

3.3 Ranking Rule Over Time (Expectation Only)

Based on the previous discussion, our stochastic information retrieval problem can be restated as follows:

$$(\mathbf{a}(1)^*, ..., \mathbf{a}(T)^*) = \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} \sum_{t=1}^{T}\sum_{j=1}^{M} p(C = 1|d_{a_j(t)}, j) \tag{3.47}$$

$$= \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} \sum_{t=1}^{T}\sum_{j=1}^{M} \left(\hat{\mu}_{a_j(t)}\pi_j + \gamma_j(1 - \pi_j)\right) \tag{3.48}$$

The optimisation is subject to the constraint that $\hat{\mu}_{a_j(t)}$ follows the dynamic function regularized by Eq. (3.37); a simpler model, making use of the derivation from the previous section can be expressed as:

$$(\mathbf{a}(1)^*, ..., \mathbf{a}(T)^*) = \underset{(\mathbf{a}(1),...,\mathbf{a}(T))}{\operatorname{argmax}} \sum_{t=1}^{T} \sum_{j=1}^{M} \left(\hat{\mu}_{a_j(t)} \eta^{j-1} \right) \tag{3.49}$$

One might seek a dynamic programming solution [AF06] to find the optimal control rule, yet, current methods are still time-consuming and we leave improvements for future work. Instead an index solution (assign index scores for documents respectively and then rank them according to the scores) is proposed by following a multi-armed bandit machine approach.

3.3.1 Iterative Expectation (UCB-IE) Algorithm

The algorithm below is a variant of the popular UCB1 algorithm [ACBF02], outlined in Section 2.4, where we attempt to optimise over the values for $\hat{\mu}_{a_j(t)}$, which is updated using Eq. (3.37) and the number of plays is given by the effective count

$$B_{a_j(t)}(t) \equiv \sum_{k=1}^{t} (\alpha_j(k))^{x_j(k)} (\beta_j(k))^{1-x_j(k)} \tag{3.50}$$

(which is the denominator of $A_{a_j(t)}(t)$).

1. Set all $B_i(t) = 1$ (to avoid division by zero) and all $\hat{\mu}_i$ to some arbitrary number between 0 and 1 (such as 0.5). Also, set parameters π_j and γ_j.

2. Display each document to the user once, requiring $\lceil \frac{N}{M} \rceil$ time steps and record clicks as described in step 5.

3. For $t = \lceil \frac{N}{M} \rceil + 1$ time steps onwards, for all documents i, calculate

$$\Lambda_i = \hat{\mu}_i + \sqrt{\frac{2 \ln t}{B_i(t)}} \tag{3.51}$$

4. Set $\mathbf{a}(t)$ to be the M documents with the highest Λ_i values in decreasing order and display them to the user

5. Record clicks x_j and update variables accordingly, for $j = 1 \rightarrow M$

$$\alpha_j(t) = \frac{\hat{\mu}_{a_j(t)} \pi_j}{\hat{\mu}_{a_j(t)} \pi_j + \gamma_j (1 - \pi_j)} \tag{3.52}$$

$$\beta_j(t) = \frac{(1 - \hat{\mu}_{a_j(t)}) \pi_j}{(1 - \hat{\mu}_{a_j(t)}) \pi_j + (1 - \gamma_j)(1 - \pi_j)} \tag{3.53}$$

$$B_{a_j(t)}(t) = B_{a_j(t)}(t-1) + (\alpha_j(k))^{x_j(t)} (\beta_j(k))^{1-x_j(t)} \tag{3.54}$$

$$A_{a_j(t)}(t) = \frac{B_{a_j(t)}(t-1)}{B_{a_j(t)}(t)} \tag{3.55}$$

$$\hat{\mu}_{a_j(t)} = \hat{\mu}_{a_j(t-1)} A_{a_j(t)}(t) + x_j(t)(1 - A_{a_j(t)}(t)) \tag{3.56}$$

6. Repeat steps 3 to 6

The Mixed Clicks and Examination Hypothesis models can be incorporated into the above algorithm by setting the parameters π_j and γ_j accordingly:

Mixed Clicks $\pi_j \equiv \pi,\ \gamma_j \equiv \eta^{j-1}$

Examination Hypothesis $\pi_j \equiv \eta^{j-1},\ \gamma_j \equiv 1$

Using this algorithm, we attempt to learn the optimal $\hat{\mu}_{a_j(t)}$, representing the best documents to display to the user. Due to the rank bias, we adjust the reward given by a click to reflect the likelihood of the click given its position, and in a similar way use the effective count $B_i(t)$ rather than the number of impressions. Thus, documents that are displayed at lower ranks will have less 'impressions' than higher ranked documents, and so will still encourage exploration into the higher ranks.

3.4 Portfolio Ranking Rule Over Time

In the previous section, we considered the expected number of clicks $E\left[R(\mathbf{x}(t), \mathbf{a}(t))\right]$ and formulated click models that helped us develop an online algorithm that maximised this value. In this section, we will continue the development by considering the variance of a ranking from Eq. (3.11) ($\lambda > 0$), and use that to motivate the diversification of a ranking and how this can be achieved iteratively using our portfolio theory bandit. As explained in Section 2.2.1, it is beneficial to display documents from a range of topics rather than those considered most relevant .

Furthermore, the variance in Eq. (3.11) can be decomposed into

$$Var[R(\mathbf{x}(t), \mathbf{a}(t))] = \sum_{j=1}^{M} Var[a_j(t)]^2 + 2\sum_{j=1}^{M}\sum_{k=j+1}^{M} Var[a_j(t)]Var[a_k(t)]\rho_{i,j} \tag{3.57}$$

where $\rho_{i,j}$ is the correlation between i and j and $\rho \in [-1, 1]$. The correlation coefficient indicates the relationship between two documents, where a value of 1 implies a positive relationship, -1 a negative relationship and 0 that there is no relationship. In the context of ranked documents, the correlation allows us to introduce the idea of topics and diversity into the ranking problem.

Setting aside rank bias for simplicity, another additional factor that can affect clicks is whether a document belongs to the topic a user is interested in. Due to the ambiguity of some search queries, diversifying search results may lead to improved overall performance for all users, and this can be achieved using the above formulation as shown by [WZ09]. In this case, $\rho = 1$ would indicate that two documents belong to the same topic, and $\rho = -1$ otherwise.

To learn the correlations between documents iteratively, one can observe whether documents are clicked together when displayed in a ranking, and also those documents that aren't clicked when others are. Let $P_{i,j}$ be the number of *co-clicks* between documents i and j. Whenever two documents in a ranking are clicked together, then this value is incremented by 1, if a click occurs for document i, but not for document j, then this is incremented by -1, and if neither i and j are clicked, no change occurs. Similarly, $Q_{i,j}$ represents the number of times that two documents have been shown together, so if one or both documents i and j are clicked then this is incremented by 1. If neither document has been clicked, then we cannot discern any new information about the relationship between the documents, so we do not

update their variables. These rules are captured below:

$$P_{a_j(t),a_k(t)}(t+1) = P_{a_j(t),a_k(t)}(t) + x_j(t)x_k(t) - (x_j(t) - x_k(t))^2 \tag{3.58}$$

$$Q_{a_j(t),a_k(t)}(t+1) = Q_{a_j(t),a_k(t)}(t) + x_j(t) + x_k(t) - x_j(t)x_k(t) \tag{3.59}$$

Thus, the value of $\hat{\rho}_{i,j}(t) = \frac{P_{i,j}(t)}{Q_{i,j}(t)}$ represents our estimate of the correlation ρ between i and j.

In conventional portfolio theory ranking, we aim to minimize the variance of the overall ranking by choosing a diverse portfolio of documents, ones that are negatively correlated with one another. Conversely, with our UCB based approach, rather than subtract the variance as in Eq. (3.11), we instead add document variance so as to encourage exploration of documents that we know little about. We can then use a non-optimal but effective sequential approach to diversifying the ranking by first displaying the highest ranked document, then for each rank j choosing the document that maximises

$$\hat{\mu}_{a_j(t)} - \lambda \sum_{k=1}^{j-1} \hat{\rho}_{j,k}(t) \tag{3.60}$$

where, without loss of generality we've replaced $\frac{\lambda}{2}$ by λ, which acts as a balance between displaying the optimally relevant documents and diversifying the ranking.

3.4.1 Portfolio-armed Bandit (PAB) Algorithm

1. Set λ and display each document to the user once, requiring $\lceil \frac{N}{M} \rceil$ time steps and record clicks as described in step 4.

2. For $t = \lceil \frac{N}{M} \rceil + 1$ time steps onwards, for all documents i, calculate

$$\Lambda_i = \frac{X_i(t)}{Y_i(t)} + \sqrt{\frac{2 \ln t}{Y_i(t)}} \tag{3.61}$$

3. Calculate ranking $\mathbf{a}(t)$ and display it to the user, where

   $a_1(t) = \max_i \Lambda_i$

   for $j = 2 \to M$ **do**

   $\Gamma_i = \Lambda_i - \lambda \sum_{k=1}^{j-1} \hat{\rho}_{i,a_k(t)}(t)$

   where $\hat{\rho}_{i,j}(t) = \begin{cases} \frac{P_{i,j}(t)}{Q_{i,j}(t)} & \text{if } Q_{i,j}(t) > 0 \\ -1 - \sqrt{\frac{2 \ln t}{Y_i(t)}} & \text{otherwise} \end{cases}$

   $a_j(t) = \max_{i \neq a_{1..j-1}(t)} \Gamma_i$

   end for

4. Record clicks $\mathbf{x}$ and update variables accordingly, for $j = 1 \to M, k = 1 \to M$

$$Y_{a_j(t)}(t+1) = Y_{a_j(t)}(t) + 1 \tag{3.62}$$

$$X_{a_j(t)}(t+1) = X_{a_j(t)}(t) + x_j(t) \tag{3.63}$$

$$P_{a_j(t),a_k(t)}(t+1) = P_{a_j(t),a_k(t)}(t) + x_j(t)x_k(t) - (x_j(t) - x_k(t))^2 \tag{3.64}$$

$$Q_{a_j(t),a_k(t)}(t+1) = Q_{a_j(t),a_k(t)}(t) + x_j(t) + x_k(t) - x_j(t)x_k(t) \tag{3.65}$$

5. Repeat steps 2 to 5

Λ_i is taken from the UCB algorithm, which handles the exploration and exploitation of the documents. X_i is the number of clicks that document i has received so far (or $\sum_{t=1}^{T} \sum_{j=1}^{M} I\{x_j(t), a_j(t) = i\}$), and Y_i the number of impressions, so $\frac{X_i}{Y_i}$ is the clickthrough rate and our estimation of its relevance $\hat{\mu}_i$.

When $Q_{i,j} = 0$, we have yet to learn anything about the correlation between documents i and j, so we set their correlation to $-1 - \sqrt{\frac{2 \ln t}{Y_i(t)}}$, this correlation being less than -1. This will encourage exploration of such document pairs rather than diversifying, and subtracting the variance $\sqrt{\frac{2 \ln t}{Y_i(t)}}$ will ensure that those documents that are being explored by the UCB part of the algorithm will have a higher chance of being shown.

Chapter 4

Experiments

In the previous chapter we studied the properties of document ranking over time, under the assumption of certain click models. This chapter continues the study by evaluating the resulting practical ranking strategies through simulations. We mainly intend to 1) understand the optimality of the proposed algorithms; 2) study the impact of parameters; and 3) analyse the robustness against noise and changing environments. Three resulting ranking strategies were evaluated. Namely, we have the UCB algorithm with Interactive Expectation from Mixed-Click model (denoted as UCB-IE-MC) and that from Examination Hypothesis (denoted as UCB-IE-EH), which are intended to deal with rank bias. To address the diversity and dependency of the clicks, we have the Portfolio-Armed Bandit algorithm (denoted as PAB). Note that they can be naturally combined in practice, however, to make our evaluation targeted, we do not combine them here and have separated the experiments into two scenarios: one for rank bias and the other for diversity.

4.1 Simulation Setup

To verify the algorithms effectiveness, it would be difficult to test on real data without access to a search engine, but simulations could be run to verify the theoretical limitations. Our simulated models were designed to be reflective of reality, but operate under the assumptions of our problem formulation and allow variations so as to assess the robustness of the algorithms.

We tested the impact of rank bias with an experiment on a set of click models proposed in [CZTR08]. We believe those click models reflect the reality well as they have been thoroughly studied and analysed using real clickthrough data. Specifically, 50 documents were randomly assigned a uniformly distributed probability of relevance $P(r_i)$, representing the probability that any user would click on the document i. We assessed each algorithm using three different click models:

1. The mixed click model described in Section 3.2, with parameters $\pi = 0.8$, $\eta = 0.8$.
2. The examination hypothesis model with logarithmic discount curve $P(E_j = 1) = \frac{1}{1+\log(j)}$
3. Another examination hypothesis model with a parabolic discount curve $P(E_j = 1) = 1 - \frac{1}{(jM)^2}$

Using these models, the user viewed all $M = 10$ documents and for the second and third models, a click occured on any displayed document with probability $p(r_i)p(E_j = 1)$.

The impact of co-clicks on diversity are evaluated by extending the simulation detailed in [RKJ08]. A Chinese Restaurant Process (with $\gamma = 3$) is used to assign 20 users to different topics, giving an average of 6.5 unique topics. Then the 50 documents are assigned to the topics in the same proportion as users assigned to topics. At each iteration, $M = 5$ documents are selected and shown to a randomly chosen user. If the user sees a document that belongs to the same topic as they are, they will click on it with probability p_R, otherwise with probability p_{NR}. Note that there is no rank bias or underlying document relevance. For each experiment, the clickthrough rate was found to determine its success. This is defined as the number of rankings that displayed at least one relevant document (1 - abandonment) over t (abandonment occurs when no documents are clicked).

Note that all experiments in the two scenarios are averaged results over 100 iterations and allowing either 100,000 or 400,000 time steps.

The experiments were created using Matlab and executed on both a personal computer (dual core 2.4 GHz with 4GB RAM) and a server (8 core 2.5GHz with 8GB RAM). Matlab was chosen as it allowed quick prototyping and experimentation of the algorithms whilst they were being developed, and offered a powerful results visualisation toolset. Additionally, its powerful matrix programming language lent itself well to the mathematics of the algorithms. An attempt was made to incorporate objects into the experiment code in order to make it more robust, but this significantly increased the running time and so was abandoned.

4.2 Rank Bias

To provide a baseline from which to compare the performance of each algorithm, a UCB-based multi-play MAB algorithm was also compared [LGJP07]. This algorithm is identical to the original UCB algorithm, with the exception that the top M scoring arms are played rather than a single arm, and differs from our algorithms in that it ignores rank position when allocating rewards.

4.2.1 Metrics

In the bandit machine literature, the overall regret from the optimal decision (with perfect information) is a commonly used metric and can be extended to assess the ability of the algorithm to find the top M relevant documents, which is given by

$$\text{Regret}(T) = \sum_{t=1}^{T} \sum_{j=1}^{M} (\mu_{a_j^*} - \mu_{a_j(t)}) \tag{4.1}$$

where $\mathbf{a}^*$ is the optimal ranking. This metric measures the cumulative loss of the chosen ranking at each time step against the optimal ranking, and it is our aim to reduce this value over time logarithmically.

Unfortunately, this metric fails to take into account the ordering of a ranking, so to further capture the ability to then correctly rank the documents, the *Overall Discounted Cumulative Gain Regret* (oDCG) can be calculated, given by

$$\text{oDCGRegret}(T) \equiv \sum_{t=1}^{T} \text{DCGRegret}(t) = \sum_{t=1}^{T} (\text{DCG}(\mathbf{a}^*) - \text{DCG}(\mathbf{a}(t))) \tag{4.2}$$

Table 4.1: Performance of each algorithm with each click model

Model	1	2	3
			Regret $\times 10^6$
Baseline	0.710	3.342	16.74
UCB-IE-MC	0.566	2.751	15.70
UCB-IE-EH	8.444	2.450	4.800
			nDCGR$\times 10^{-3}$
Baseline	2.141	16.76	31.84
UCB-IE-MC	1.040	9.818	28.94
UCB-IE-EH	38.37	18.49	18.99

where $\mathrm{DCG}(\mathbf{a}) \equiv \sum_{j=1}^{M} \frac{\mu_{a_j(t)}}{\log_2(j+1)}$. This equation measures the Discounted Cumulative Gain (DCG) of the optimal ranking $\mathbf{a}^*$ against that chosen by the algorithm, a measure that is minimised when the most relevant documents are ranked in the correct order. The formula for the DCG can be considered similar to Eq. (3.13), where we take $1/\log_2(j+1)$ to be the rank bias. We can then normalise this value as follows:

$$\mathrm{nDCGR}(T) \equiv \frac{1}{T} \sum_{t=1}^{T} \left(1 - \frac{\mathrm{DCG}(\mathbf{a}(t))}{\mathrm{DCG}(\mathbf{a}^*)}\right) \tag{4.3}$$

allowing us to compare the results of different algorithms using different click models; the closer that this value is to 0, the better the performance

4.2.2 Click Model Investigation

In order to provide some preliminary analysis, each algorithm was tested with each of the user browsing models using the Regret, DCGRegret and nDCGR scoring metrics, the results of which can be seen in Table 4.1 and Figures 4.1 and 4.2. In comparison to the baseline, the UCB-IE-MC algorithm performed consistently better, having both a statistically significant (with a p-value of less than 0.01 on the Wilcoxon signed-rank test) lower Regret and nDCGR. On the other hand, the UCL-IE-EH algorithm proved less consistent, and so was abandoned from future experiments. In addition, no algorithm seemed to respond well to the parabolic discount curve, and less so to the logarithmic model, and so future experiments were completed using model 1.

A secondary purpose of this experiment was also to assess the validity of the metrics and determine if the DCGRegret (and subsequently nDCG) did discriminate between the ranking and non-ranking algorithms. Figures 4.1 and 4.2 show respectively the performance of the algorithms using the Regret and the DCGRegret. Particularly between the Baseline and UCB-IE-MC algorithms it can be seen that the DCGRegret provided a greater distinction, penalising the Baseline algorithm for not learning a correct ordering of documents. This relationship can also be seen in the nDCGR in Table 4.1 and so we chose this normalised metric for future experiments.

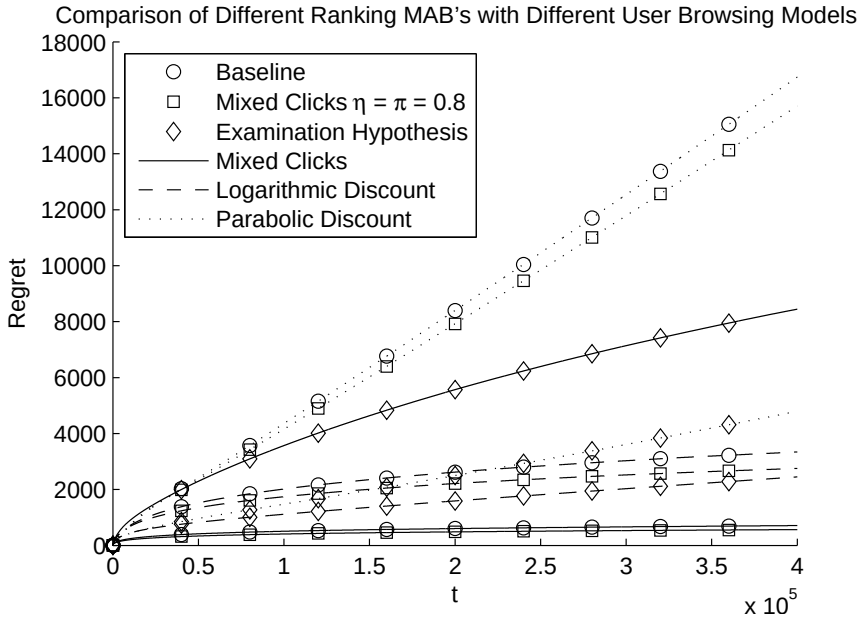

Figure 4.1: The performance of each algorithm with each model measured using regret

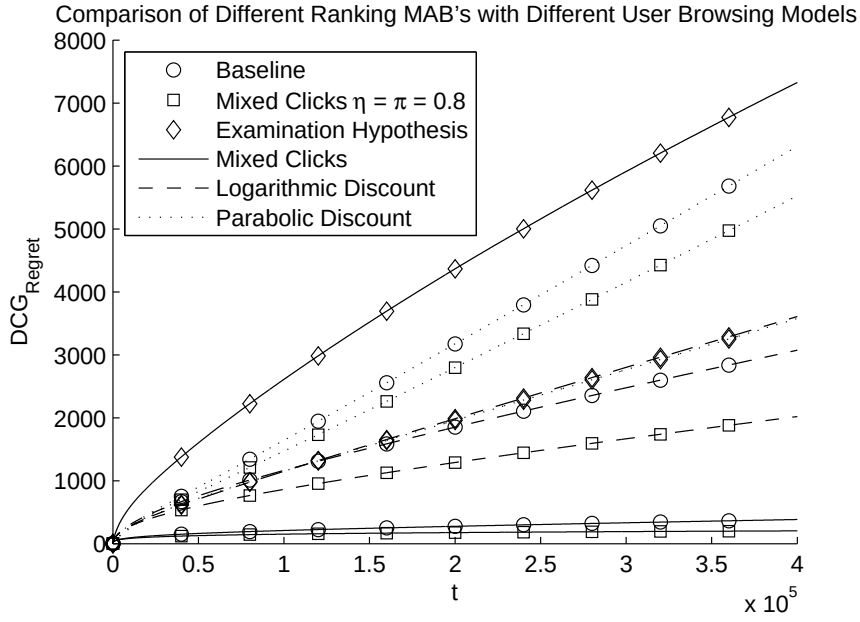

Figure 4.2: The performance of each algorithm with each model measured using DCGRegret

Table 4.2: Effect of parameter changes on UCB-IE-MC (values are nDCGR$\times 10^{-3}$)

Parameter	0.4	0.5	0.6	0.7	0.8	0.9
π	31.18	16.33	5.937	1.410	1.067	1.419
η	2.400	2.067	1.636	1.196	1.029	1.2957

4.2.3 Parameter Sensitivity Experiment

Next, we study the sensitivity of the parameters. A drawback of the UCB-IE-MC algorithm is that it requires knowing its parameters η and π a priori, which practically would have to be learnt from offline clickthrough logs. For this experiment we adjusted the parameters against the parameters of the model to observe how disruptive inaccurate parameter values were. Table 4.2 shows the effect of varying π and η from their true values. It can be seen that the algorithm is very sensitive to changes in π, whereas performance degrades gracefully when η is adjusted from its ideal value. Thus, it is more important to accurately learn the parameter π than η before making use of UCB-IE-MC, although in practice it is easier to observe the rank bias η than the latent probability of trust in a search engine that π represents.

4.2.4 A Priori Information Experiment

We continue our experiment by studying the impact of introducing new documents on the overall performance of the algorithm. For many IR systems, in particular search engines, new documents are continually being added (and other documents removed), and as such any ranking algorithm should be able to adjust to the addition of new documents. In our case, introducing new documents should cause the UCB-IE-MC to dynamically place more emphasis on exploration in order to learn more about the new documents, before settling into a new optimal ranking that reflects the new information.

We repeat the experiment as before up to 100,000 iterations, before introducing varying amounts of new documents (with subsequent unknown probabilities of relevance) and recording performance up to 500,000 iterations. Figure 4.3 shows that the introduction of the new documents causes a sharp degradation in performance (during the explorative phase), before gradually starting to converge once more to an optimal ranking, and that increasing the number of new documents affects performance gracefully.

It is worth noting that the worst case scenario for this experiment (here adding 50 additional documents) effectively doubled the number of documents that needed to be ranked. It is unlikely that in a real IR system such a large proportion of new documents would be added at once, yet we observe that optimal convergence is still achieved even with such a large disturbance.

4.2.5 Relevance Change Experiment

Lastly, we investigate the algorithm's responsiveness to change in its environment. A key motivator for the work in this dissertation was the inability of current ranking algorithms to be responsive to changes in document relevancy, and so we test this directly here. Due to the fact that the algorithm always performs some exploration (although less so as time progresses), any changes to the underlying

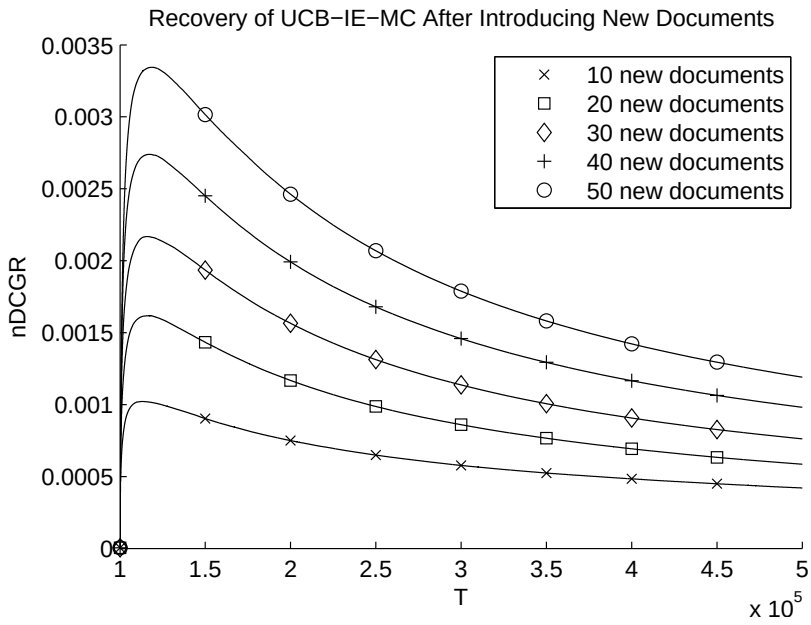

Figure 4.3: The effect of introducing new documents, the first 100,000 time steps have been removed from the graph

document relevancies should eventually be recognised and adjusted and the regret should converge once more.

Similarly to the last experiment, we ran our algorithm over 100,000 iterations, at which point, we randomly changed the relevance of a proportion of the documents and continued to run the experiment as normal. Figure 4.4 shows that the algorithm is able to recover from changes in its environment, the larger the change the slower the convergence, although once again the performance degrades gracefully.

It is worth considering that UCB1 based MABs are generally designed for *well-behaved* distributions i.e. those that do not change too much over time. In an environment where large changes in relevance can occur often, then an adversarial bandit such as Exp3 or a restless bandit algorithm may be considered more appropriate [KRS10, UNK10].

4.3 Dependency and Co-clicks

In this section, we shift our focus to the situation where the dependency of the clicks is modelled. For that, the click model has been updated to include different *topics*, modelling users/documents diversified interests, which has been outlined fully in Section 4.1.

The performance of the PAB was compared to the UCB1-Ranked Bandits Variant (UCB1-RBV), which was shown by [RKJ08] experimentally to perform better than their Exp3 based algorithm, but does not hold the same theoretical guarantees. Also, in their original experiment, only one document at most was clicked for each ranking, but as our algorithm is able to handle the click dependency through the ranked list, multiple clicks have been allowed in this experiment; although when running the UCB1-RBV algorithm only the first click is regarded (as multiple clicks caused a degradation in its performance).

In addition, similarly to [RKJ08] the performance will be bounded above by the value OPT, rep-

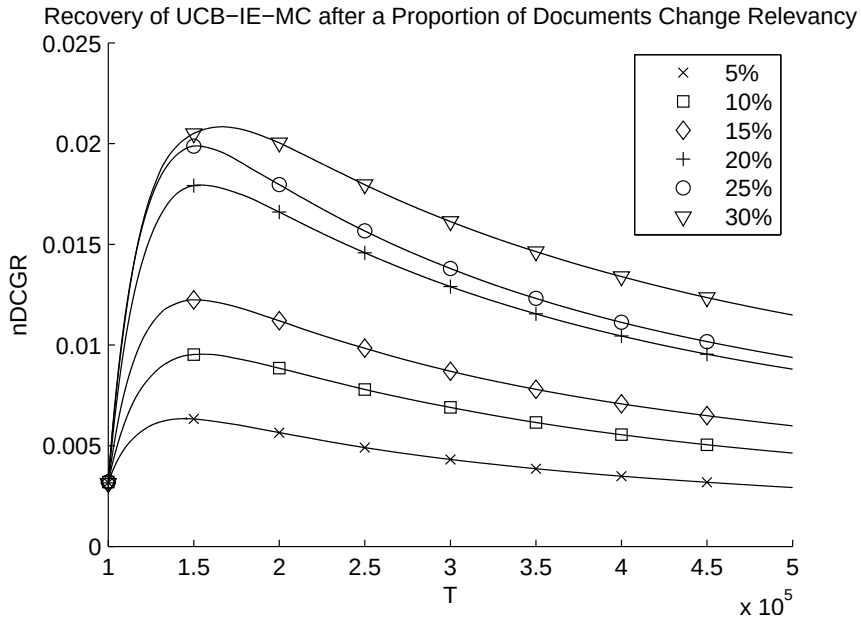

Figure 4.4: The effect of changing the relevance of a proportion of documents, the first 100,000 time steps have been removed from the graph

resenting the highest clickthrough rate possible (where each displayed document belongs to a different topic from the most popular topics), and below by $(1 - \frac{1}{e})OPT$, being the worst case scenario bound.

4.3.1 The Impact of the Parameter λ

We first examine the impact of the parameter λ. The λ value in the PAB formulation provides a balance between exploration/exploitation and diversification, and an optimal value may be learnt from offline data or using existing knowledge of the diversity of a query. In this experiment we varied the value of λ in order to find an optimal value to use in later experiments, and set the values of p_R and p_{NR} to 1 and 0 respectively (so as to observe the effect of λ with no noise), the results of which can be seen in Figure 4.5.

The graph shows that the algorithm is able to correctly find a diverse ranking of documents, whereby users interested in different topics are able to at least find one document in the ranking that is relevant to them. The poor results for $\lambda = 0$ indicate that the diversification is responsible for the excellent performance of the algorithm, particularly when compared to UCB1-RBV for all non-zero values of λ. There is also a trend towards better improvement with higher values of λ, culminating in near optimal convergence when $\lambda = 1$. As such, for the next experiment λ was set to 1.

4.3.2 Resistance to Noise

Finally, we test whether the algorithm is resistant to random noise (in particular its correlation matrix), and so the values for p_R and p_{NR} were adjusted accordingly to allow clicks on non-relevant topics. Rather than measure the now distorted clickthrough rate, we instead evaluate the performance by using the fraction of rankings with a relevant document, which is the expected clickthrough rate if p_R were set to 1. It is vital that the algorithm is robust to noise as real users will be unlikely to only click on

documents that are relevant to their topic of interest, and may also miss documents that are.

Figure 4.6 shows that as the level of noise increases, the performance of the PAB degrades gracefully and still outperforms UCB1-RBV.

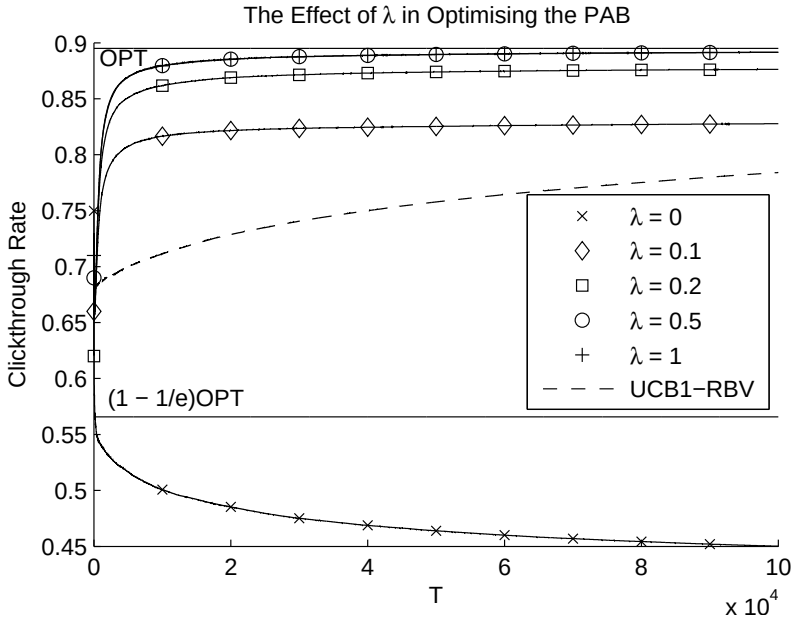

Figure 4.5: The effect that λ has on the performance of the PAB algorithm

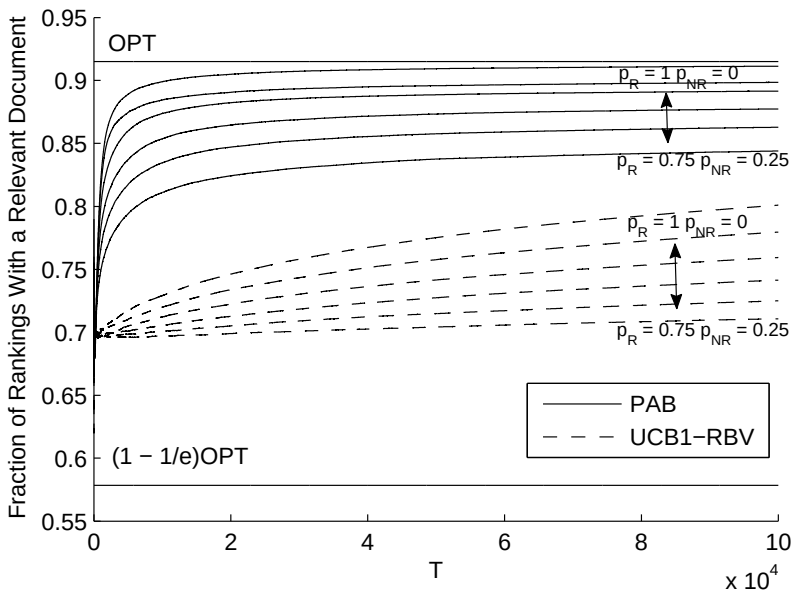

Figure 4.6: Decreasing p_R to determine how noise affects the performance of the algorithms

Chapter 5

Conclusion and Discussion

In this chapter, we summarize our work and the results obtained in our experiments, and show how they meet the objectives of this dissertation. We also make note of the limitations of our theoretical model with suggestions for improvements. Then we discuss the future directions that this research could take, including the combination of the two algorithms and the need to continue the experiments using real data. Furthermore, variations of the formulation are considered and applied to a broader research topic. We end by examining how our work could be used in the field of finance.

5.1 Summary

We have presented a probabilistic optimisation framework for stochastic information retrieval, where unlike the static IR problem; here we have opportunities to learn the relevance of documents over a period of time through the interaction with the users (by considering clicks as observations of relevance). Unlike previous studies, we consider the following assumption about the users' feedback: the probability of a viewer examining and clicking a document is dependent on its ranking position, as well as the documents ranked before it. Through the theoretical derivation and analysis, we showed how the belief about the relevancy of documents (the posterior probability of being clicked) evolves over time from the rank-biased user feedback (clicks). Borrowing the idea from the portfolio theory of information retrieval, we also explained how the dependency of clicks should be handled in order to maximise the expected end utility.

Besides the theoretical understandings and insights, two practical stochastic ranking strategies have been derived on the basis of the bandit machine theory, and the significance of the ranking strategies has been properly evaluated through simulations. Our optimisation framework allowed us to plug in different click models and evaluate them experimentally, and we found a certain degree of robustness when choosing a model different from the underlying model. We were also able to demonstrate the effectiveness of our normalized DCG metric in evaluating ranked lists. Next we demonstrated the responsiveness of our algorithm to the addition of new documents and the changing relevancies of existing documents, fulfilling the criteria set out in our problem motivation.

Finally, we experimented on the portfolio-armed bandit, first learning an appropriate value for the tuning parameter λ, and then showing a much-improved performance over the UCB1-RBV baseline,

even in the presence of noise, and demonstrating the ability of our algorithm to learn a diverse ranking of documents dynamically.

5.1.1 Drawbacks

While helping us get an insight to the optimal document ranking over time, the probabilistic framework has a number of strong assumptions that may not necessarily correspond with established observed user browsing practice. For example, we fix the number of documents (M) that has been examined. Although typically a subset of documents is shown to a user who has performed a search (normally the documents in the first page), the user often does not view all displayed documents, or can request to view more documents. In this regard, a possible next step would be to allow M to be a random variable that models the likelihood of a user viewing documents; a Poisson model may be appropriate in this case.

In addition, whilst we have endeavoured to reflect reality in the design of our simulations and demonstrated the theoretical guarantees, the experiments are nonetheless subsequently biased and so it is vital that we perform similar experiments using actual data, which will be further explored in the next section.

5.2 Future Work

Having demonstrated the effectiveness of the algorithms separately, the next step is to unify the expectation and variance of the multi-period IR model into one dynamic algorithm that takes into account iterative expectation and ranking diversity simultaneously. Already this could be achieved by replacing $Y_i(t)$ from Eq. (3.62) in the PAB algorithm with $B_i(t)$ from Eq. (3.50) in the UCB-IE algorithm, and likewise $\frac{X_i(t)}{Y_i(t)}$ from Eq. (3.61) with $\hat{\mu}_i(t)$ from Eq. (3.56), and updating all variables accordingly. Nonetheless, an appropriate click model will need to be formulated that takes into account document relevancy, topic relevancy and rank bias, as well as a suitable metric that rewards both diversity and correct ranking, which is itself an ongoing research topic [ZCL03, AGHI09].

Furthermore, the algorithm will need to be tested using real search data, the acquirement of which for dynamic algorithms is also an active area of research [LCLW11, MLC$^+$10]. An ideal collection of data would be one in which a number of different users were presented with every document in a collection and clicked on *all* documents they considered relevant. In this way, we could simulate a live environment by showing a ranked subset of the documents to each 'user' and allowing them to 'click' on those documents they deemed relevant. We are currently considering the viability of data from the Exploitation and Exploration challenge [1].

It would also be beneficial to explore additional click models using our framework, both to test its robustness further, and to experiment with more complex, realistic click models, such as the general click model [ZCM$^+$10]. The probability of a document being clicked is likely to change over time and we have shown that our algorithms are able to handle that change; to expand further, a restless bandits (bandits whose distributions change over time) formulation could also be explored [Whi88].

The dynamic optimisation framework created in this dissertation is also relevant to the direction the

[1] `http://explo.cs.ucl.ac.uk/`

writer of this dissertation intends to take their research in pursuance of their PhD, that of understanding the stochastic processes underlying the relevancies of documents. In addition, this will be preceded by an investigation into how financiers model the stochastic processes that drive market prices, continuing the collaboration between finance and IR.

5.2.1 Applications to Finance

In the portfolio theory of IR [WZ09], we take portfolio theory from finance and analogise investing in assets as choosing documents to display in a ranking. Having extended the theory to be incorporated into the multi-armed bandit setting, we can reverse this analogy and apply it to the problem of asset allocation. Given a collection of assets over multiple periods, which do we invest in?

[Sam69, FS82] were able to demonstrate that stochastic portfolio theory is a continuous, dynamic process that needed to be optimised over time, and bears similarities with our IR based work. A difference between theirs and our work is the exploration aspect that increases the risk; allowing investments in assets that are perhaps unknown, whilst balancing this with a diverse portfolio that reduces risk. Whilst it is unreasonable to assume that an investor would be willing to explore as much as our algorithm is designed to, or operate over a similar period of time steps, the parameters can be tuned to reflect such preferences.

Bibliography

[ABD06] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 19–26. ACM, 2006.

[ABDR06] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 3–10, 2006.

[ACBF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47:235–256, May 2002.

[ACBFS95] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, pages 322–. IEEE Computer Society, 1995.

[AF06] Michael Athans and Peter L. Falb. *Optimal Control: An Introduction to the Theory and Its Applications*. Dover Books on Engineering Series. Dover Publications, 2006.

[AGHI09] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14. ACM, 2009.

[AGI07] Nikolay Archak, Anindya Ghose, and Panagiotis G. Ipeirotis. Show me the money!: Deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 56–65. ACM, 2007.

[AVW86] Venkat Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays part I: I.I.D. rewards, part II: Markovian rewards. Technical Report UCB/ERL M86/62, EECS Department, University of California, Berkeley, 1986.

[Azz11] Leif Azzopardi. The economics in interactive information retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, pages 15–24. ACM, 2011.

[BBF+03] Peter Bank, Fabrice Baudoin, Hans Föllmer, L. C. G. Rogers, Mete Soner, and Nizar Touzi. American options, multi-armed bandits, and optimal consumption plans: A unifying view. In *Paris-Princeton Lectures on Mathematical Finance 2002*, volume 1814 of *Lecture Notes in Mathematics*, pages 1–42. Springer Berlin / Heidelberg, 2003.

[BH05] Dirk Bergemann and Ulrigh Hege. The financing of innovation: Learning and stopping. *RAND Journal of Economics*, 36(4):719–752, Winter 2005.

[Bil97] Jeff A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. In *International Conference on Systems Integration*, 1997.

[BV96] Dirk Bergemann and Juuso Valimaki. Learning and strategic pricing. Cowles Foundation Discussion Papers 1113, January 1996.

[CG98] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336. ACM, 1998.

[CK06] Harr Chen and David R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 429–436. ACM, 2006.

[CKRU08] Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Neural Information Processing Systems*, pages 273–280, 2008.

[CT08] Kevyn B. Collins-Thompson. *Robust Model Estimation Methods for Information Retrieval*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, December 2008.

[CZTR08] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 87–94. ACM, 2008.

[DGST09] Louis Dorard, Dorota Glowacka, and John Shawe-Taylor. Gaussian process modelling of dependencies in multi-armed bandit problems. *Proceedings of the 10th International Symposium on Operational Research SOR09*, pages 721–728, 2009.

[FKM+05] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23:147–168, April 2005.

[Fra11] Massimo Franceschet. Pagerank: Standing on the shoulders of giants. *Commun. ACM*, 54:92–101, June 2011.

[FS82] Robert Fernholz and Brian Shay. Stochastic portfolio theory and stock market equilibrium. *Journal of Finance*, 37(2):615–24, May 1982.

[Fuh08] Norbert Fuhr. A probability ranking principle for interactive information retrieval. *Inf. Retr.*, 11(3):251–265, 2008.

[GIS07] Anindya Ghose, Panagiotis G. Ipeirotis, and Arun Sundararajan. Opinion mining using econometrics: A case study on reputation systems. In *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

[Git79] John C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B Methodological*, 41(2):148–177, 1979.

[GLK+09] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 11–20. ACM, 2009.

[GLW09] Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 124–131. ACM, 2009.

[GMS07] Sudipto Guha, Kamesh Munagala, and Peng Shi. Approximation algorithms for restless bandit problems. *CoRR*, abs/0711.3861, 2007.

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 154–161, 2005.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142. ACM, 2002.

[Jov79] Boyan Jovanovic. Job matching and the theory of turnover. *Journal of Political Economy*, 87(5):972–90, October 1979.

[KRS10] Satyen Kale, Lev Reyzin, and Robert Schapire. Non-Stochastic Bandit Slate Problems. In *Advances in Neural Information Processing Systems 23*, pages 1045–1053. 2010.

[LCLW11] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 297–306. ACM, 2011.

[LGI11] Beibei Li, Anindya Ghose, and Panagiotis G. Ipeirotis. Towards a theory model for product search. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 327–336. ACM, 2011.

[LGJP07] Lifeng Lai, Hesham El Gamal, Hai Jiang, and H. Vincent Poor. Cognitive medium access: Exploration, exploitation and competition. *CoRR*, abs/0710.1385, 2007.

[LR85] Tze L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[Mar57] Harry Markowitz. A simplex method for the portfolio selection problem. Technical report, 1957.

[MLC+10] Taesup Moon, Lihong Li, Wei Chu, Ciya Liao, Zhaohui Zheng, and Yi Chang. Online learning for recency search ranking using real-time user feedback. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1501–1504. ACM, 2010.

[Oga01] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall PTR, 4th edition, 2001.

[RKJ08] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 784–791. ACM, 2008.

[RL03] Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, 18:95–145, June 2003.

[Rob77] Stephen E. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

[Rot74] Michael Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2):185–202, October 1974.

[RZ09] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389, April 2009.

[Sam69] Paul A Samuelson. Lifetime portfolio selection by dynamic stochastic programming. *The Review of Economics and Statistics*, 51(3):239–46, August 1969.

[Sor07] Morten Sorensen. Learning by investing: Evidence from venture capital. SIFR Research Report Series 53, May 2007.

[SSBT08] Falk Scholer, Milad Shokouhi, Bodo Billerbeck, and Andrew Turpin. Using clicks as implicit judgments: expectations versus observations. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 28–39. Springer-Verlag, 2008.

[UNK10] Taishi Uchiya, Atsuyoshi Nakamura, and Mineichi Kudo. Algorithms for adversarial bandit problems with multiple plays. In *Proceedings of the 21st international conference on Algorithmic learning theory*, ALT'10, pages 375–389, Berlin, Heidelberg, 2010. Springer-Verlag.

[Whi88] Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.

[WZ09] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *In SIGIR' 09: Proc. 32nd Int. ACM SIGIR Conf. on Research and Development in IR*, pages 115–122. ACM, 2009.

[YL11] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 177–186. ACM, 2011.

[ZCL03] Cheng Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of SIGIR*, pages 10–17, 2003.

[ZCM+10] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu, and Zheng Chen. A novel click model and its applications to online advertising. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 321–330. ACM, 2010.

[ZGVA07] Xiaojin Zhu, Andrew B. Goldberg, Jurgen Van, and Gael David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Physics Laboratory University of Washington*, pages 97–104, 2007.

[Zha08] Cheng Xiang Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, 2008.

[ZHL+06] Qiankun Zhao, Steven C. H. Hoi, Tie-Yan Liu, Sourav S. Bhowmick, Michael R. Lyu, and Wei-Ying Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 543–552. ACM, 2006.

[ZJ07] Wei Vivian Zhang and Rosie Jones. Comparing click logs and editorial labels for training query rewriting. WWW '07, 2007.