Foundations and Trends® in Information Retrieval
Vol. 10, No. 1 (2016) 1–117

DOI: 10.1561/1500000051

now
the essence of knowledge

Online Evaluation for Information Retrieval

Katja Hofmann
Microsoft
katja.hofmann@microsoft.com

Lihong Li
Microsoft
lihongli@microsoft.com

Filip Radlinski
Microsoft
filip.radlinski@microsoft.com

Contents

Abstract

Online evaluation is one of the most common approaches to measure the effectiveness of an information retrieval system. It involves fielding the information retrieval system to real users, and observing these users' interactions in-situ while they engage with the system. This allows actual users with real world information needs to play an important part in assessing retrieval quality. As such, online evaluation complements the common alternative offline evaluation approaches which may provide more easily interpretable outcomes, yet are often less realistic when measuring of quality and actual user experience.

In this survey, we provide an overview of online evaluation techniques for information retrieval. We show how online evaluation is used for controlled experiments, segmenting them into experiment designs that allow absolute or relative quality assessments. Our presentation of different metrics further partitions online evaluation based on different sized experimental units commonly of interest: documents, lists and sessions. Additionally, we include an extensive discussion of recent work on data re-use, and experiment estimation based on historical data.

A substantial part of this work focuses on practical issues: How to run evaluations in practice, how to select experimental parameters, how to take into account ethical considerations inherent in online evaluations, and limitations. While most published work on online experimentation today is at large scale in systems with millions of users, we also emphasize that the same techniques can be applied at small scale. To this end, we emphasize recent work that makes it easier to use at smaller scales and encourage studying real-world information seeking in a wide range of scenarios. Finally, we present a summary of the most recent work in the area, and describe open problems, as well as postulating future directions.

K. Hofmann, L. Li, and F. Radlinski. *Online Evaluation for Information Retrieval.* Foundations and Trends® in Information Retrieval, vol. 10, no. 1, pp. 1–117, 2016. DOI: 10.1561/1500000051.

1

Introduction

Information retrieval (IR) has a long and fruitful tradition of empirical research. Since early experiments on indexing schemes, and the development of the Cranfield paradigm, researchers have been striving to establish methodology for empirical research that best supports their research goals – to understand human information seeking, and to develop the most effective technology to support it.

In the past decade, IR systems, from large-scale commercial Web search engines to specialized analysis software, have become ubiquitous. They have transformed the way in which we access information, and are for many an integral part of their daily lives. This shift towards everyday, ubiquitous IR systems is posing new challenges for empirical research. While it was previously possible to substantially improve IR systems by measuring and optimizing reasonably objective criteria, such as topical relevance, this is no longer sufficient. IR systems are becoming increasingly contextual and personal. They take into account information about their users' current situation as well as previous interactions, and aim to predict their users' requirements and preferences given new contexts. No longer can users or experts be asked to provide objective assessments of retrieval quality for such complex scenarios.

Online evaluation for IR addresses the challenges that require assessment of systems in terms of their utility for the user. The current state of the art provides a set of methods and tools, firmly grounded in and informed by the tradition of controlled experimentation. Giving an overview of these methods and their conceptual foundations, as well as guiding the reader on how to run their own online evaluations are the purposes of this survey.

In the next section, we define key concepts and terminology used throughout this survey. Then, we closely examine the motivations for online evaluation, and provide example use cases. Finally, we outline the scope and organization of the remainder of this survey.

1.1 Terminology

For the purpose of this survey, we adopt the following definition of *online evaluation*.

Definition 1.1. Online evaluation is evaluation of a *fully functioning* system based on *implicit measurement* of *real users'* experiences of the system in a *natural* usage environment.

The first key to the definition is *implicit measurement*, which we take to include any measurements that can be derived from observable user activity that is part of users' *natural* or *normal* interaction with the system [Kelly and Teevan, 2003]. Implicit measurements can range from low-level and potentially noisy signals, such as clicks or dwell-times, to more robust signals, such as purchase decisions. The key distinction we make between implicit and explicit measurements is that implicit measurements are a by-product of users' natural interaction, while *explicit* ones are specifically collected for feedback purposes. Both types of measures can also be combined into composite metrics capturing higher-level concepts, such as user satisfaction. These considerations give rise to a wide range of metrics, as discussed in Chapter 3.

We specifically include methods for *offline estimation*, *i.e.*, the estimation of online evaluation metrics based on past observations of users' behavior, in Chapter 4. Such estimation substantially increases the flex-

ibility of online evaluation and facilitates theoretically well-founded end-to-end evaluation of system components.

1.2 Motivation and Uses

Online evaluation is often seen as a set of methods that are particularly applicable in industry and industrial research. In these settings, a fully functioning IR system is typically available and in need of constant innovation. These factors have significantly contributed to the rapid adoption of online evaluation techniques in these settings. In industry, online evaluation approaches such as AB tests (*c.f.*, Section 2.4) and interleaved comparisons (Section 2.6) are now the state of the art for evaluating system effectiveness [Kohavi et al., 2009, Radlinski and Craswell, 2010, Li et al., 2011, Bendersky et al., 2014].

However, it is important to recall that much of the initial work on online evaluation originated in academic settings. Important motivations here were the need for reliable measurement of search quality of specialized search services [Radlinski et al., 2008c]. This line of work originated in the tradition of interactive IR. The fruitful exchange of ideas between applications and research continues today. On the one hand, practical challenges from IR applications motivate the development of online evaluation methodology; Chapter 2 gives a few examples. On the other hand, lessons learned in practical applications make their way into the state-of-the-art methodological tool set of IR researchers.

In the context of both practical applications and basic research, a key aspect of online evaluation is its reliance on controlled experiments. This allows the researcher to answer explanatory questions, which can explain causal relations in observed phenomena. In practical settings, this is crucial for correctly attributing observed changes in user behavior to system behavior. In research, this allows the development of theory in terms of causal concepts. More details on controlled experiments for online evaluation are provided in Chapter 2.

Finally, in Chapter 5, we discuss pros and cons of online evaluation, compared with more traditional offline evaluation methodology. This will help guide the reader to understand when an online evaluation is suitable, and when it is not.

1.3 This Survey

Online evaluation comprises a specific set of tools and methods that we see as complementary to other evaluation approaches in IR. In particular, online evaluation addresses questions about users' experience with an IR system that are quite distinct from those answered by *offline* evaluation using a test-collection-based approach, surveyed by Sanderson [2010]. Test-collection-based evaluation models users at varying levels of abstractions, and uses explicit assessments and offline metrics to assess system performance under these abstractions. Questions that are more appropriate for offline evaluation are those for which reliable and unbiased judgments can be collected from assessors (be they trained experts or crowdsourced representative users), but would be hard to infer from user interactions; an example being the quality of a document. Vice versa, online evaluation is more appropriate when the opposite is the case: for example, which of two topically relevant documents users find more interesting.

This survey does not discuss test-collection-based approaches in any detail, but points out conceptual differences when deemed appropriate. Furthermore, Chapter 5 focuses on online evaluation and test-collection-based approaches along a few dimensions.

Closely related to online evaluation is the long tradition of interactive IR (IIR) and the experimental framework developed for it, as surveyed by Kelly and Gyllstrom [2011]. We see online evaluation as a continuation of the IIR tradition, with considerable overlap. However, online evaluation extends to the specific requirements, limitations, and opportunities afforded by the scale, natural settings, and levels of control that are available in online settings. Generally speaking, IIR approaches, such as lab studies, are more appropriate for answering questions that require a high level of experimental control: for example, which tasks or queries a study participant is asked to solve. Conversely, online evaluation is preferred when researchers aim to study natural interactions at scale. This survey necessarily overlaps with some of the material that is relevant in the IIR setting, and we endeavor to point out connections as much as feasible. Our main focus will be on methodological questions that are specific to online evaluation settings.

Throughout this survey, we consider IR in a broad sense, including for instance recommender systems and advertisement placement. Many aspects of online evaluation are shared across these areas. For example, early work on using historical information for estimating online performance focused on ad placement [Langford et al., 2008] and news recommendation [Li et al., 2011]. We cover work in all these areas, and emphasize work that is specific to IR, such as search result ranking evaluation, as appropriate.

We have also highlighted particular places in the text with tips (such as the one below) that may be particularly useful for experimenters performing online evaluation without having access to very large user bases. While at first glance online evaluation may appear to be best suited to settings such as commercial search engines, in fact it has been widely used in academic settings as well.

Tip for small-scale experiments #1

Online evaluation can also be used with just tens of users, or hundreds of queries. Particular tips for experiments with few users are highlighted in the text with a box like this one.

1.4 Organization

We start in Chapter 2 by motivating the need for controlled experiments and detailing common experiment designs used in online evaluation, with a focus on experimentation methodologies that are particularly useful for IR. Following this, Chapter 3 gives an extensive overview of the variety of metrics that have been proposed for different tasks and research questions. Considering how to re-use online measurement data, Chapter 4 details offline estimation of online metrics from historical data. Turning to more practical issues, Chapter 5 discusses advantages and limitations of online evaluation, while Chapter 6 discusses practical issues around running online experiments. Finally, Chapter 7 concludes this survey with an outlook on emerging trends and open challenges.

2

Controlled Experiments

The building block of online evaluation is the controlled experiment, which allows us to identify whether observed changes in user behavior have been caused by changes to an IR system.

2.1 Online Controlled Experiments in Information Retrieval

Consider a developer at a news website who has developed a news ranking algorithm that ranks news articles based on readers' past reading behavior. This developer may strongly believe that this new approach will perform much better than the current ranking algorithm, causing higher user engagement as measured by click rates. How can the developer convince their management that this is indeed the case, and that the algorithm should be used for ranking news items in the future?

In this typical example of ranker development, the developer's colleagues would likely ask for empirical evidence to validate the claims. The most convincing empirical evidence can be obtained by conducting a *controlled experiment* [Shadish et al., 2002]. A controlled experiment is a type of scientific experiment that aims to explain cause-and-effect relationships, for instance, that switching the current algorithm for a

new one would cause users to have a better experience. In scientific discovery, this type of evidence is particularly valued because the *causal relationships* that can be discovered in this way are robust, for instance to certain changes in the distribution (or statistical properties) of the observed data. For example, a causal relationship between highlighting search result titles and click rate would be stable under changes in the distribution of the actual titles, while their correlation would typically be affected by such changes. [1]

The same characteristics make controlled experiments attractive for the practical development of interactive systems. Developers of such systems are typically interested in understanding effects of system changes on the users who interact with these systems. For example, it is important to understand whether an observed increase in user clicks (or any other measure of user engagement) is caused by a given algorithm and its parameter changes. The terminology, *online* controlled experiment, reflects the emphasis on identifying causal effect on user metrics. In particular, an online controlled experiment is able to distinguish the effects of the algorithm from external causes, such as a breaking news story that may increase engagement with all online search systems. An excellent introduction to the topic of causality from the perspective of interactive system development is [Bottou et al., 2013].

This survey focuses on controlled experiments as a means for identifying causal relationships using *explanatory* studies. The goal of these studies, as the name implies, is to explain the cause of a given phenomenon. In research programs, explanatory studies typically complement other types of studies, such as exploratory studies and descriptive studies. *Exploratory* studies are more open-ended than explanatory studies and are typically conducted at the beginning of a research program, to identify salient concepts or phenomena that are then investigated using narrower and more formal experiments. *Descriptive* studies describe the characteristics of the phenomena under investigation, while the explanatory studies that we focus on aim to explain

[1] An orthogonal issue is model completeness. Detected causal models can appear to change if a model is incomplete, and unobserved variables change. This underlines the need for systematic theory building, one of the foundations of modern science.

how these phenomena come about and how they would change in response to changes in their causes. In the context of IR, search logs are a particularly rich data source for exploratory and descriptive studies. Work on search log analysis and relevant methodology are reviewed by Jansen [2006] and Silvestri [2010]. Popular research methodology includes *content analysis* [Krippendorff, 2012], *data mining* [Han et al., 2011], and *visualization* [Tufte and Graves-Morris, 1983, Andrienko and Andrienko, 2006, Unwin, 2015].

Testing the effectiveness of algorithmic changes in large-scale Web applications is only one of the many important uses of controlled experiments in IR. When information retrieval initially required an evaluation framework, researchers turned to manually created collections of queries and documents. This type of controlled experiment is usually referred to as the Cranfield approach [Cleverdon, 1967]. It was first used to address the question of which document indexing scheme performed better, by creating static collections of documents and queries that allow repeated experimentation. The Cranfield studies led to the collection-based experimentation paradigm [Sanderson, 2010] and to the TREC tradition [Voorhees and Harman, 2005]. Today, evaluation campaigns are commonly modeled on this experimental paradigm.

A complementary experimental paradigm is needed to study interactive effects of information retrieval. This is the focus of study of interactive information retrieval (IIR) [Kelly, 2009]. While test-collection-based evaluation approaches typically abstract away from users and user-system interactions to focus on, *e.g.*, topical relevance, IIR aims to understand and model precisely those user and interaction characteristics that can inform the development of more effective interactive IR systems. Research in IIR heavily relies on methodologies such as laboratory studies or longitudinal studies, as surveyed by Kelly [2009]. Controlled experiments are one of the core methodologies. For example, an experiment that aims to compare users' reactions to two different search interfaces may randomly split participants into two groups, and expose each to a single search interface. Experiment designs like this form the basis of online evaluation. The designs that are most common in online evaluation are discussed in the following sections.

As IR systems become ubiquitous, scale to millions of users, and start providing information access on more and more diverse devices and platforms, new challenges and opportunities for controlled experimentation arise. For many of these systems, users and their information needs are too diverse and dynamic to be adequately captured in static test collections. Further, metrics and interaction effects that are best captured in-situ make even large-scale laboratory studies infeasible. In response, experimentation methodology is extended to address these new challenges. These developments also facilitate a new scale, immediacy, and new natural settings for conducting controlled experiments, that allow researchers to study phenomena that were previously difficult to capture. Towards the end of this chapter, we specifically focus on these new challenges and opportunities for controlled experimentation that arise from the characteristics of online experimentation.

Chapter Outline In this chapter we detail the ingredients for designing online controlled experiments. We define key terminology and ideas of controlled experiments more formally, and give an overview of how experiments are typically designed in Section 2.2 and analyzed in Section 2.3. Then, we turn our focus to two experiment designs that are the most commonly used in online controlled experiments for IR: *between-subject* experiments (Section 2.4) and *within-subject* experiments (Section 2.6). For each type, we provide descriptions of and references to a number of case studies that illustrate how they are applied and types of studies they support. We also discuss recent development towards more complex designs for online controlled experiments (for between-subject designs in Section 2.5, for within-subject designs in Section 2.7).

2.2 Planning Controlled Experiments

Information retrieval strongly relies on controlled experimentation both for developing theory and for validating practical applications. In both cases, experiment planning starts from one or more hypotheses that are to be tested in the experiments. We discuss hypotheses and how they inform experiment design and data collection in the first part of this

section. Next, we discuss possible choices of experiment designs, which define how experimental treatments are applied during data collection.

2.2.1 Hypotheses and Variables

The goal of a controlled experiment is typically to test one or more hypotheses [Shadish et al., 2002]. A hypothesis connects two or more variables in the form of one or more testable predictions. For example, a hypothesis in the news recommendation problem described at the beginning of the chapter could be:

> H_1: Increasing the weight X given to document recency in the ranking function will increase user click-through rate Y.

In contrast, the corresponding *null hypothesis* could be:

> H_0: Increasing the weight X given to document recency in the ranking function will *not* increase user click-through rate Y.

A controlled experiment can then be performed to test whether H_1, also known as the *alternative hypothesis*, can be accepted or rejected against H_0.

The example hypothesis H_1 connects two variables (recency ranking weight X and click-through rate Y) and makes a testable prediction (Y increases as X increases). The hypothesis is testable, because we can construct an experiment to falsify it against an appropriate null hypothesis (for example, increase X and measure Y). Note that this hypothesis is formulated in the form of a causal mechanism. We hypothesize that X affects Y, but not vice versa (manipulating X is expected to change Y, while manipulating Y is not expected to result in changes in X). Thus, measuring the correlation between variables is not sufficient, only a controlled experiment can test the hypothesis (we come back to this example towards the end of this section to examine possible effects of confounding variables on measured correlations).

Two common types of variables can be involved in the formulation of hypotheses. First, *independent variables* are those that can be manipulated or controlled by the experimenter (we refer to them as

variables belonging to the set $\mathcal{X}$). In the cause-and-effect relationships modeled in explanatory studies, these are the *causes* or *explanations* that bring about the hypothesized effects. Depending on the community, independent variables are also known as predictor, condition, factor, or input. Effects are then captured by *dependent variables* (belonging to set $\mathcal{Y}$), called so because their values are hypothesized to depend on the independent variables. Dependent variables are also known as response, outcome, output, key-performance indicator (KPI), overall evaluation criterion (OEC), performance metric or measure. In the example above, the click-through rate Y is a dependent variable. It cannot be set directly by the experimenter, but rather is hypothesized to respond to changes in the independent variable X.

Also common, but less often discussed, are *quasi-independent* variables. These are variables over which the experimenter has only partial control or that are difficult to select (for example, the expertise of the experimental subject, or gender). Because these variables cannot be (randomly) assigned by the experimenter, there typically is a risk that quasi-independent variables can suffer from confounding. *Confounding variables*, also known as *confounders*, refer to extraneous variables (variables that the experimenter did not record or control for) which may affect both dependent and independent variables. Confounders can have strong effects on an analysis, *e.g.*, resulting in spurious correlations between independent and dependent variables (leading to the risk that the experimenter detects a relationship that is not actually causal), or mask actual effects (leading the experimenter to falsely miss causal relationships between independent and dependent variables). Randomized controlled experiments (*i.e.*, the randomized assignment of subjects to experiment conditions) is the safest way to avoid confounding. However, in experiment designs with quasi-independent variables this may not be possible, and great care must be taken to minimize the risk of confounding [Campbell and Stanley, 1966].

Considering our running example of news recommendation, we can now exemplify the effect that confounding variables can have on measured correlations between variables. Assume the experimenter had access to a previously collected data set, that contained measurements

of the recency weight X and click-through rate Y, but importantly the data set did not originate from a controlled experiment. If a high correlation between X and Y is measured, it is possible that this reflects an underlying mechanism that is consistent with hypothesis H_1. However, it is also possible that a confounding variable has affected the observed measurements. For example, the recency weight may have been set to a higher overall value at some point in the past, and (independently) user behavior may have changed to a higher click-through rate. Alternatively, different types of queries may be treated with different recency weights and at the same time have naturally higher click-through rates. In both cases, any level of correlation could be measured, without any causal relation being present. Possibly worse, a positive correlation could be measured even if the inverse causal mechanism were present (a positive correlation can be measured even if in fact higher recency weight caused lower click-through rates) – this could lead to examples of the co-called Simpson's Paradox [Simpson, 1951, Bottou et al., 2013]. Careful analysis can help identify possible confounding variables and strengthen analysis. However, the only safe method to establish causal relationships is controlled experimentation.

The process in which experimenters arrive at a hypothesis can be more or less structured, depending on the type of experiment. For example, in practical applications such as the development of ranking algorithms for a search engine, there may be a default hypothesis that states that a newly developed variant of the system will improve the search engine's target metric in comparison to the current production system. In more theory-oriented research, hypotheses may be derived from a broader model or a more general theory (see [Azzopardi, 2014] for a recent example), or they may be derived from insights gained through an earlier exploratory or descriptive study. In either case, the same methodology for hypothesis testing can be employed.

2.2.2 Experiment Design

An experiment design specifies how data will be collected to test a given hypothesis [Lawson, 2014]. It specifies how experiment subjects are assigned to the various experiment conditions (dependent

variables) that are defined by the hypothesis. Experiment designs need to be carefully constructed to ensure the reliability of the experiment to detect causal effects (*e.g.*, to avoid confounding). In other words, it must be such that any measured effects truly reflect causal relationships between the variables, and that effects of potential external factors are eliminated as much as possible.

For a concrete example, let us again consider the news ranking scenario discussed above. Assume that we want to construct an experiment where two levels of ranking weight X are tested: low and high. Naïvely, one could construct an experiment where the system uses a low ranking weight on day 1, and a high ranking weight on day 2. Notice that with this design, it is unclear whether any changes in click behavior between days can be attributed to X, or whether additional factors may play a role. For example, there may be changes in click behavior over time, or differences between weekends and weekdays. To avoid such *spurious correlations*, we have to ensure that the assignment in the experiment is independent of potential external influences (confounding variables).

The key insight that the assignment of subjects to experimental conditions must be independent of potential confounding factors has led to a series of standardized experiment designs that represent best practices and insights gained from prior research. A commonly used type is known as factorial designs (see [Kelly, 2009]). They allow flexible testing of one or more independent variables (factors) at different levels. In the news ranking example above, only one factor with two levels was used, but we could add more factors, such as the color of the links to news items, and whether or not related images are shown alongside links. The type of factorial design is then specified by the number of factors, and levels in each factor that are available. A $2 \times 3 \times 2$ design would indicate that there are three factors, two with two levels each and one with three levels. The design then specifies the combinations presented to each subject. The challenge is to carefully avoid introducing spurious correlations between factors and sequences of factors, while minimizing the number of subjects required.

In factorial designs, different combinations of factors are presented in a sequence in a pre-specified order so that an equal number of

subjects observe each combination of factors. However, in online experiments this sort of assignment by subject is typically difficult, especially as different subjects usually interact with the system for different lengths of time. Therefore, randomized assignment to conditions is preferred. This is a major difference between the online setting and laboratory experiments, where the main concern is typically the need to assign test subjects in a way that allows valid results with relatively few test subjects.

The choice of experiment design affects the sample size required for a given experiment. When factorial designs are used, the sample size should be a multiple of the number of possible assignments. In randomized experiments, the required sample size can be estimated using variance estimates for the response variables, a minimum difference between conditions that the experiment is intended to detect, and additional details related to the planned analysis. A number of tools for computing sample sizes are summarized in Section 6.3.5. If variance is expected to be high and large samples are required, variance reduction techniques may reduce required sample size, to lower experiment impact on users and make more effective use of the collected data (see Section 2.5).

Online evaluation in industry is often concerned with relatively incremental changes to a production system and may require hundreds of thousands or millions of impressions for detecting changes of the expected magnitude at the observed levels of variance [Chapelle et al., 2012]. In studies that aim to detect larger effects, much smaller amounts of data may suffice, *e.g.*, from tens of users [Matthijs and Radlinski, 2011]. We provide further details on these practical aspects of planning and analyzing controlled experiments in Chapter 6.

Finally, the experiment design should be planned together with the methodology for data analysis. Data analysis will be discussed in section Section 2.3.

2.2.3 Unit of Experimentation

One of the key attributes of a controlled experiment is the unit of experimentation: At what granularity are experimental conditions

tested? Kohavi et al. [2009] describes that typically the unit of experimentation is a "user" (subject) so that a given user always has a consistent experience throughout the duration of an experiment. Smaller units are possible when interactions at these levels can be neglected (for instance, different ranking functions may be used for different queries by the same user in a search scenario, or perhaps the user can be randomly assigned to a condition the first time they visit a website on a given day). Using smaller experimental units can dramatically reduce data requirements, but whenever such assumptions are made they should be validated in a preliminary experiment. Larger units of experimentation may be required when users cannot be assumed to be independent, *e.g.*, in the presence of network effects. Work on extensions to such settings is reviewed in Section 2.5.

For the remainder of this chapter, we assume the unit of experimentation is the user unless stated otherwise.

Tip for small-scale experiments #2

Experiment designs with small units of experimentation have a high effective sample size and therefore make very effective use of experiment data. For example, analysis on the query level is typically more sensitive than analysis on the session or user level. However, be cautious about possible biases when units are not in fact independent.

2.3 Data Analysis

Data analysis is designed to answer questions about the collected data. These could include questions related to modeling (*e.g.*, Can the structure of the data be summarized in a compact model?), prediction (*e.g.*, What response should I expect if I change some subset of the modeled variables in a specific manner?), and hypothesis testing (*e.g.*, Are my observations likely due to chance, rather than a specific mechanism I proposed as a hypothesis?). These questions are in many ways related to each other, and can be answered using the same set of mathematical and statistical tools. The approach for data analysis

is tightly coupled with the experiment design and should be planned at the same time, as the planned analysis affects decisions about data collection, such as the required sample size (*c.f.*, Section 2.2).

In this survey we focus on data analysis using a statistical technique called regression analysis [Gelman and Hill, 2006]. We start with an overview of regression analysis and illustrate it using simple and more complex examples. Finally, we draw parallels to the complementary analysis technique ANOVA, and briefly discuss recent work on issues that arise in continuous testing.

2.3.1 Regression Analysis

Regression analysis is a very flexible tool for modeling, prediction, and hypothesis testing, and is widely used in online experimentation [Bakshy and Frachtenberg, 2015, Bendersky et al., 2014, Deng et al., 2013, Drutsa et al., 2015a, Gui et al., 2015, Hofmann et al., 2014]. In its simplest form it covers simple linear models with a small number of variables (including the arguably most common use case, the t-test) but can be easily extended to a wide range of complex models and hypotheses [Gelman and Hill, 2006], including hierarchical models or models that include network effects.

Regression analysis models dependent variables Y as a function of independent variables X and coefficients β:

$$\mathbf{Y} = f(\mathbf{X}, \beta)\,, \tag{2.1}$$

where $\mathbf{X}$ is a matrix with rows corresponding to the experiment units' independent variables and one column per unit of experimentation, and $\mathbf{Y}$ the vector recording the corresponding units' dependent variables. Considering our running example of news ranking, a model of this form formalizes the notion that click-through rate Y is some function of recency ranking X.

The functional form of the model f has to be specified by the experimenter. A frequent choice is a linear functional form:

$$\mathbf{Y} = \mathbf{X}^T\beta + \epsilon\,, \tag{2.2}$$

where the experimenter assumes that the dependent variables can be modeled as a weighted linear combination of the independent

variables, plus a noise term, ϵ. In our running example, choosing this form models the assumption that the click-through rate Y changes linearly with recency ranking X.

Given a model specification like the linear model above, and data collected in a controlled experiment, standard tools for regression analysis can be used to estimate the parameters (coefficients) of the specified model, and to conduct significance tests.

One of the standard tools employed for this purpose is ordinary least-squares (OLS). Robust estimation methods for regression analysis for a wide range of models have been developed over the past years and now available as part of commonly used data analysis toolboxes. The available tools make this type of analysis possible and practical for analyzing even large amounts of data (*e.g.*, millions of samples). Example packages are listed in Section 6.6.1. An excellent resource for further study is by Lawson [2014].

There is a close connection between regression models and tests of statistical significance. For example, it turns out that the functional form of an independent sample t-test is equivalent to the linear model above with a single dependent and independent variable, and two coefficients (β_0 which can be interpreted as the response level without treatment, and β_1, which corresponds to changes in the response due to the treatment). The analysis consists in estimating the two coefficients from experiment data, and then assessing whether β_1 is statistically significant (unlikely to have occurred due to chance).

An example study that uses regression analysis in a more complex model is [Hofmann et al., 2014]. It uses a so-called mixed-effects generalized linear model to capture effects of subject, task and experiment condition in an eye-tracking study of query auto completion. The following model is used for analysis:

$$g(y_{ij}) = \beta_0 + \beta_1 x_{ij} + p_i u_i + t_j v_j + \epsilon_{ij}. \tag{2.3}$$

Here, $g(y_{ij})$ denotes generalized response variables that can be transformed using link functions $g()$. For example, this allows modeling binary response variables, or the logarithm of the response. Responses are modeled as linear combinations of the treatment condition x_{ij}, the effects of participant u_i and topic v_j, and the noise component ϵ_{ij}. The

resulting model allows the experimenter to accurately and automatically estimate the model coefficients that best account for the variance due to the crossed effects of topic and participant, and separate them from the treatment effects that are the main interest of the study.

Tip for small-scale experiments #3

Regression analysis is particularly well-suited for analyzing small-scale experiments.

In cases where the analyzed data are the result of a carefully constructed controlled experiment, the described regression analysis can be used to support causal claims. Note that this heavily relies on the assumptions that underlie controlled experimentation, such as independence between experimental assignments and observed outcomes.

2.3.2 Relation to ANOVA

In laboratory studies, a traditional tool for data analysis is Analysis of Variance (ANOVA). Briefly, ANOVA analysis uses a model to quantify variance contributed by the different factors to the outcome of an experiment (*e.g.*, the tested hypothesis), comparing this to the overall sample variance, and assessing whether the model explains a significant portion of the overall variance [Tabachnick and Fidell, 2013]. Traditionally, ANOVA can refer to the method of decomposing variance for exploratory data analysis, to significance testing of whether some of these components significantly contribute to the observed variance (using F-tests), or to a specific procedure for arriving at the desired decomposition and significance test. The methodology is widely used, especially in the psychology and the social sciences.

The relationship between regression analysis and ANOVA is well understood [Gelman et al., 2005, Gelman and Hill, 2006]. In particular, Gelman et al. [2005] argue that the type of analysis afforded by ANOVA is more relevant than ever. At the same time, this analysis can be conducted on top of regression models. This results in a flexible modeling and hypothesis testing framework. In particular, regression anal-

ysis makes it straightforward to deal with missing or unbalanced data, differences in group sizes, and complex hierarchical models. Regression models can also be used to gain additional insights, beyond attributing variance. For example, they can be used to predict changes in response variables as a function of unit changes in the independent variables.

2.3.3 Continuous Testing

A recent development is the introduction of tools (dashboards) that allow experimenters to continuously monitor running experiments [Johari et al., 2015]. This introduces the problem of *continuous testing*. Traditional analysis and hypothesis testing assumes that the experiment setup, including the sample size, is fixed before the start of the experiment, and the corresponding analysis techniques are only valid if they are applied once data collection is complete (in particular, data collected in an experiment cannot be used to make decisions on when to stop data collection). Decision making with continuous observation requires an alternative analysis method that models the probability of observing any particular outcome at any point throughout the experiment. Otherwise, the probability of error may be underestimated. First approaches that address this problem have been proposed recently [Johari et al., 2015, Kharitonov et al., 2015c].

In the next sections, we examine experiment designs (and their corresponding regression models) that are particularly common in IR research.

2.4 Between-subject Experiments

By far the most common type of controlled experiment on the web is AB testing [Kohavi et al., 2009]. This is a classic *between-subject experiment*, where each subject is exposed to exactly one of two conditions (e.g., control – the current system, or treatment – an experimental system that is hypothesized to outperform the control). In this setup, there is a single independent variable, namely a binary indicator that identifies the system that a given subject is exposed to. A typical hypothesis is that the system has a significant effect on some performance

metric (the dependent variable). Crucially, the assignment of subjects to conditions (systems) is randomized, so that a causal relationship between condition and performance metric can be established.

On the surface, AB tests are very simple to implement, as they constitute the most basic type of controlled experiment. However, correctly implementing randomized assignment of experiment units, data collection, and validation for systems that are distributed over hundreds or thousands of servers, and serve millions of users is a major challenge. Kohavi et al. [2009] provide a detailed account of AB testing at scale, with a strong emphasis on practical aspects encountered in a production environment. For example, they discuss how the choice of performance metric can affect variance, and thereby the required sample size for an experiment, how to compute confidence intervals for percentage changes, and strategies for dealing with robots. Kohavi et al. [2012] detail several case-studies that illustrate surprising results of online experiments that can result from the choice of performance metric, instrumentation, or from carryover effects. Kohavi et al. [2013] give an updated overview and expands on engineering lessons of how to implement large scale AB tests in practice.

Previous work on controlled experiments on the web has primarily focused on practical challenges posed by the scale and complexity of systems like commercial web search engines. Understanding of these issues is now maturing. We discuss this topic in Section 6.3.

Case Studies AB testing has quickly developed into a gold standard for online experimentation in industry settings. For instance, the work of Dong et al. [2011] focuses on learning personalized news recommendation. Their learning approach relies on estimating user engagement with a news model, and models are learned from exploration data collected in a randomization user group. An AB test (there called *bucket test*) is conducted to compare two methods for interpreting user actions (*e.g.*, clicks) for learning personalized recommendations. Performance is measured in terms of click-through.

Bendersky et al. [2014] also focus on recommendation, but specifically addresses the problem of recommending related videos after

visitors to a video viewing website have finished watching a video. Recommendation is modeled in terms of video topics and relies on estimates of topic transitions. The resulting algorithm and an alternative retrieval-based recommendation approach are compared to their production system in a 3-way AB test. Performance is assessed in terms of the metrics *watch time*, *completion rate*, and *abandonment rate*.

Kohavi et al. [2012] provide several examples of online controlled experiments, related to information retrieval and other web based systems, to illustrate potential pitfalls in online experimentation. For example, they highlight the difficulties involved in choosing an evaluation metric for controlled experiments on web search engines. Metrics that focus on long-term effects, such as *users/month* are often closely aligned with the high-level objectives of the search engine operator (*e.g.*, query share), but may be difficult to estimate. Shorter term metrics may be easier to measure, but may not be as highly correlated with the operator's objectives.

2.5 Extensions to AB testing

As the use of online controlled experiments is becoming more common, research has shifted to extending this methodology. Of particular interest are methods for increasing experiment throughput (the number of experiments that can be run on a given amount of online traffic), on enabling more flexible experiment designs, and on automating online controlled experimentation. We discuss each of these in turn below. More specific extensions to the development of online metrics (Chapter 3), and to estimating experiment outcomes using historical data (Chapter 4) are discussed in their own chapters.

While large online systems can have millions of users a day that may potentially participate in online controlled experiments, this has by no means removed the need to design experiments to be as sample efficient as possible. For *offline* experiments recruiting large numbers of users is typically prohibitive and these are therefore typically limited to tens or hundreds of users. Having access to millions of potential subjects per day may seem ideal. However, in practice, experiments are on the one

hand still costly (they may have adverse effects on users, which should be minimized through economical experiment designs) and on the other hand, the number of experiments that *could* be run is limited only by system developers' and researchers' ingenuity, which seems boundless.

This continued need for economic experimentation has triggered recent research that focuses on increasing the throughput of online controlled experiments. Deng et al. [2013] propose the use of pre-experiment data to improve the sensitivity of online controlled experiments. The idea is to identify variables along which dependent variables tend to be highly variable. Once identified, these co-variates can be used to stratify sampling during data collection. Data analysis can then be narrowed to comparing effects of independent variables within each stratified sample, and aggregating over these to obtain a combined effect estimate. They demonstrate that the technique can result in a 50% reduction in required sample sizes in real online experiments. An extension of this technique to an objective Bayesian setup is proposed in Deng [2015]. In a similar vein, Drutsa et al. [2015a] propose the use of predicted user engagement as a dimension for stratification during experimentation and analysis.

Tip for small-scale experiments #4

The above stratification techniques can greatly improve experimental sensitivity.

A second way in which sensitivity can be improved in AB tests is by considering how the metrics are summarized. Rather than simply computing means, Drutsa et al. [2015c] and Nikolaev et al. [2015] recently showed that considering medians, or extreme value changes, may lead to more statistically significant results for AB tests. They also showed that the choice of statistical significance test can affect the outcome.

A third approach to increasing the throughput of online controlled experiments is to run many experiments in parallel on the same traffic. Practical implementations of such systems are discussed in [Tang et al., 2010, Kohavi et al., 2013]. The experiment design corresponds to a fully

factorial experiment design, meaning that each user can be randomly assigned to one of the conditions for each of tens or hundreds of experiments. The key underlying assumption is that there is no interaction between the experiments that run in parallel; such an assumption must be verified during data analysis to avoid invalid analysis.

A very different proposal for increasing experimental agility is counterfactual reasoning [Bottou et al., 2013]; see Chapter 4 for a highly related topic. The proposed method has two main components. First, the use of causal graphs [Pearl, 2009] can capture causal (in-)dependence relations that are known in a complex system, *e.g.*, due to the system's design. Known independence can be used to draw additional conclusions from a single controlled experiment. Second, the authors propose to relax the experimental levels to sampling from a continuous variable (*e.g.*, some continuous system parameter). During analysis, data collected in this way can be used to estimate system performance under alternative distributions over this variable. This setup is equivalent to running infinitely many controlled experiments simultaneously.

Tip for small-scale experiments #5

Counterfactual analysis allow the estimation of likely outcomes given historical log data, without needing to run controlled experiments on new users.

So far, we have assumed the individual user as the experimental unit. In settings such as web search this assumption is typically reasonable (although it may not always hold, and should be verified). However, the advent of social media has introduced new challenges for online controlled experiments. In social media, users are linked to each other, meaning that exposing a single user to an experimental treatment can potentially affect other users they are connected to. For example, assume a user searches Twitter, and is assigned to a treatment condition that tests a new result ranking algorithm. If the user retweets one of the search results, their followers are now affected by the experimental treatment (without the new ranking

algorithm the user may not have found the result, may not have retweeted it, and their followers would not have seen the results). If the resulting *network* effects are strong, they need to be modeled and can substantially complicate the design of online experiments.

A detailed survey of online experiments with network effects is given by Walker and Muchnik [2014]. Ugander et al. [2013] propose an approach to identifying possible units of experimentation using graph algorithms. They propose to identify highly connected components of a social network graph, and randomizing across these connected components. By increasing or decreasing the threshold at which a component is considered dense, the experimenter can balance the trade-off between the risk of interactions between components and the effective sample size. Gui et al. [2015] propose a method for analyzing experiment data when network effects are present. Their approach uses a regression model to separately estimate the effects of cluster assignment and subject identity.

As the amount of online experimentation within a system increases, the need to automate aspects of online experimentation arises. Taking a step in this direction, Bakshy et al. [2014] introduce PlanOut, a system that allows experimenters to specify experiment designs in a simple experiment description language. Their system supports AB tests and more complex designs, including overlapping and factored designs. A commercial system that provides this functionality is SigOpt.[2]

Research in sequential design of experiments aims to remove the need to specify each experimental comparison separately. In particular so-called *bandit* approaches have been proposed to automatically select the most promising experimental comparisons to run. Bandit approaches are particularly suitable for this purpose as they balance the need for exploration (ensuring that new, potentially promising areas of the experiment design space are explored) with exploitation (focusing on areas that are known to perform well). Li et al. [2010] propose a *contextual bandit* approach for learning personalized news recommendation through automated controlled experimentation. A recent overview of this rapidly growing research area is provided by Burtini et al. [2015].

[2]See `https://sigopt.com/`.

An emerging research area focuses on *dueling* bandit approaches that provide performance guarantees even when only relative feedback (*e.g.*, preferences) can be observed [Yue et al., 2012, Dudík et al., 2015]. A dueling bandit approach to large-scale ranker evaluation is proposed in Zoghi et al. [2015].

2.6 Within-subject Experiments

A second popular experiment design uses a *within-subject* setup. In contrast to between-subject experiments, within-subject designs expose study participants to both experimental conditions (two levels are typical, although extensions to multiple conditions are possible and are an area of active research (*c.f.*, [Schuth et al., 2014] and the discussion below). In laboratory studies, this could mean that study participants are exposed to two search interfaces and are asked to specifically compare these. A novel form of within-subjects experiments, so-called interleaved comparison [Joachims, 2002, Radlinski et al., 2008c], has been developed specifically for online experimentation in information retrieval. These interleaved comparison methods are especially valuable for online evaluation of rankings, such as search result rankings. In this section, we focus on within-subject experimentation using interleaved comparison, as this is particularly relevant to online evaluation in information retrieval.

Tip for small-scale experiments #6

Within-subject experiments can dramatically reduce variance, allowing much higher sensitivity that may be essential to run practical experiments at small scales.

Between-subject and within-subject experimental designs have complementary advantages and limitations. Between-subject designs make less restrictive independence assumptions. They only require that independence holds between experimental units. This makes between-subject experiments the most generally applicable. At the

same time, between-subject experiments tend to suffer from high variance. Because each subject is exposed to only one of the treatments, models of subjects' behavior within a group need to account for all the variance that naturally occurs across users with different information needs and preferences. In contrast, within-subject experiments match each individual subjects' response to both conditions. As a result, differences between treatments can be separated from variance between subjects. This difference in data analysis can lead to one to two orders of magnitude increases in effective sample size for studies of comparable dependent variables, and therefore much reduced data requirements [Chapelle et al., 2012]. On the other hand, within-subject designs are less flexible than between subject experiments. They require additional independence assumptions (*e.g.*, across queries), and are only applicable in settings where a within-subject setup is expected to not introduce significant bias (*e.g.*, where it can be applied without substantial changes to the underlying user experience).

In the remainder of this section, we first give an overview of the general approach to interleaved comparisons. Then, we zoom in on specific instantiations of this framework that have been proposed in the past. Finally, we discuss extensions that go beyond individual rankings (*e.g.*, verticals) and towards comparing multiple rankings in a single experiment.

Joachims [2002] first proposed the use of interleaved comparison for IR online experimentation. The proposed algorithm is called *balanced interleaving*. It is based on the axiom that the constructed interleaved result should minimally differ from either candidate ranking. All interleaved comparison methods consist of two components. First, a method for constructing interleaved comparisons lists (which are shown to search users) specifies how to select documents from the original rankings. Second, a method for inferring comparison outcomes takes as input the original lists, the interleaved list shown to study participants (and possibly additional information on how this list was constructed), and observed interactions of study participants with the interleaved lists. This second component then specifies how to interpret observed interactions.

Data analysis of interleaved comparison outcomes can be formulated in terms of regression analysis as follows:

$$\mathbf{T} = f(\mathbf{X}_I, \beta), \tag{2.4}$$

where $\mathbf{T}$ is a transformation of the paired dependent variables, such that $t_i := y_{i1} - y_{i2}$, where y_{ik} is the observed outcome for condition k in trial i. Following from this formulation, results are typically analyzed using paired-sample t-tests. The use of more complex generalized linear models is possible and follows from this formulation; however, to the best of our knowledge, it has not been explored in the literature to date.

Now that we have outlined the general principle of within-subject interleaved comparisons and approaches to data analysis, we briefly review specific interleaved comparison approaches. A detailed discussion of metrics for interleaved comparisons is given in Section 3.5.

Case Study Although interleaving was and is primarily developed in academia, it is today predominantly applied in industry research and product development. A notable recent exception is the work of Matthijs and Radlinski [2011], who demonstrate its use in a laboratory study of Web search personalization. The authors propose a personalization approach that personalizes results rankings based on user profiles extracted from users' browsing history. The approach is evaluated both offline (using user judgments) and in an online interleaved comparison. The online experiment is implemented in a browser plugin that intercepts and re-ranks search results. The personalized results are interleaved with non-personalized rankings using Team-Draft Interleaving. Results were obtained for 41 study participants who used the plug-in for two months and generated just over 6,000 query submissions. This volume of data was sufficient to detect substantial and significant improvements of the proposed personalization approach over the non-personalized baseline.

Tip for small-scale experiments #7

[Matthijs and Radlinski, 2011] can be considered a tutorial on how to run evaluations at small scale. Having recruited 41 users, who issued 6,000 queries during normal search use over two months, the statistically significant results allowed the evaluation of a number of ranking systems.

2.7 Extensions to Interleaving

A number of refinements have been subsequently published.

Radlinski et al. [2008c] first showed that the original "balanced" interleaving algorithm is in fact biased in certain rare circumstances, proposing improved mixing and scoring policies termed Team Draft interleaving. This policy alternates (randomly) between the two input rankings — like teams constructed in friendly sports games — appending selected documents to the ranking shown to users. This also simplifies the scoring rule, as documents are always taken to indicate a preference for the input ranking that selected the document. However, this policy was in turn showed to be insensitive to certain ranking changes by Hofmann et al. [2011b], who proposed a probabilistic mixing policy that can in principle produce any ordering of documents, and computes the preference between the ranking systems as the expectation over all sequences of random choices that could have led to the particular ranking and paves the way towards estimating interleaved comparison outcomes from historical data (Chapter 4). Alternatively, Radlinski and Craswell [2013] proposed how the mixing policy can be seen as the solution to an optimization problem given a scoring policy and constraints on what rankings may be shown to users. Schuth et al. [2014, 2015a] showed how interleaving can be extended to apply to sets of retrieval systems, rather than simply one pair at a time.

If instead of changing the mixing policy, we consider modifying just the scoring rule, Yue et al. [2010a] showed how the scoring rule can be optimized to minimize the number of required observations of user behavior to reach a conclusion. Similarly, Yue et al. [2010b]

and Hofmann et al. [2012a] proposed different approaches to improve sensitivity by removing more subtle biases in user behavior beyond position – such as bolding effects and caption length. In a contrasting approach, Schuth et al. [2015b] recently showed how interleaving can be re-cast as predicting the outcome of a standard A/B evaluation with increased sensitivity, showing how the scoring rule can be learned to maximize performance on the corresponding prediction problem.

Chuklin et al. [2013a, 2014] extended interleaving approaches to search results with specialized verticals. Kharitonov et al. [2015b] present a Generalized Team-Draft approach that extends beyond ranked lists, to results arranged in grids (*e.g.*, for image search). These works demonstrates that interleaving can be applied to settings with more complex user interfaces that go beyond individual ranked lists. An open research problem is the extension of within-subject experiment designs to more general user interfaces and interaction models.

Finally, Hofmann et al. [2012b, 2013c] presented a technique to avoid online testing altogether, showing how the outcome of an interleaving experiment can be predicted using previously collected logs of user behavior if the user behavior is collected using a suitably randomized IR system. More discussions on this technique are in Chapter 4.

In addition to the extensions discussed above, which are specific to online interleaving experiments, the general extensions to controlled experiments that we discussed in Section 2.5 can be extended to interleaved comparisons. Mixed between-subject / within-subject designs are also possible, *e.g.*, to test effects of ranking changes in a within subject design, and interface changes in a between-subject design simultaneously.

3

Metrics for Online Evaluation

This chapter presents an overview of the most commonly observed indicators of online user engagement, showing how these can be interpreted to measure retrieval quality. Metrics based on observable user behavior aim to capture system effectiveness in terms of the quality of documents, which in turn drives user satisfaction. There is a large amount of literature on such evaluation metrics, with a variety of complexities in both observing the user interactions, and interpreting them. Therefore, we group the presented metrics by the unit of observation, and the types of evaluation questions that they allow to be answered.

In particular, from an IR perspective we consider *document-level*, *result-list-level* and *session-level* metrics, as well as observations that permit *relative* as well as *absolute* metrics to be computed.

3.1 Introduction

Historically, metrics for online evaluation in IR were inspired by the relevance feedback literature, which considers how user feedback can inform a retrieval system about which documents are relevant. In particular, Kelly and Teevan [2003] provide a detailed survey of many

implicit relevance feedback indicators. Here, we take a complementary approach, providing an overview of how implicit feedback can be interpreted as a metric that describes the quality of an IR system. Further details of many related metrics are also summarized by Fox et al. [2005].

It is important to note that *quality* can be defined in many ways. For instance, when we measure the quality of an IR system, we may be interested in computing an *absolute* real valued score that measures the relevance of documents returned by the system. We may wish this score to be a value that can be compared to that produced by other systems over time, to track how this IR system performs relative to other systems, and relative to itself at other times. On the other hand, when we measure quality we may instead be interested in asking a much simpler question: Given two IR systems, which is better at retrieving relevant documents right now? This is a *relative* comparison, which is easier to make but harder to generalize in the future. For example, knowing that systems A and B are better than system C does not imply the relative performance between A and B. To make the problem more challenging, even transitivity may not hold in pairwise relative comparisons [Chapelle et al., 2012, Dudík et al., 2015].

Another dimension is the granularity of the quality assessment that we require. Often, we may be interested in the quality of the ranking system as a whole: Given a user query, the IR system produces a ranked list of results, and we would like to know how well this list and presentation satisfy the user. The performance in question is at the *list* level. On the other hand, we may be interested in a question such as which of the documents returned for a given query are best. This is a *result-level* question, and would be answered most efficiently using a different experimental design.

We may even view the goals of the data collected during an online evaluation in a number of ways: Do we wish to simply perform an evaluation, or do we also want to use the data for future optimization? Do we wish to focus on short-term relevance metrics, or longer-term metrics that may better reflect long-term user engagement? Can our online evaluation explicitly request user feedback, or do we wish to limit the evaluation to metrics that can measure the quality of the IR

system without requiring explicit assessments from users? In providing an overview of the literature of online evaluation metrics, we attempt to point out individual examples of work that focuses on these questions.

We now present the metrics by filling in the following table

Table 3.1: Approaches to Online Evaluation

Granularity	**Absolute evaluation questions**
Document	*Are documents returned by this system relevant?* (3.2)
Ranking	*How good is this system for an average query?* (3.4)
Session	*How good is this system for an average task?* (3.6)
Granularity	**Relative evaluation questions**
Document	*Which documents returned by this system are best?* (3.3)
Ranking	*Which of these IR systems is better on average?* (3.5)
Session	*Which of these IR systems leads to better sessions on average?* (3.7)

3.2 Absolute Document-level Metrics

Most common online metrics start with the click on a search result as the basic unit of observation. We can then consider particular attributes of the clicking behavior, such as dwell time on the clicked document, to compute metrics. Recently, learned metrics have been developed that integrate information about user behavior and other context information with the goal of accurately estimating user satisfaction with a clicked document. We discuss commonly-used metrics and recent developments below. An overview is provided in Table 3.2.

Click-through rate is the simplest click-based metric, and is commonly used as a baseline (such as in Chapelle et al. [2012]). For example, it can represent the average number of clicks a given document receives when shown on the search result page (SERP) for some query. Click-through rate can be defined similarly at the ranking level (see Section 3.4). As clicks are attributed to individual documents, it can also provide a relevance metric for each document. However, per-document

CTR has been shown to be noisy and strongly biased, particularly due to document position [Joachims et al., 2007]. Other sources of bias in CTR include caption and other presentation attributes [Clarke et al., 2007, Yue et al., 2010b, Hofmann et al., 2012a]. On the other hand, Hofmann et al. [2010] showed that CTR agrees well with purchase-based evaluation in a commercial/professional search setting.

Dwell time is frequently used to improve over simple click metrics: A click on a search result is most often an indication that a user *expects* to find relevant information in the document, based on the caption that they were presented with. A common refinement aimed to reflect the quality of the document rather than the caption produced by a search engine is to only consider *satisfied clicks*. Simple satisfaction classification is typically based on dwell time cut-offs, based on results from log analysis [Morita and Shinoda, 1994]. For example, a common choice is to identify the click as satisfied if the user spends at least 30 seconds on the result page [Yilmaz et al., 2014].

Learned click satisfaction metrics combine several features and information sources to obtain more accurate estimates of document-level user satisfaction. An early example is [Fox et al., 2005], which considers combinations of dwell time, scrolling behavior, and characteristics of the result document in a Bayesian network model. Hassan and White [2013] combine query (*e.g.*, query length, frequency in logs), ranking (*e.g.*, number of ads, diversity of results), and session (*e.g.*, number of queries, time in session so far) features to predict user satisfaction at the click level. They further investigate personalized models, either trained on the level of cohorts of similar users, or on the level of the individual user, and find that cohort models provide the most accurate prediction of search satisfaction. Kim et al. [2014a] propose a sophisticated query-dependent satisfaction classifier, that improves prediction of user satisfaction by modeling dwell time in relation to query topic and document complexity. Investigating contribution of dwell time to learned metrics in more detail, Kim et al. [2014b] found benefits of considering dwell time across a search trail, and measuring server-side dwell time over the use of client-side dwell time for predicting document-level satisfaction.

Table 3.2: Summary of *absolute* online evaluation metrics discussed in this survey (part I: document and ranking-level metrics).

Level	Metric	References
Document	Click-through rate (CTR)	[Joachims et al., 2007, Radlinski et al., 2008c, Deng et al., 2013]
	Dwell time (simple click satisfaction)	[Yilmaz et al., 2014]
	Other simple metrics (*e.g.*, time to first click, visit count)	[Fox et al., 2005, Kelly and Teevan, 2003]
	Learned click satisfaction metrics (*e.g.*, integrate mouse movement, per-topic reading time)	[Fox et al., 2005, Hassan and White, 2013, Kim et al., 2014b,a]
	Click behavior models (*e.g.*, cascade model, dynamic Bayesian network click model)	[Craswell et al., 2008, Chapelle and Zhang, 2009, Dupret and Liao, 2010, Guo et al., 2009a, Grotov et al., 2015]; detailed survey: [Chuklin et al., 2015]
Ranking	Click rank (variant: reciprocal rank)	[Boyan et al., 1996, Radlinski et al., 2008c]
	CTR@k	[Chapelle et al., 2012]
	pSkip	[Wang et al., 2009]
	Time to click	[Chapelle et al., 2012, Radlinski et al., 2008c]
	Abandonment (good / bad)	[Li et al., 2009, Hassan et al., 2011, Diriye et al., 2012, Radlinski et al., 2008c]
	Learned list-level metrics	[Hassan et al., 2013]

Click behavior models take this further, learning a latent relevance score for individual documents from clicks and possibly other observations. In one of the earliest click model papers, Craswell et al. [2008] propose the cascade model to describe the alternative actions that a user of a search system can perform at any point in time. While the goal of this work was to explain observed online search behavior rather than inform evaluation, it has been followed by numerous click models that have been used for evaluation. While there are now over a hundred papers on click modeling, we select a small set of representative examples.

One early click model for evaluation was proposed by Chapelle and Zhang [2009], learning a dynamic Bayesian click model based on observed actions. Alternatively, Dupret and Liao [2010] estimate the relevance of documents from log data by training a probabilistic model of observation and click action. Guo et al. [2009a] presented a different click-chain model. A detailed study of click-model inspired evaluation metrics was recently presented by Chuklin et al. [2013b, 2015], which was later extended to take estimation uncertainty into account under the Bayesian framework [Grotov et al., 2015].

3.3 Relative Document-level Metrics

It was observed by Joachims et al. [2007] that user interaction with individual documents returned by a search system is dependent of the context of other available alternatives. Hence, user actions can also be described as relative preferences among the available choices, rather than as absolute statements of document relevance. In their work, Joachims et al. proposed a relative interpretation of search engine behavior: When a user skips over a search result only to click on a lower ranked one, they are expressing a preference for the lower ranked document over the higher ranked document. Although this was used for training a search system rather than for evaluation, it was followed by a number of evaluation-focused works.

Among the first of these, Radlinski and Joachims [2006] showed how to estimate which of any two documents is more relevant to a search query. By randomizing the document presentation order in a

predefined way, every adjacent pair of documents is shown to users in both possible orders. Observing which is more often clicked when at the lower position, they showed that search users provide a preference as to which appears more relevant.

Tip for small-scale experiments #8

Randomization techniques may be particularly useful to obtain document-level evaluation data from real users.

A hybrid between absolute and relative metrics was proposed by Agrawal et al. [2009]. Here, they start with a pairwise interpretation of clicks as above. However they ground the most preferred and least preferred documents on an absolute relevance scale. This allows them to transform the preferences into absolute document relevance judgments.

3.4 Absolute Ranking-level Metrics

Moving beyond metrics that capture interactions with individual documents, a variety of metrics capture retrieval quality at the ranking level by aggregating across interactions with all documents. A summary of the document-level and ranking-level absolute online metrics discussed in this survey is provided in Table 3.2.

Click rank is the most basic such metric: It measures the position of the document selected in an IR system [Boyan et al., 1996]. Clearly, presenting clicked documents at a higher position is better, hence the lower the mean click rank, the higher the estimated performance of the IR system. However, we note that if one was to compare a ranking with just one relevant document at the top position, with a ranking that offers more relevant choices, the one with more choice would necessarily (and initially counter-intuitively) have a lower mean click rank.

Because click rank has a value between 1 and essentially an unbounded number, the most common variant is the inverse click position, usually called the *mean reciprocal rank*.

Tip for small-scale experiments #9

Beware unbounded metrics at small scales. They often have much poorer statistical properties (see Section 3.4 of Kohavi et al. [2012])

A related metric is the click rate within the top k positions, with a special case being the clickthrough rate at position 1 (that is, the CTR on the top search result). In fact, Chapelle et al. [2012] observed that, among a variety of absolute ranking-level metrics in a large-scale comparison of different evaluation approaches, this metric most reliably agreed with known experimental outcomes.

A more advanced variant of click position, pSkip, was proposed by Wang et al. [2009]. Intuitively, this metric encodes the probability of a user skipping over any result and clicking on one that is lower. The lower the probability of skipping, the higher the quality of the search ranking produced. Given the complexity of search behavior, for instance with multi-column layouts of results, the pSkip metric was further refined to model clicks with a partially observable model [Wang et al., 2010].

Time to click: Once we move beyond documents to entire rankings, the overall time from the results of a search query being shown to further user interaction with the search engine have been shown to reflect search engine quality. As time spent is the key cost to search system users, reducing this time is considered good (*e.g.*, [Chapelle et al., 2012]). Variants include time to first click and time to last click (*e.g.*, [Fox et al., 2005]).

Interestingly, as search systems have improved in recent years, the user interactions on which such online metrics are based — clicking — is increasingly often no longer required. This means that click-based metrics have become less effective when IR systems produce search results that are so good that the user does not need to interact further. This has led to two key improvements.

Abandonment: First, it is recognized that lack of interaction (usually termed *abandonment*) and dissatisfaction are not necessarily equivalent. In a seminal work on this topic, Li et al. [2009] observed

the existence of good abandonment that captures satisfaction without users needing to interact with the search system. Hassan et al. [2011] more recently presented a more comprehensive machine-learning approach to better account for good abandonment, resulting from more sophisticated search result page elements such as factoids and entity panes. Diriye et al. [2012] presented an analysis of *why* users choose to abandon in different cases.

The need to recognize when abandonment is in fact good has led to the second improvement of considering follow-on actions beyond the current search query. This is discussed in Section 3.6 on absolute session-level metrics.

Learned metrics at the absolute rank level have historically been less widely investigated than learning absolute user satisfaction at either the click (Section 3.2) or session level (Section 3.6). An exception is work by Hassan et al. [2013], who propose to condition ranking level user satisfaction on what happens *next*. They develop a satisfaction model that takes into account follow-on query reformulations to decide whether the interactions with results for the previous query was indicative of success or not.

3.5 Relative Ranking-level Metrics

While absolute ranking metrics produce a fixed numerical score for a given information retrieval system, we must remember that this score is computed in a particular context: At a given time, with a particular audience, who have time-specific information needs while interacting with the IR system through a particular technology and interface. To control for this, systems that we wish to compare tend to be evaluated simultaneously in an AB test.

However, in many cases the experimenter's goal is to determine which of two system for ranking documents in response to a user query produces better rankings on average. A natural way to answer this question would be to perform an absolute measurement of each system, and compare the values obtained. Yet it can be argued that this approach solves a more difficult problem (absolute measurement)

to obtain a simple preference. In this section, we present an alternative called *interleaving*.

To more efficiently obtain a preference between two retrieval systems, Joachims proposed the Interleaving approach [Joachims, 2002, Chapelle et al., 2012]. The intuition behind interleaving is that rather than showing each user the results from just one of the systems, the results are combined in an unbiased way so that every user can observe results produced by both systems. In this way, if one system returns better results on average, users can select them more often. This contrasts with absolute metrics in that users of one system have no way of knowing what else might be returned by the other system, and thus are likely to simply select the best alternative in both cases leading to user behavior on the two systems being less distinguishable.

Formally, interleaving can be written as consisting of two parts: (i) a *mixing* policy $\psi : R_A \times R_B \rightarrow R_I$ that takes two permutations of documents (*i.e.*, the output of two ranking systems) and produces a third permutation that is shown to users (*i.e.*, a combined ranking), and (ii) a *scoring* rule $\Delta : R_A, R_B, R_I, O \mapsto \mathbb{R}$ that takes observations O of user behavior and returns a real-valued score that indicates the degree of preference for one system over the other. This scoring rule also needs the original rankings, as well as the combined ranking, to compute the preference.

Tip for small-scale experiments #10

If interleaving is practical in your setting, it is particularly well suited to be run for smaller scale studies.

3.5.1 Formal Presentation

We now present the Team Draft interleaving algorithm [Radlinski et al., 2008c] in detail, followed by a brief description of a few alternatives. However, we refer the reader to Chapelle et al. [2012] for a more detailed discussion of the specifics.

As stated above, interleaving algorithms consist of two parts: A mixing policy, and a scoring rule. Team Draft interleaving uses a simple

constructive iterative mixing algorithm, operating two documents at a time: At each step, each input ranking (R_A and R_B) selects one document to append to the combined ranking. Specifically, each ranker selects the highest ranked document that is not yet in the combined list. To enable scoring, Team Draft interleaving also maintains a record of which document was added at which step, as represented by sets T_A and T_B in Algorithm 1.

Algorithm 1 Team Draft Interleaving : Mixing Algorithm

1: $k \leftarrow 0$, $R_I \leftarrow \emptyset$, $T_A \leftarrow \emptyset$, $T_B \leftarrow \emptyset$
2: **while** $k < N$ **do**
3: **for** $r \in permutation(A, B)$ **do**
4: $d^+ \leftarrow$ top-ranked$_{d \in R_r}\, d \notin R_I$
5: $R_I[k+1] \leftarrow d^+$
6: $T_r \leftarrow T_r \bigcup d^+$
7: $k \leftarrow k + 1$
8: **end for**
9: **end while**

Given an interleaved ranking R_I and team assignments T_A and T_B, let O be the observed set of documents that the current user of the ranked retrieval system clicks on. In the simplest form, the Team Draft scoring rule Δ produces a preference for R_A if $|T_A \bigcap O| > |T_B \bigcap O|$, a preference for R_B if $|T_A \bigcap O| < |T_B \bigcap O|$, and a preference for neither otherwise.

Extensions of Team Draft interleaving have been proposed, e.g., replacing the loop in Step 3 with a sampling algorithm [Hofmann et al., 2011b], extending the algorithm to an arbitrary number of input rankings [Schuth et al., 2014], and more sophisticated approaches to both mixing and scoring as described in Section 2.7.

Of particular note, this algorithm is *constructive*, as it constructs an interleaved list step by step. As such, analysis of the properties of the mixing and scoring algorithms requires careful consideration of all possible outcomes. As shown by Hofmann et al. [2011b], this leads to potential pitfalls. As a result, Radlinski and Craswell [2013] described an alternative *optimized* approach to evaluation algorithm

design. They proposed to start with a set of formal properties of the mixing and scoring algorithms, then solve for a distribution over mixes $\{p_i, \psi_i\}_{i=1...k}$ which determines the rankings users are presented with. They suggested that these properties should be:

1. Each mixing $\psi_i \in \Psi$ is shown with a probability $p_i \in [0, 1]$ where $\sum_i p_i = 1$.
2. The expected score from observations of a user who does not consider document relevance (*i.e.*, the distribution of clicks is independent of relevance) must be zero[1]: $E_{\text{random user}}[\Delta] = 0$.
3. If only one document d is selected by the user, the scoring function Δ prefers R_A if only and only if R_A ranks document d higher than R_B.
4. The sensitivity of the algorithm is maximized.

In their formulation, the set of allowable mixes $\psi \in \Psi$ is constrained to be such that any prefix of $\psi(R_A, R_B)$ contains all documents in some prefix of R_A and all documents in some prefix of R_B, a generalization upon the rankings that may be produced by the constructive Team Draft algorithm and most of its variants.

An extension of some of these criteria is proposed in Hofmann et al. [2013c]. For example, they extend criterion 3 (above) to the notion of Pareto dominance: a ranker R_A is to be preferred if its ranking of clicked documents Pareto dominates R_B. Alternatively, Kharitonov et al. [2013] propose the use of historical data to optimize sensitivity of interleaved comparisons.

3.5.2 Validation

Interleaving has been directly compared to both AB online evaluations, and traditional Cranfield-style judgment-based IR system evaluation. Research has showed that interleaving most often agrees with other online [Radlinski et al., 2008c, Chapelle et al., 2012] and offline evaluation metrics [Radlinski and Craswell, 2010, Chapelle et al., 2012],

Table 3.3: Summary of *relative* online evaluation metrics discussed in this survey. Note that papers on interleaving are split between Chapter 2 (focus on the interleaving algorithm) and Chapter 3 (focus on click scoring). Here, we include both.

Level	**Metric**	**References**
Document	Click-skip	[Joachims et al., 2007]
	FairPairs	[Radlinski and Joachims, 2006]
	Hybrid relative-absolute approach	[Agrawal et al., 2009]
Ranking	Interleaving (simple click scoring)	[Joachims, 2002, Radlinski et al., 2008c, Chapelle et al., 2012, Hofmann et al., 2011b], [Hofmann et al., 2012b, 2013c] (reuse of historical data), [Chuklin et al., 2013a, 2014] (consider vertical results), [Schuth et al., 2014, 2015a] (multileave – compare several rankers)
	Interleaving (learned / optimized click scoring)	[Yue et al., 2010a] (optimize scoring rule for sensitivity), Hofmann et al. [2012a] (reduce presentation bias), [Radlinski and Craswell, 2013, Kharitonov et al., 2013] (optimize list construction for sensitivity), [Schuth et al., 2015b] (optimize scoring rule for agreement with absolute metrics), [Kharitonov et al., 2015b] (optimize scoring and generalize to grids of results)

with cases of disagreement most often favoring interleaving, as seen in a number of examples presented by Radlinski and Craswell [2010]. This body of work also showed that interleaving exhibits one to two orders of magnitude of improved sensitivity over most absolute online evaluation techniques [Radlinski et al., 2008c, Chapelle et al., 2012].

However, when comparing retrieval systems with similar performance, disagreements between different metrics are common. Schuth et al. [2015b] showed that this also applies when comparing systems with different online evaluation techniques, such as any given absolute metric as compared to an interleaved evaluation. While Schuth et al. [2015b] presented one technique to optimize interleaving towards an online absolute metric taken as a gold standard, identifying true user satisfaction in both absolute and relative setting continues to be an unsolved problem.

3.6 Absolute Session-level and Longer-term Metrics

It is reasonable to assume that for complex search tasks, a user's needs may not be satisfied by a single query. In the case of difficult information needs, it may even be unreasonable to expect that the user's first query is effective. Even more, in the case of exploratory search the user's goal is to learn about a topic as they search. We review session and longer-term metrics below and give an overview in Table 3.4.

Simple session-level metrics: relatively simple session-level metrics such as the number of queries per session have been long used. For example, Song et al. [2013b] measure session level performance using *e.g.*, unique queries per session and session length. Chapelle et al. [2012] compare ranking level (interleaving) metrics to queries per session and time to first / last click.

Work in session-level online evaluation metrics reflects the recognition that a more holistic view of user satisfaction may be required. One reason is that these metrics may behave counterintuitively. For instance, Song et al. [2013b] recently studied the interaction between short-term and long-term metrics if search system quality is degraded, finding that degradation in retrieval quality may appear to increase

user engagement in the short term (as users need to work more to find good results), while decreasing it in the long term (as users are generally dis-satisfied and may stop using the system). Similarly, Chapelle et al. [2012] find simple absolute session-level metrics to be unreliable.

Learned metrics combine several session-level and lower-level user actions to obtain reliable estimates of search success. Note that *session*, *task*, and *goal* level are often used as roughly synonymous, depending on whether a study was based on logs (where extracting session is easier than identifying tasks) or in a laboratory (where study participants complete one task or search goal at a time). An early study of implicit satisfaction indicators is by Fox et al. [2005]. The authors consider a combined model that includes *e.g.*, query count, average dwell time, the number of results visited, and the type of action used to end a session. Ageev et al. [2011] propose both a session-level model of search success, and a game interface for collecting training data for such a model.

Several works found that incorporating more fine-grained information, *e.g.*, about the user's behavior or their context, can substantially improve accuracy of session success prediction. For instance, Guo and Agichtein [2012] consider models that incorporate mouse movement (as users may, for example, hover the mouse cursor over relevant text) and other user behavior. Chen et al. [2015] investigate effects of heterogeneous result pages in the form of verticals, and find that incorporating features that reflect types of verticals, along with fine-grained eye-tracking features, improves accuracy of success prediction. Apart from improving session-level metrics, Zhang et al. [2011] show that session or task-level features can also help improve the accuracy of click behavior models (cmp. click models in Section 3.2).

An interesting question concerns whether session-level satisfaction is more than the sum of its parts, *i.e.*, more than a simple aggregate of document-level satisfaction. Hassan et al. [2010] show that directly learning task satisfaction provided more accurate results than aggregating document-level ratings. More recently, Wang et al. [2014] propose a model that integrates document and session-level satisfaction prediction, by modeling document-level satisfaction as latent factors in a hierarchical model.

Table 3.4: Summary of *absolute* online evaluation metrics discussed in this survey (part III: session-level or higher – typically user-level).

Level	Metric	References
Session	*Simple session metrics* (*e.g.*, queries per session, session length, time to first click)	[Chapelle et al., 2012, Song et al., 2013b]
	Searcher frustration, struggling	[Feild et al., 2010, Odijk et al., 2015, Arkhipova et al., 2015b]
	Learned metrics (session or task success)	[Fox et al., 2005, Hassan et al., 2010, Guo and Agichtein, 2012, Wang et al., 2014, Chen et al., 2015, Jiang et al., 2015c]
User	Absence time	[Dupret and Lalmas, 2013, Chakraborty et al., 2014]
	Engagement periodicity	[Drutsa et al., 2015b]
	Other loyalty metrics (*e.g.*, queries per user, daily sessions per user, success rate per user	[Deng et al., 2013, Song et al., 2013b, Kohavi et al., 2012]

Predicting search satisfaction has typically been formulated as a binary classification task. However, recent work by Jiang et al. [2015c] shows that finer-grained types of satisfaction can be learned effectively, and gives insight into a rich structure of user interaction and search satisfaction.

We may also consider success separately from frustration: Whether satisfying an information need was more difficult than the user considers it should have been, as studied by Feild et al. [2010]. More recently, Odijk et al. [2015] propose an approach to detect whether users are struggling in their current search. Alternatively, search engine switching can be used to discover when users fail in their search [Arkhipova et al., 2015b].

Loyalty metrics: A common long-term goal (especially for commercial search systems) is for users to repeatedly engage with a search system. These are typically computed at the user level. Dupret and Lalmas [2013] proposed to measure how long it takes until a user returns to a question answering system as a metric of overall user satisfaction. They model this long-term engagement using statistical tools from survival analysis. A follow-up study by Chakraborty et al. [2014] extended this work to other types of search systems. The use of engagement periodicity as the basis for online evaluation metrics is explored in [Drutsa et al., 2015b].

Other user-level metrics that have been used in previous work include queries per user (presumably users who find a system effective will engage with it more), sessions per user (or sessions per user per day; to measure unique information needs for which users turn to the system) [Kohavi et al., 2012], daily sessions per user, success rate per user [Song et al., 2013b], and clicks per user [Deng et al., 2013]. For systems beyond search (such as social networking sites), a common metric is daily active users divided by monthly active users (commonly called DAU/MAU).

Although these loyalty metrics are important for measuring long-term success, they usually change slowly as users establish habits, making them difficult to apply as experimental criteria.

3.7 Relative Session-level Metrics

To the best of our knowledge, there is no work on relative session-level metrics. Part of the difficulty is that sessions are, by definition, generated during interaction between the user and a search engine. Consider, for example, an interleaving analogue for session-level comparisons. When we try to compare two engines in terms of session-level metrics, it is difficult to create a session that blends the two engines to produce a final session, as done by interleaving for list-level metrics. It is an interesting open problem to design a reliable way to compare session-level metrics between two search engines.

3.8 Beyond Search on the Web

One of the key recent shifts in information retrieval is that different entry points for user search — such as mobile phones or voice-powered personal assistants — change the inherent interactions between users and an IR system. Song et al. [2013a] analyze characteristics of user behavior, such as dwell time, across desktop, mobile, and tablet user. Finding substantial differences, they argue that online metrics need to be adapted to these contexts. Similar insights have led to work that focuses on interactions other than clicks, such as touch-based interaction with mobile devices [Guo et al., 2011, 2013a], and extensions of the concept of abandonment to mobile devices [Williams et al., 2016].

Recently introduced personal assistants such as Siri, Cortana, and Google Now create new challenges in online evaluation. First approaches to evaluating these using log data are explored in [Jiang et al., 2015a]. Kiseleva et al. [2016] find that high-accuracy prediction of search success is possible with the introduction of new types of features, such as acoustic and touch features.

3.9 Practical Issues

As we have seen in this chapter, a wide variety of online metrics are available for the evaluation of IR systems. This leaves the experimenter with the difficult task of identifying the correct metric for evaluating

a given search system. Often, a key solution is to rely on more than one online metric simultaneously. As each of the metrics presented above has at least some blind spots, ensuring that a search system is evaluated with multiple metrics makes it less likely that it is simply exploiting a weakness of a single metric. For instance, most published analyses of practical IR systems report a variety of online metrics. More difficult cases involve direct disagreements. For instance, where the amount of engagement may be anti-correlated with the time it takes users to find relevant content. Depending on the setting, either of these measures may be indicative of a better IR system.

This also leads to the question of how new online metrics can be designed and validated. The most common approach relies on the comparison of ranking systems of known relative quality online with a variety of metrics. Such known validation can be constructed either using less performant systems as baselines (for example older systems [Radlinski and Craswell, 2010, Chapelle et al., 2012], or intentionally performing per-query degradation such as randomization [Radlinski et al., 2008c]), or systems measured with offline relevance judgments tailored specifically to the task at hand. This allows general validation of a new metric.

A framework for considering online metric validation was proposed by Radlinski and Craswell [2010] where the goals are fidelity and sensitivity. Hofmann et al. [2013c] rather formulated it as fidelity, soundness and efficiency. These goals can be applied to the validation of any online metric, guiding the process of establishing if a new metric should be trusted.

A different practical issue is to do with *user averaging*. While it may seem natural to use query-level experimental units for list-level metrics it is worth considering the effect of heavy users: Suppose we average performance per query, yet there is one user (or small set of users) who is much more active. Per-query averaging would mean this single user has more influence on future online evaluations. Particularly in cases where there may be a small number of malicious or incorrectly identified users (*e.g.*, web crawlers), double-averaging is often preferred: Average metrics over the experimental unit (queries, sessions, etc), then average experimental units per user to give each user identical total impact.

Tip for small-scale experiments #11

Carefully consider the averaging used in smaller scale studies, as a handful of users may very easily dominate results.

Another issue that is not frequently discussed is how to aggregate single metrics within an online evaluation. Metrics can be designed at the query, session, user, or system level, and then they can again be aggregated at these various levels (or an average of averages can be used). Chapelle et al. [2012] has some discussion of this, mainly for interleaving experiments, but there is no consensus in the published literature.

Finally, it is common in practice that multiple metrics are of interest at the same time. For example, a search engine may have to face a trade-off between relevance and revenue of results it shows on a page. One common approach is to optimize one metric while fixing others at a certain level [Radlinski et al., 2008a, Agarwal et al., 2011]. Another option is to linearly combine these metrics into a single one [Bachrach et al., 2014], although it is not always obvious how to set the coefficients in the linear combination.

4

Estimation from Historical Data

While this article focuses on online evaluation for information retrieval, it is natural to ask whether behavioral data collected online for an IR system can be reused to evaluate new algorithms, in particular to estimate the online performance they would achieve. This chapter shows how under some conditions this can be done.

4.1 Motivation and Challenges

Until now, we have been discussing how information retrieval systems can be compared using online metrics in controlled experiments. However, this general approach is expensive for several reasons. First, to ensure the natural usage environment, controlled experiments are run on real users. This means that a system that performs worse than expected would lead to a negative user experience that may, for instance, cause users to stop using the system entirely. Second, production environments for information retrieval are typically highly complex and optimized, meaning that substantial engineering effort is often needed to take a technique that might improve result relevance, and make it robust and fast enough to work reliably in a production environment

before a controlled experiment can be considered. Discovering that a technique is ineffective after such a large investment hurts experimental agility. Third, due to daily, weekly and even seasonal variation of online behavior and thus online metrics, controlled experiments may take days or even weeks to yield reliable generalizable results. This means that even for systems with high traffic volume there is a limited amount of experimentation that can be performed. As a consequence, controlled experiments in IR often involve a long turnaround time, substantial engineering resources and opportunity costs.

Such limitations do not exist with *offline evaluation*: instead of running an experiment on live users, one aims to estimate the quality of a system using historical data. If offline evaluation could be done in a reliable way, system evaluation would only require a *static* dataset. This would mean that such *offline experimentation* can be done much faster, therefore improving experimentation agility and allowing one to test potentially many more ideas for system improvement.

Using historical data to compare two systems is not a new idea. In fact, it has been used for *offline evaluation* of retrieval systems for half a century [Cleverdon, 1967]. It is also a standard approach in areas such as supervised learning [Asuncion and Newman, 2007, Deng et al., 2009], and in particular supervised learning to rank [Qin et al., 2010, Chapelle and Chang, 2011], where new algorithms are typically evaluated on benchmark datasets. Such a dataset normally consists of feature-label pairs, where each item (in our case, query-document pair) has its features computed ahead of time and is also labeled in terms of correct score (in our case, relevance). Given such scores, standard offline metrics such as the average accuracy or rank-based relevance metrics such as NDCG [Järvelin and Kekäläinen, 2002] can be easily computed for any given classifier, regressor, or ranking function.

We address a subtly different challenge, namely to estimate changes in *online metrics* from historical data previously collected. Rather than aiming to produce a purely offline evaluation pipeline, the goal is to increase experimental agility and filter out the poorer candidates before resorting to the gold standard of online evaluation. The key difference between using labeled data for offline evaluation and using

historical data for estimating online metrics, is the problem of *partial observability*. Offline labels are typically collected for a representative set of possible items (in supervised machine learning this is typically a random sample of problem instances), or documents (various pooling approaches have been investigated in the IR literature). The degree to which such a set is representative determines the reliability of the computed offline metric.[1] In online evaluation, the experimenter typically only observes "labels" that reflect users' responses to the currently deployed system. To evaluate a new variant of the system, the experimenter needs to reason about how users *would have responded* if the alternative system had been used — a scenario for which data was not actually collected. This problem is also referred to as *counterfactual reasoning*. See the next section (Section 4.2) for the formal problem definition.

To understand the challenges that need to be addressed when estimating online metrics from historical data, consider the following example. Suppose we wish to estimate CTR@1 (click-through rate of the top-ranked items) of a new ranking algorithm for news recommendation (as in Chapter 2). A naïve approach could estimate online performance of the new algorithm from data logged while the old algorithm was used for ranking. For example, it could assess the rate at which newly top-ranked items were clicked according to the logged data. However, the resulting estimate would likely be severely biased: If the new ranking algorithm presents new items at the top position (or ones that were shown at low ranks previously), users may never have clicked them before simply because they did not notice these items. This would lead the naïve approach to under-estimate online performance of the new algorithm. While we might imagine a hand-tuned click re-weighting approach reducing the bias of CTR@1, our goal in this chapter is to consider general purpose approaches that are metric agnostic.

The difficulty in estimating online metrics thus comes from metrics depending on user feedback, which in turn is affected by system

[1]For a more rigorous discussion of these issues, see Hastie et al. [2009] for a supervised machine learning perspective or Voorhees [2014] for an IR perspective. In general, the challenging of handling unrepresentative samples is related to the covariate shift problem [Quiñonero-Candela et al., 2008].

output. This is an example of the more general statistical problem of estimating causal effects from historical data [Holland, 1986]. The general formulation is that one aims to infer, from data, the average effect on some measurement (metric) by changing the system (often known as "intervention" in the statistics literature). Therefore, our problem of estimating online metrics of a system is equivalent to estimating the causal effect on the metric if we choose the intervention (running the system online). Other related areas in machine learning include off-policy reinforcement learning [Precup et al., 2000] and learning in the presence of covariate shifts [Quiñonero-Candela et al., 2008].

Solving this estimation problem is hard, as shown in the example above. In fact, in many situations it is known to be impossible [Langford et al., 2008]. For example, estimating online metrics from logged data is trivially impossible if the actions chosen by system used for logging have no overlap with those of the system we aim to evaluate. This typically happens with deterministic systems, and is often hidden in systems that take actions (deterministically) based on context information. One therefore has to impose conditions on the data collection process to allow the use of reasonable estimation methods.

Since the problem of estimating online metrics from historical data is not unique to IR, this chapter also reviews work in related Web applications like online advertising, and will also draw connections to important results in related literature (especially statistics).

4.2 Problem Setup

As described in Chapter 3, most metrics can be expressed as the expectation of a certain measurement averaged over experimentation units. *E.g.,* click-through rate is the click-or-not binary signal averaged over all search pages, and time-to-first-click is the average amount of time to observe the first click from a user on a search result page. In both examples, an experimentation unit is a search result page, although in general it can refer to a user, a search session, a query, or other units.

Formally, let $\mathcal{X}$ be the set of experimentation units, $\mathcal{A}$ the set of actions that an IR system chooses from. Each action is a possible

output of the IR system for a particular experimentation unit. A *policy* is a function π that maps $\mathcal{X}$ to $\mathcal{A}$. When action A is chosen for experimentation unit X, a numerical reward signal Y is observed. For example, if $X \in \mathcal{X}$ is a query, $\mathcal{A}$ could be the set of permutations of documents on the first search result page given X, and π is a ranking function that produces one such ordered list $A \in \mathcal{A}$. Here, we consider the general case where π a randomized function: slightly abusing notation, we use $\pi(\cdot|X)$ to denote the conditional distribution of actions π selects for unit X. While most production policies are deterministic in practice, randomized ones can be useful, for example, in probabilistic interleaving experiments.

The definition of reward depends on the quantity we are interested in; it can be a binary signal indicating if there is a click or not, the time to get the first click, and so forth. Clearly, Y is in general a random variable, whose distribution depends on both X and A. The quantity we try to estimate, the *value* of a given particular policy π, $v(\pi)$, is the average reward we obtain by running π:

$$v(\pi) := \mathbf{E}_{X,\pi}[Y]\,, \tag{4.1}$$

where the expectation is taken with respect to X and $A = \pi(X)$. Therefore, $v(\pi)$ is an aggregated measure of how much reward is obtained across all units. If Y is a binary click-or-not signal, $v(\pi)$ is simply the overall click-through rate. Similarly, if Y is the continuous-valued time to first click in a search event, $v(\pi)$ is the *average* time to first click. Other reward definitions lead to similar interpretations of $v(\pi)$.

In the formulation above, two stochastic assumptions are made:

- The units X are drawn IID from an unknown distribution μ; and
- The reward Y is drawn IID from an unknown distribution, conditioned on X and A.

Whether these assumptions hold in reality is problem-dependent. In many IR applications, however, the first assumption appears to hold to some extent, with a proper choice of $\mathcal{X}$. For example, users visiting a search engine are largely independent of each other. The second assumption may require more justifications, given common seasonal

variations of user behavior [Kleinberg, 2004]. Fortunately, most of the techniques below apply without change to the more general situation where Y may be non-stationary.

The most straightforward way to estimate $v(\pi)$ is to run π for a long enough period, measure rewards while the policy is running, and then average the rewards. The law of large numbers ensures that the empirical average will eventually converge to $v(\pi)$. In fact, this is the idea behind *online* experimentation (Chapter 2). In the *offline* case, where we try to estimate $v(\pi)$ from historical data, the problem becomes more challenging.

For most problems, historical data $\mathcal{D}$ can be written as triples of the following form:

$$\mathcal{D} = \{(X_i, A_i, Y_i)\}_{i=1,2,\ldots,m} \tag{4.2}$$

where $X_i \sim \mu$, Y_i is the reward signal after taking some action A_i for experimentation unit X_i. Such a dataset is *partially* labeled (or has missing values), in the sense that it only contains reward information for actions that were actually chosen, not other actions. Therefore, for unit X_i, if the policy π chooses an action $A_{\text{new}} = \pi(X_i)$ that is different from the one in the data, A_i, then one has to infer what the reward signal *would have been* if A_{new} were chosen for unit X_i. Answering such what-if questions requires addressing the *counterfactual* nature of the problem. This chapter surveys an array of solutions for this problem.

4.3 Direct Outcome Models

A natural way to address the counterfactual issue in estimating $v(\pi)$ is to directly estimate the reward given a context and action. If such a reward can be accurately predicted, we can essentially fill in all the missing rewards in the data set, which can be used to estimate $v(\pi)$ for any policy π. In the literature, such an approach is sometimes referred to as a direct method [Dudík et al., 2011] or a model-based method [Jiang and Li, 2016].

Specifically, from the data set $\mathcal{D}$ in Equation 4.2, we may construct a supervised-learning dataset of size m,

$$\mathcal{D}_{\text{DOM}} := \{(X_i, A_i) \mapsto Y_i\}_{i=1,2,\ldots,m} \tag{4.3}$$

which is used to "learn" an outcome model, denoted $\hat{f}$, such that $\hat{f}(x,a) \approx f(x,a) := \mathbf{E}[Y|X=x, A=a]$ for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$. Then, the policy value can be estimated from $\mathcal{D}_{\text{DOM}}$ by:

$$\hat{v}_{\text{DOM}} := \frac{1}{m} \sum_i \hat{f}(X_i, \pi(X_i)) \,, \tag{4.4}$$

when π is deterministic, and

$$\hat{v}_{\text{DOM}} := \frac{1}{m} \sum_i \sum_{a \in \mathcal{A}} \pi(a|X_i) \hat{f}(X_i, a) \,, \tag{4.5}$$

for randomized π in general. To avoid overfitting, $\hat{f}$ should be learned on a separate data set other than $\mathcal{D}_{\text{DOM}}$.

In the context of IR, this model often has the natural interpretation of a user model, say a click model [Chapelle and Zhang, 2009, Guo et al., 2009a] or a browsing model [Dupret and Piwowarski, 2008]. Although most existing click models have separate parameters for individual queries, more general regression techniques can be applied, such as linear models, decision trees, and neural networks [Hastie et al., 2009]. On the other hand, the outcome model may simply be an empirical average computed from data, without having to build more complex user models. For example, Li et al. [2015b] estimated $\hat{f}(x,a)$ for query x and ranked document list a simply by averaging observed rewards Y in the subset of data where $X_i = x$ and $A_i = a$; this approach does not generalize, so works best when $|\mathcal{X}|$ and $|\mathcal{A}|$ are relatively smaller compared to the data size m.

As an alternative to learning a direct estimator from data, a simulator may be constructed based, for example, on expert knowledge of user behavior. For example, Hofmann et al. [2011a, 2013b] develop such a simulation approach by combining manually annotated IR test collections with insights into user behavior gleaned from click models such as the Dependent Click Model [Guo et al., 2009b], and use it to evaluate online learning to rank algorithms. Chuklin et al. [2014] take a similar approach to click simulation on interleaved ranking results. Finally, click models can be used to extract useful information which, together with other sources of information, can be used to predict the outcome of an online experiment [Kharitonov et al., 2015a].

A major benefit of the direct outcome method is its flexibility. Once a reasonably good model is built, it can be used to simulate essentially all online experiments, including evaluating policies that depend on the history (such as online-learning algorithms mentioned above [Hofmann et al., 2013b]). It can also be run multiple times, making it easy to generate confidence intervals and other statistics of interest.

On the other hand, the estimation quality of the direct outcome method depends critically on the accuracy of the outcome model. Furthermore, the error does not necessarily diminish to 0 even if infinite data are available to fit the model $\hat{f}$. This happens when the class of regressors (like generalized linear models) is not expressive enough to fully model the true reward function, f. For example, if an over-simplistic user model is adopted, we do not expect the model to accurately mimic real user behavior, even if infinite data are available to fit any unknown parameters in the model. As a consequence, the model $\hat{f}$ returned by any regression method has to trade off prediction errors in different regions of the evaluation space $\mathcal{X} \times \mathcal{A}$. Hence, if π tends to select actions that are under-represented in $\mathcal{D}_{\mathrm{DOM}}$, the estimation error, $|\hat{v}_{\mathrm{DOM}}(\pi) - v(\pi)|$, can be (arbitrarily) large, no matter how many historical observations are available.

Despite the above limitations, the direct outcome method remains a reasonable solution in practice, especially for *comparing* ranking algorithms (*e.g.*, Hofmann et al. [2011a, 2013b], Chuklin et al. [2014]). It can also be combined with another approach to yield an algorithm that is often better than both, as we discuss next.

4.4 Inverse Propensity Score Methods

An alternative to inferring user behavior based on evaluation of existing ranking techniques is to take advantage of the control available to online systems: instead of simply running an existing search engine, the engineers may be able to change the behavior of the search engine so that the data it collects may be useful for *future* evaluations of *other* rankers. Such data collection, typically called *exploration data*, may intentionally introduce random modifications ("exploration") to the output of a system, and records the corresponding user feedback.

The methods surveyed in this section assume access to such a *randomized* data set where, given an experimentation unit X, a random action is chosen according to a known conditional distribution, $p(\cdot|X)$. This gives rise to a randomized dataset:

$$\mathcal{D}_{\mathrm{R}} = \{(X_i, A_i, Y_i, P_i)\}_{i=1,2,\ldots,m}\,,$$

which is the same as $\mathcal{D}$ with the only addition of propensity scores:

$$P_i = p(A_i|X_i)\,.$$

Note that p is selected by the experimenter, and is thus known.

4.4.1 IPS Estimators

In this section, we will assume that $p(\cdot|x)$ assigns non-zero probabilities to all actions in every context x. The Inverse Propensity Scoring (IPS) approach is based on the following key observation: for any $i \in \{1, 2, \ldots, m\}$ and any $a \in \mathcal{A}$,

$$\mathbf{E}_{A_i \sim p}\left[\frac{\mathbf{I}\{a = A_i\}}{P_i} Y_i\right] = f(X_i, a)\,,$$

where $\mathbf{I}\{C\}$ is the indicator function that evaluates to 1 if C is true and 0 otherwise. In other words, the newly defined random variable,

$$\hat{Y}_i(a) := \frac{\mathbf{I}\{a = A_i\}}{P_i} Y_i\,,$$

is an *unbiased* estimate of the unknown quantity $f(x_i, a)$, even if a is counterfactual (that is, if $a \neq A_i$). This technique, also known as importance sampling, has been used in other statistical and machine-learning algorithms (*e.g.*, Auer et al. [2002]); see Liu [2001] for a survey.

The observation above immediately implies unbiasedness of the following IPS estimator with a possibly randomized policy π:

$$\hat{v}_{\mathrm{IPS}}(\pi) := \frac{1}{m} \sum_i \frac{\pi(A_i|X_i)}{P_i} Y_i\,. \tag{4.6}$$

If π is deterministic, the above can be slightly simplified as:

$$\hat{v}_{\mathrm{IPS}}(\pi) := \frac{1}{m} \sum_i \frac{\mathbf{I}\{\pi(X_i) = A_i\}}{P_i} Y_i\,. \tag{4.7}$$

One problem with the above estimator is that its variance is unbounded if the ratio $\pi(A_i|X_i)/P_i$ increases to infinity. An easy fix to this problem is to provide a threshold, $W_{\max} > 1$, to control variance, at the cost of introducing a small amount of bias. This technique, known as *truncated importance sampling* [Ionides, 2008], often leads to a lower mean squared estimation error:

$$\hat{v}_{\text{tIPS}}(\pi) := \frac{1}{m} \sum_i \min \left\{ W_{\max}, \frac{\pi(A_i|X_i)}{P_i} \right\} Y_i \,. \tag{4.8}$$

An alternative solution is to make sure P_i is always sufficiently large when designing the data-collection policy.

Another variant of IPS, sometimes called weighted importance sampling, is also popular in practice:

$$\hat{v}_{\text{wIPS}}(\pi) := \left(\sum_i \frac{\pi(A_i|X_i)}{P_i} \right)^{-1} \sum_i \frac{\pi(A_i|X_i)}{P_i} Y_i \,. \tag{4.9}$$

Similar to $\hat{v}_{\text{tIPS}}$, $\hat{v}_{\text{wIPS}}$ is biased. However, since large weights (corresponding to large $\pi(A_i|X_i)$ and small P_i values) in the numerator also appear in the denominator, the variance is often greatly reduced, compared to the basic importance-sampling estimator, $\hat{v}_{\text{IPS}}$ [Liu, 2001]; such a phenomenon is sometimes referred to as *self-normalization*. Empirically, the reduced variance of $\hat{v}_{\text{wIPS}}$ often outweighs the introduced bias, thus leads to a lower mean squared error than $\hat{v}_{\text{IPS}}$. Finally, it should be noted that, under rather weak assumptions, $\hat{v}_{\text{wIPS}}$ is asymptotically unbiased, in the sense that it eventually converges to the true policy value with infinite data [Liu, 2001].

The IPS estimator and variants provide the basis for much recent work on offline estimation of online metrics in Internet applications. Bottou et al. [2013] use a variant of Equation 4.6 to predict the consequences of changes to online metrics of a complex ad system modeled as a causal graph [Pearl, 2009], such as an ad engine. Li et al. [2015a] use the IPS estimator in a query reformulation task within a search engine to evaluate online performance of new reformulation selection models. The unbiased IPS estimation technique may also be used in other situations where the action set $\mathcal{A}$ is structured. For

example, in order to predict the outcome of an interleaving experiment, one can first collect randomized interleaving data, using probabilistic interleaving [Hofmann et al., 2011b] (*c.f.*, Section 2.7), and then apply IPS to do offline estimation [Hofmann et al., 2012b, 2013c].[2] Randomization schemes other than probabilistic interleaving can be used (for instance, exploration scavenging [Langford et al., 2008]). However, care must be taken to ensure that the distribution used for data collection has covered the action space of the target distribution sufficiently well to avoid introducing bias in the final estimate.

The IPS approach is related to the exploration scavenging technique which also produces unbiased estimates of policy values under a more specialized assumption about the data. Another special case of IPS occurs when one uses uniformly random exploration, that is, when $P_i \equiv 1/|\mathcal{A}|$. This approach has been applied to estimating click-through rates in personalized news recommendation [Li et al., 2011] and user engagement metrics in federated search [Ponnuswami et al., 2011].

Although we have presented IPS for the fixed-policy case, this idea can be extended to estimate online metrics of an online-learning algorithm (a history-dependent policy). This approach [Li et al., 2011], sometimes referred to as *Replay*, requires that actions in $\mathcal{D}_{\mathrm{R}}$ are chosen uniformly at random (that is, $P_i \equiv 1/|\mathcal{A}|$). If p is not uniform, rejection sampling can be used as a preprocessing step to yield a subset of data where actions are uniformly randomly selected [Dudík et al., 2012]. The Replay method has been found useful in a number of scenarios, such as online learning-to-rank [Moon et al., 2012, Zoghi et al., 2016], click shaping in news recommendation [Agarwal et al., 2012], and advertising [Tang et al., 2013].

Finally, IPS-based estimation can also be used as a subroutine for the more challenging offline optimization problem, in which one aims to find a policy π^* from a given policy class Π with maximum value $v(\pi^*)$. Conceptually, offline policy optimization may be reduced to offline value estimation: if we can estimate the value $v(\pi)$ for

[2] While interleaving algorithms like TeamDraft (Section 3.5) is also randomized when selecting which *list* is used to contribute the next document, they normally do not randomly select which *document* in the list is chosen. This type of randomization is therefore of limited use when estimating other interleaving results offline.

every $\pi \in \Pi$, the policy with highest estimated value would be a reasonable approximation to π^*. Such an approach, as taken by some previous work [Beygelzimer and Langford, 2009, Strehl et al., 2011], ignores estimation variance, and therefore does not always produce near-optimal policies reliably. Recently, Swaminathan and Joachims [2015a] propose to use estimation variance as a regularization term to stabilize offline policy optimization. The authors coined the term *counterfactual risk minimization*, which resulted in the POEM (Optimizer for Exponential Models) algorithm for a class of structure learning problems. The authors later propose an improved approach based on weighted importance sampling [Swaminathan and Joachims, 2015b]. IPS-based estimation of interleaved comparison outcomes was used to show that reuse of historical data can accelerate online learning using interleaving signals [Hofmann et al., 2013a].

4.4.2 Variance Reduction Techniques

Despite the unbiasedness guarantee, the main drawback of the basic IPS estimator in Equation 4.6 is that its variance is high and can be potentially unbounded. Since mean squared error is the sum of squared bias and variance (for example, see [Hastie et al., 2009, Section 7.3]), a large variance directly translates into a large mean squared error.

Given a policy π to be evaluated, recall that IPS computes the average of a newly defined random variable, $\hat{Y}_i := \frac{\pi(A_i|X_i)}{P_i} Y_i$, for the ith example in the dataset $\mathcal{D}_{\mathrm{R}}$. If the data collection distribution, p, and target policy, π, are not very different, meaning that the ratio $\pi(A_i|X_i)/P_i$ is often small, the variance of $\hat{Y}_i$ is small. Otherwise, a small variance in the reward signal Y_i can be greatly magnified, introducing a large variance to $\hat{Y}_i$. So far, we have seen two variants of IPS in Section 4.4.1 that try to control the variance for smaller P_i values: one applies a threshold if P_i gets too small (Equation 4.8), the other uses self-normalization (Equation 4.9).

Below, we survey other ways to reduce variance; in addition to these methods that are specific to offline evaluation, general variance reduction techniques exist and some have been successfully applied to online metric evaluation in information retrieval systems [Deng et al.,

2013]. It should be noted that multiple variance reduction techniques can be combined to yield better results than any single one.

Tip for small-scale experiments #12

At smaller scales, variance reduction techniques become essential.

IPS with Estimated Propensity Scores

So far, we have assumed P_i in $\mathcal{D}_{\mathrm{R}}$ are known, as often one has full control of the IR system to decide what action distribution to use to collect data. Although it may sound surprising, it is possible to obtain a more accurate estimator by replacing P_i by its estimates from data, even if the true value is known [Hirano et al., 2003]. One intuition is that randomness in sampling A_i contributes to the variance of IPS estimators like Equation 4.6. By using the empirical frequencies of sampled actions in the actual data, it is possible to remove such variance, as can be seen more clearly in the single-unit case [Li et al., 2015c].

More precisely, from $\mathcal{D}_{\mathrm{R}}$ we construct the following data set

$$\mathcal{D}_{\mathrm{P}} := \{(X_i, A_i)\}_{i=1,2,\ldots,m}\,, \tag{4.10}$$

which is used to estimate the conditional probability, $p(\cdot|X = x)$. The estimate, denoted $\hat{p}$, is then used in IPS-based estimators. For example, $\hat{v}_{\mathrm{IPS}}$ becomes

$$\hat{v}_{\hat{\mathrm{p}}\mathrm{IPS}} = \frac{1}{m}\sum_{i=1}^{m}\frac{\pi(A_i|X_i)}{\hat{p}(A_i|X_i)}Y_i\,. \tag{4.11}$$

Estimating p from data $\mathcal{D}_{\mathrm{P}}$ is a typical conditional probability estimation problem, which can be solved by well-established statistical techniques such as multi-class logistic regression. This variant of IPS based on estimated propensity scores has been shown to work well in recommendation and advertising problems [Strehl et al., 2011].

In practice, there is yet another important advantage of using estimated propensity score. For many complex systems as in information retrieval, sometimes it is hard to make sure actions are indeed sampled

from the intended distribution, p, which would invalidate the use of estimators like Equation 4.6. Fortunately, one can still estimate the propensity scores, sometimes with even more accurate estimates.

Doubly Robust Estimation

The *doubly robust* (DR) technique is another improvement to basic IPS estimators. The idea is to incorporate a direct outcome model (Section 4.3) in IPS estimators, so that if the model is accurate, the DR estimator can produce much better estimates than the original IPS estimator. On the other hand, if the model is inaccurate, a DR estimate still maintains the properties of IPS.

Suppose we are given a direct outcome model, $\hat{f}$, which approximates the unknown outcome function, f. Such a model $\hat{f}$ may be obtained using any method mentioned in Section 4.3. The DR estimator is given by

$$\hat{v}_{\mathrm{DR}} := \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{a}\pi(a|X_i)\hat{f}(X_i,a) + \frac{\pi(A_i|X_i)}{\hat{p}(A_i|X_i)}(Y_i - \hat{f}(X_i,A_i))\right). \tag{4.12}$$

Intuitively, $\hat{v}_{\mathrm{DR}}$ uses $\hat{f}$ to estimate the outcome (as in DOM), and then uses IPS to correct discrepancies between these outcome predictions and actual outcomes Y_i.

In Equation 4.12, there are two estimates, one for the outcome model and one for the propensity scores. A particularly useful property of DR estimators is that, as long as one of them is correct ($\hat{f} = f$ or $\hat{p} = p$), the DR estimator remains unbiased (see, *e.g.*, Rotnitzky and Robins [1995]), which justifies the name of this estimator. In practice, one cannot expect either estimate to be accurate. However, it has been shown that DR still tends to reduce bias compared to DOM, and to reduce variance compared to IPS [Dudík et al., 2011]. Furthermore, DR can also be combined with the replay approach (*c.f.*, Section 4.4.1) to evaluate nonstationary systems [Dudík et al., 2012].

A summary of all estimators discussed in this chapter is provided in Table 4.1.

Table 4.1: Summary of estimation methods covered in this survey. The summary assumes stochastic policies, see text for simplifications for deterministic policies.

Name	Estimator	Notes	References
Direct outcome method (DOM)	$\hat{v}_{\mathrm{DOM}} := \frac{1}{m}\sum_i \sum_{a\in\mathcal{A}} \pi(a\|X_i)\hat{f}(X_i,a)$	Bias depends on $\hat{f}$ and is not easy to estimate	[Dudík et al., 2011]
Inverse propensity score (IPS)	$\hat{v}_{\mathrm{IPS}}(\pi) := \frac{1}{m}\sum_i \frac{\pi(A_i\|X_i)}{P_i} Y_i$	Unbiased, often high-variance	[Liu, 2001]
Truncated IPS	$\hat{v}_{\mathrm{tIPS}}(\pi) := \frac{1}{m}\sum_i \min\left\{W_{\max}, \frac{\pi(A_i\|X_i)}{P_i}\right\} Y_i$	Truncation trades off bias and variance	[Ionides, 2008]
Weighted IPS	$\hat{v}_{\mathrm{wIPS}}(\pi) := \left(\sum_i \frac{\pi(A_i\|X_i)}{P_i}\right)^{-1} \sum_i \frac{\pi(A_i\|X_i)}{P_i} Y_i$	Can reduce variance; asymptotically unbiased	[Liu, 2001]
IPS with estimated propensities $\hat{p}$	$\hat{v}_{\hat{\mathrm{p}}\mathrm{IPS}}(\pi) := \frac{1}{m}\sum_i \frac{\pi(A_i\|X_i)}{\hat{p}(A_i\|X_i)} Y_i$	Can reduce variance; useful when propensities are unknown	[Hirano et al., 2003, Strehl et al., 2011, Li et al., 2015c]
Doubly robust (DR) estimator	$\hat{v}_{\mathrm{DR}} := \frac{1}{m}\sum_i \Big(\sum_{a\in\mathcal{A}} \pi(a\|X_i)\hat{f}(X_i,a) + \frac{\pi(A_i\|X_i)}{\hat{p}(A_i\|X_i)}(Y_i - \hat{f}(X_i,A_i))\Big)$	Unbiased if $\hat{f} = f$ or $\hat{p} = p$; can reduce bias of DOM, and reduce variance of IPS	[Dudík et al., 2011]

4.4.3 Estimating Confidence Intervals

So far, we have surveyed a few *point estimators* to estimate $v(\pi)$, in the sense that they return a real-valued estimate but do not quantify the amount of uncertainty in the estimate. Arguably, a point estimate is of limited use unless a certain level of certainty is also available, commonly reported in the form of confidence intervals [Casella and Berger, 2001]. This subsection describes several ways to compute such confidence intervals that have been applied to offline estimation of policies.

One standard way to assess confidence is through normal approximation, based on the central limit theorem. Consider the example of IPS. The point estimate computed in Equation 4.6, $\hat{v}_{\text{IPS}}$, is an average of the random variable $\hat{Y}_i$, defined by

$$\hat{Y}_i := \frac{\pi(A_i|X_i)}{P_i} Y_i \,. \tag{4.13}$$

One can compute its sample variance by

$$\hat{\sigma}^2 := \frac{1}{m-1} \sum_{i=1}^{m} \left(\hat{Y}_i - \hat{v}_{\text{IPS}} \right)^2 . \tag{4.14}$$

Then, a $(1-\alpha)$-confidence interval can be constructed:

$$\left(\hat{v}_{\text{IPS}} - t_{\alpha/2}\hat{\sigma}, \hat{v}_{\text{IPS}} + t_{\alpha/2}\hat{\sigma} \right) . \tag{4.15}$$

Often, α takes a value of 0.1 and 0.05 (corresponding to 90% and 95% confidence levels), and the $t_{\alpha/2}$ values are 1.645 and 1.96, respectively. The calculations for other IPS variants are similar.

If the sample size m is large and if the P_i's are not too close to 0, the normal approximation above is reasonably accurate. For example, it has been used to produce useful intervals in advertising [Bottou et al., 2013] and in a Web search problem [Li et al., 2015a].

Another common approach is based on the bootstrap [Efron and Tibshirani, 1993], a general statistical technique to approximate an unknown distribution by resampling. Given a dataset $\mathcal{D}_{\text{R}}$ of size m, one can get a bootstrap sample, $\tilde{\mathcal{D}}_{\text{R}}$, of the same size by sampling with replacements. Estimators like IPS (or even a direct outcome method) are applied on $\tilde{\mathcal{D}}_{\text{R}}$ to yield a policy-value estimate, $\tilde{v}$. The above

process (of resampling followed by estimation) is repeated B times, and the $\frac{\alpha}{2}$- and $(1-\frac{\alpha}{2})$-quantiles of the bootstrapped estimates, $\{\tilde{v}\}$, constitute the lower and upper bounds of a $(1-\alpha)$-confidence interval.

The above approach, known as the percentile interval, can be improved in various ways, resulting in better bootstrap-based confidence intervals [Efron and Tibshirani, 1993, Chapter 14]. One of them, BCa, has been successfully applied in an advertising problem [Thomas et al., 2015], demonstrating more accurate interval estimates than other approaches, such as normal approximation. Furthermore, the bootstrap method can be combined with the Replay method to obtain tight confidence intervals for nonstationary policies [Nicol et al., 2014].

Finally, it is worth noting that in many situations, it suffices to obtain a *one-sided* confidence interval with the lower confidence bound (LCB) only [Thomas et al., 2015]. A common use case is to compare the LCB of a new system to a baseline system (such as the production system), to see whether the new system is likely better than the baseline. Therefore, such one-sided confidence intervals can naturally be used in a robustness test to decide whether a new system should be tested in a controlled experiment.

4.5 Practical Issues

The accuracy of offline estimates of policy values depends critically on the quality of the data available to the estimators. A general requirement is that the data needs to be *exploratory* enough, so that it covers the whole $\mathcal{X} \times \mathcal{A}$ space reasonably well. Otherwise, there would be little or no information to estimate $v(\pi)$ for actions chosen by π that are not sufficiently represented in data. The need for exploration data is even greater for IPS-based approaches, as they rely directly on the (estimated) propensity scores.

Designing exploration distributions for data collection is not a trivial task, and has to balance two conflicting objectives. On the one hand, we would like to explore more aggressively in order to collect more exploratory data that are better for offline estimation. On the other hand, having too much exploration may potentially hurt

the current user experience, as the system behaves more randomly. Finding the right balance is almost always problem-dependent.

In some cases, exploration is not very expensive, for example when few actions have a strong negative impact on the user experience. Then, one can employ more aggressive exploration strategies, even the most extreme one of uniform random exploration (that is, $p(a|x) \equiv 1/|\mathcal{A}|$). Such data has maximum exploration and has been proved useful in personalized news recommendation [Li et al., 2010, 2011, Agarwal et al., 2012] as well as a recency search problem [Moon et al., 2012].

A more common scenario is that exploration is expensive, so that it makes sense to use a more conservative distribution to collect data. An effective way is to add a certain amount of randomization to a baseline (say, production) system which is known to work well. Then, by controlling the degree of randomization, one can easily control the risk and cost of exploration. Such an approach has worked well in advertising [Bottou et al., 2013] as well as Web search [Li et al., 2015a].

After exploration data is collected, it is often worth verifying that the data collection works as intended before using it. Li et al. [2015a] propose a few simple-to-use tests to identify potential data quality issues, which seem to be very useful in practice.

4.6 Concluding Remarks

As demonstrated by the cited works throughout this chapter, the methods surveyed here work well in a variety of important applications. These successes motivate creation of systems that facilitate deployment of such techniques [Agarwal et al., 2016]. Despite these advances, however, several challenges remain.

The first arises when the set of actions is large. Examples of large action spaces are the exponentially large set of ranked lists of documents on a page, or simply a set of many potentially relevant documents for a query. For direct outcome models, more actions imply that the outcome function $f(x, \cdot)$ requires more data to learn in general. For IPS-based approaches, more actions usually require more aggressive exploration during data collection, and result in higher variance in offline estimates.

Dealing with large action spaces is inherently a difficult problem. Existing solutions are all problem specific. One common approach is to reduce the space of exploration, by taking advantage of structural assumptions, such as various models for position biases in Web search [Craswell et al., 2008, Swaminathan et al., 2016], knowledge of the structure of ranked lists in Hofmann et al. [2013c], and by approximating the original problem with a "scaled-down" version [Yankov et al., 2015].

The second challenge is to estimate long-term effects of a new system. In this chapter, we have focused on the situation when user feedback, $f(x, a)$, does not depend on history. However, when an IR system changes, user behavior also evolves over time (*e.g.*, the "carryover effect" [Kohavi et al., 2012]).

There has not been much work on this problem in the literature. Bottou et al. [2013] use equilibrium analysis in online advertising to infer behavior of a new system that slightly deviates from an existing system. More recently, Jiang and Li [2016] and Thomas and Brunskill [2016] extend the doubly robust technique in Section 4.4.2 to a very related multi-step decision making problem, which can be useful for estimating session-level or long-term effects in IR systems. A similar, doubly-robust method was proposed by Murphy et al. [2001] for estimating treatment effects.

5

The Pros and Cons of Online Evaluation

This chapter discusses general pros and cons of online evaluation. In contrast to the Cranfield approach [Cleverdon, 1967], the offline, traditional and most popular way to evaluate and compare operational effectiveness of information-retrieval systems, online evaluation has unique benefits and challenges, which is the focus of this chapter.

As is made explicit in Definition 1.1, online evaluation is based on implicit measurement of real users' experiences in a natural usage environment. In modern Web search engines, interaction between users and the search engine is dominantly stored in the *search log*, which usually consists of queries, returned search result pages, click information on those pages, and landing-page information, among others, in a search session of individual users. It is often much cheaper and faster to collect such data in modern search engines, making it particularly easy to scale up online evaluation. Furthermore, one may argue that satisfaction in a natural usage environment is a more direct and truthful indication of operational effectiveness of an IR system. These reasons make online evaluation particularly attractive.

On the other hand, information in click logs, like clicks and dwell time on the landing page, is implicit in the sense that a user is

not asked explicitly whether a search session is successful or not, or whether any document on the search result page is relevant to their information need. Therefore, non-trivial effort is required to infer the *hidden* signal of whether documents satisfying the user's information need were retrieved. Furthermore, in a natural usage environment, user behavior is affected by how documents are presented on the search result page, with position bias being the most widely recognized effect. Such effects lead to the challenge of de-biasing user behavior collected in search logs when inferring session success and document relevance.

The rest of the chapter is organized into several sections that cover these issues. For a related discussion, see Chapelle et al. [2012].

5.1 Relevance

We first turn our focus to the issue of *fidelity* of implicit feedback from users: What is the gold standard of online evaluation? How do things like relevance and user satisfaction relate to online metrics? To what degree do we think user satisfaction can be estimated from observed user behavior? These questions are nontrivial to answer because of the implicit nature of signals in online evaluation.

A few studies have shown that there is not necessarily a strong correlation between offline evaluation metrics, such as mean average precision and precision at k, and online benefits for users [Hersh et al., 2000, Turpin and Hersh, 2001, Turpin and Scholer, 2006]. In particular, it is also demonstrated that there is a wide range of offline metric values which translate into an essentially flat region of online user benefits. It is therefore tempting to look for online metrics that are more aligned with user satisfaction and search success.

While it seems natural to consider a click on a document as a relevance label, such a naive approach can be problematic, as argued by, for example, Scholer et al. [2008]. One source of difficulty is the various biases present in click log, discussed in Section 5.2. To address this problem, Joachims et al. [2007] propose to use relative feedback derived from clicks, which is less prone to biases and results in a much higher agreement than absolute feedback.

Another limitation with click-through data is that they measure a user's *anticipated* relevance of the document based on snippet shown on the page. Fox et al. [2005] show that a combination with other implicit information like dwell time on landing pages gives reasonable correlation with judgments. Related recent work by Guo and Agichtein [2012] proposes to use post-click behavior to incorporate other information on the landing page, such as cursor movement and page scrolling. When session information is available, queries issued later can be a useful indicator of user satisfaction of previous queries in the session, as shown for the case of query reformulation [Hassan et al., 2013].

5.2 Biases

One of the main challenges in online evaluation comes from various biases present in typical search logs. Online behavior of users can be affected by a range of factors that are unrelated to the relevance of documents on the result page. These factors introduce bias in the user feedback recorded by a search engine.

The best-known effect is probably position bias: documents in more prominent areas of a page get more attention of a user on average, and therefore have a higher chances of being clicked on. Examples of these areas are top/left regions, as evidenced by multiple eye-tracking studies, where user attention heat maps form an F-shaped pattern or a golden triangle, among others; see Granka et al. [2004], Guan and Cutrell [2007a,b] and the references therein. Position bias has also been directly demonstrated in controlled experiments where a substantial difference in click-through rate is observed even when documents on the result pages are randomly shuffled [Moon et al., 2012]. Removing position bias, with the FairPairs algorithm [Radlinski and Joachims, 2006] for example, leads to more reliable signals of relevance.

Another factor is presentation bias. Clarke et al. [2007] investigate how click-through patterns are affected by caption features, such as the length of a snippet and whether there is a string match between query and document title. A further analysis is given by Yue et al. [2010b], who quantify the effect of bolded-keyword matching in captions

by controlling for position bias and document relevance. Therefore, one has to remove such systematic biases when using click data to infer ranker quality. For example, Hofmann et al. [2012a] propose an approach that works well in interleaving experiments, resulting in more reliable online evaluation results.

Finally, a few other sources of biases have also received attention in the literature. Joachims et al. [2007] study two types of biases: trust bias and quality-of-context bias. Trust bias exists because users believe a strong search engine tends to rank more relevant documents before less relevant ones, and therefore view/click top documents more often. The quality-of-context bias reflects that clicks on a document depend on the overall relevance quality of other documents on the result page. Buscher et al. [2010] show a similar affect of ad quality on clicks of relevant documents.

Studies of bias in click log, as well as approaches to remove these biases, remain an important research problem.

5.3 Experiment Effects

In contrast to offline evaluation, online evaluation collects data directly from users who actually use the search engine for their information needs during evaluation time. As such, with proper sampling, the distribution of users upon whom evaluation is based is representative of the actual user experience should the changes be deployed. However, an obvious practical concern is risks, namely negative impact on user experience when users are included in an online experiment. Minimally invasive approaches such as FairPairs [Radlinski and Joachims, 2006] and the related Click-based Lambdas approach [Zoghi et al., 2016], as well as interleaving (Section 3.5), are therefore particularly useful to avoid catastrophically adverse effects on user experience.

This problem is also related to the exploration/exploitation trade-off in a class of machine-learning problems known as multi-armed bandits; see Bubeck and Cesa-Bianchi [2012] for a recent survey. In the context of IR, this trade-off implies the need to intentionally vary an existing ranker to collect user click data, in the hope of discovering

signals that can be used to improve the ranker but are otherwise not available without exploration. Effectiveness of these online-learning algorithms is roughly measured by the cost of exploration needed to find optimal rankings; for example, Radlinski et al. [2008b], Moon et al. [2012], Hofmann et al. [2013b], Slivkins et al. [2013]. Motivated by interleaving experiments, Yue and Joachims [2009] propose dueling bandits where an algorithm can only learn from noisy, relative signals between two candidate rankers.

Online evaluation is also complicated by social and/or temporal changes of user behavior. For example, when a new feature is introduced or when a new ranker is deployed, it often takes some time for user click patterns to converge to equilibrium when users try to adapt to the new feature/ranker. This is sometimes called a carryover effect [Kohavi et al., 2012], implying that data collected right after the new feature/ranker is introduced is less reliable. Another example is the network effect [Ugander et al., 2013, Gui et al., 2015]. Socially connected users affect each other so that their click patterns are in general *not* independent. If one runs an online experiment to compare two rankers, extra care is needed to remove such inter-user dependency to reach statistically valid conclusions from data.

Tip for small-scale experiments #13

Beware of carryover effects in particular, they can reduce experimental sensitivity dramatically. Pre-experiment A/A tests can be used to verify their absence.

5.4 Reusability

A huge benefit of offline evaluation, such as the Cranfield approach, is reusability of data: once manually labeled data are collected, one can readily compare ranking systems on metrics like average precision and NDCG that are easily computed from the labeled data [Voorhees and Harman, 2005] (although bias may arise due to missing relevance judgments, *c.f.*, [Zobel, 1998, Carterette et al., 2010]).

In online evaluation, however, one usually is concerned with click-based metrics, such as those reviewed in Section 3.2. These metrics, unfortunately, typically depend on how documents are ranked on the result page, thanks to various biases (Section 5.2). Therefore, in general, click logs collected by running one ranker cannot be reliably used to evaluate a different ranker. Under certain assumptions, however, it is possible to reuse data to estimate online evaluation results of a new ranker, as shown in Chapter 4.

6

Online Evaluation in Practice

Although online evaluation is conceptually straightforward, many elements need to be correctly brought together to obtain reliable evaluations. This chapter focuses on such practical issues, aiming to guide the reader from theory to practical results on an actual IR system. It is intended for readers who wish to implement or validate online evaluations in practice.

This chapter will also present something of a generic recipe for online evaluation, briefly describing alternative ways to implement an experiment given a retrieval system, recruit & retain users, log results then analyze these. However, as there is limited literature directly considering the practice of online evaluation, our focus in prior work in this chapter is a discussion of examples.

6.1 Case Studies Approach

Online evaluation is easily interpreted as an evaluation where we observe how users are behaving when they interact with an IR system. However, such a view is misleading: The user is only observed through the limited instrumentation present in a website, and may be perform-

ing any number of different tasks while interacting with the system we are attempting to evaluate. Therefore, the evaluation must be carefully validated and assumptions made must be explicitly considered. It is most useful to do this through a series of case studies that highlight key issues that need to be considered in online experimentation. As the largest challenges come up at larger scales, such studies are the focus of this chapter. However, the same themes also apply to smaller-scale online evaluations.

While a variety of such case studies will be described, four are of particular note. Kohavi et al. [2012, 2014] present several case studies, and rules of thumb for AB testing. Bakshy and Frachtenberg [2015] discuss design and analysis from a performance point of view (see also Section 2.5). Finally, Li et al. [2015a] discuss how to design an exploration scheme, with many practical tips on analysis.

6.2 Ethical Considerations

Ethical considerations must be considered prior to any online evaluation. It is essential to realize the extent to which observational data of a person with a retrieval system is often personal data that the user does not necessarily, a priori, recognize as something that is recorded and possibly analyzed. Depending on the scenario, IR system users may not realize that their queries (even anonymously) can leak substantial personal or professional information.

An excellent illustration of this was the release of a sample of anonymized search engine logs by AOL in 2006. These contained anonymous user numbers and associated user search queries with limited metadata. However, the text of the queries themselves was sufficient for a number of the search engine users to be personally identified by journalists [1]. While approaches for anonymized logging [Feild et al., 2011] and anonymization of logs [Navarro-Arribas et al., 2012] are an area of ongoing research, in this chapter we do not discuss log distribution.

Even when logs are restricted to limited internal access, the private nature of collected data creates a natural tension. For analysis purposes

[1] http://www.nytimes.com/2006/08/09/technology/09aol.html?_r=0

to maximize the utility of an online evaluation, a general rule of thumb is to log as much of the interaction as possible. On the other hand, this increases the risk to user privacy. Achieving a good balance between powerful search evaluation and enforcing respect for user privacy must be a criterion when developing an online evaluation strategy.

A key consideration is *consent*, where it is clear to users what is being recorded, how the user can control that data (for example, delete it), and what potential evaluations are being performed. For instance, consider a recent AB test at Facebook that involved controlled experiments that modify the ranking of notifications listed on a social network feed [Kramer et al., 2014]. Given the known impact of social feeds on user well-being, this work prompted an extensive discussion of ethical considerations, for instance in the popular press.[2] and in blogs of privacy researchers[3]

Ethical and privacy concerns become even stronger when personalized systems are evaluated, where the evaluation is not of a straightforward server-side ranking algorithm but rather depends on personalization of search results based on sensitive user data. For instance, research on personalizing search results based on private history has sometimes been conducted with recruited users agreeing to participate but with all personal data deleted as soon as practical [Matthijs and Radlinski, 2011], while the traditional (offline) approach would be to log data client side and never require user interaction logs to be seen by the experimenter while still allowing reliable evaluation.

Depending on the situation, researchers should be aware of relevant organizational, institutional and professional requirements, for instance to do with working with human subjects.

6.3 Implementing Online Evaluations

Suppose that we have two retrieval systems that we wish to evaluate. In this section, we briefly describe the steps that need to be followed

[2] http://www.theatlantic.com/technology/archive/2014/06/everything-we-know-about-facebooks-secret-mood-manipulation-experiment/373648/

[3] https://medium.com/message/what-does-the-facebook-experiment-teach-us-c858c08e287f

to perform an evaluation. A more in-depth analysis of many of the steps that follow was also presented by Kohavi et al. [2009].

6.3.1 Metric Choice

First, we must decide on the metric or metrics that indicate experimental success – commonly called Key Performance Indicators (or KPIs). It is critical that this choice is made before the experiment is run, as often changes in a search system can improve some behavioral metrics but degrade others. Without this choice having been made prior to the experiment, it is impossible to objectively define success of an online evaluation or even reliably select the evaluation sample size required.

As an example of the importance of this step, Kohavi et al. [2012] describe an online evaluation where the number of queries per user for a search engine increased by 10%, and the revenue per user increased by 30%. However, this was the result of a bug whereby search result quality dropped dramatically leading to users needing to issue more queries, and sponsored advertisements becoming more relevant relative to the web results being shown to users. Clearly, when evaluating an search system, the success criterion should not be as simple as the short-term number of queries per user or revenue per user.

Another choice that should be made at this point is the experimental unit, as described by Kohavi et al. [2009]. Recall that common choices are per-user, per-query, or per-session. This choice determines how the metric of interest is computed. Readers are referred to Chapter 3 for a detailed discussion.

6.3.2 Choice of Implementation

The engineering effort required to perform an online evaluation can be substantial, and the choice of approach needs to trade off the experimenter's resources against the flexibility provided. In particular, the experimenter needs to consider whether there is a need to build a framework for future evaluations, or whether the evaluation is essentially one-off.

Supposing that we have an initial ranking system, and we have developed an improvement to it, how do we perform an evaluation?

Table 6.1: Some Online Evaluation Implementations

Method	Proxy	Add-On	Search/Server
Possible Scale	Small-Medium	Medium	Large
Habit Support	Yes	Yes	No
Observations	Web Traffic	Everything	Own Site
Robustness	Poor	Medium	High
Maintenance	Medium	Expensive	Easy
Effort Required	Small	Medium	High

We briefly list a number of considerations that lead to different evaluation approaches in Table 6.1. More details of these alternatives were discussed by Kelly et al. [2014].

The standard approach to online evaluation is server-side logging, implementing a full search system: When a user arrives at the search system, they are assigned to an experimental condition. Results are presented according to an AB or paired evaluation strategy and observations are collected. Many researchers have taken this approach, from small scale (*e.g.*, [Radlinski et al., 2008c]) to commercial web search scale (*e.g.*, [Arkhipova et al., 2015a]). This approach can run at any scale, and allows observation of all user interactions with the specific site, such as queries, clicking, mousing, and other related behavior. The biggest limitation of a server-side approach is that the system requires users engage with the system being evaluated. While not a concern for large web services, online evaluation of this sort may be harder for new systems: Users who do not have the habit of visiting the site being evaluated may drift away after only a few interactions and revert to their previously habitual online services.

One way to avoid this challenge is to build systems that essentially intercept habitual user interactions, and modify what is shown to the user. We now briefly describe two ways this can be achieved.

The first approach is to evaluate directly on the client using an add-on specifically developed for user's web browsers. To do this, an additional piece of software is installed on participant's web browser that can intercept and modify web pages before they are displayed to

users. As the evaluation involves running client-side software, there is more control, potential availability of richer input features, and the ability to measure a wider variety of user interaction metrics. On the other hand, this add-on must be actively installed by users to be part of the evaluation. Such an add-on essentially provides a transparent post-processing stage as the user browses, thus exploiting *habitual access* of existing services, while allowing *everything* to be observed with regards to the user's interaction with their web browser. This approach requires more maintenance than a server-side approach, as the add-on is necessarily browser dependent, and must correctly post-process the target web pages. However, the effort required to develop a simple add-on is typically smaller than the effort needed to develop a server-side system. We note that while the add-on approach most suitable for academic scale online evaluation (*e.g.*, [Matthijs and Radlinski, 2011]), some products inherently use client-side reprocessing hence are naturally suited to such an evaluation (*e.g.*, [Hardtke et al., 2009]).

Tip for small-scale experiments #14

Client-side approaches are particularly useful at small scales, and allow building a detailed understanding of user experiences, and development of custom metrics suited for highly specialized retrieval settings.

At smallest scales and with least implementation effort required, online evaluation can also be performed by modifying the configuration of a select group of users to *intercept* requests to any existing system by a third-party server[4], replacing the content served to users with new content that allows evaluation. This is done using a web proxy. It is only applicable when participants are recruited, and their web browser is reconfigured to route all the web traffic via this proxy. This method also exploits *habitual access* of existing services, in that the participant contin-

[4] Assuming the connection is not encrypted. An increasing number of search systems today encrypt the data connection by default, making this method less generally applicable.

ues to interact with their browser as always, and visit the same websites as habits dictate. It also allows the logging of all *web traffic*[4], specifically all requests to load any URL from the user's browser are routed through the proxy. However, such implementations are generally *not robust*: The proxy is separate from the search system, and if anything changes in the search system (such as the HTML formatting of results), the proxy must also be adapted. This makes ongoing maintenance of the system *relatively complex*. Furthermore, once the evaluation is completed, participants must have their systems reconfigured to no longer depend on the proxy. Finally, the most appealing aspect of such an evaluation approach is that implementation of a proxy is straightforward. This approach was used in research work, for example by Fujimoto et al. [2011].

6.3.3 Other Evaluation Strategies

Beyond these three general approaches, specific information retrieval tasks can be suited to more specific evaluation strategies. For instance, *Living Labs for IR Evaluation*[5] designed an online evaluation setup for testing web ranking approaches [Balog et al., 2014]. In addition to providing a way to obtain users, the framework also amortizes the cost of developing the experimentation infrastructure across many experimenters. Some commercial services today have similar aims.[6] As another example, Boll et al. [2011] designed smartphone apps with evaluation hidden in app tutorials, as games, and as general AB tests hidden in natural interactions. While their focus was not on information retrieval, the general strategy could be applied to selected information retrieval tasks.

Tip for small-scale experiments #15

These approaches may be particularly worth investigating if the small scale of a system makes other approaches difficult to use.

[5] `http://living-labs.net/`
[6] For instance, Optimizely or Visual Website Optimizer

6.3.4 What Should Be Logged?

We briefly return to the question of recording user interactions with an information retrieval system. Notwithstanding the earlier discussion of ethical considerations presented, we summarize information that is commonly useful for measuring common information retrieval quality metrics and for analyzing the outcomes of individual evaluations.

For this discussion we assume that the retrieval system is a search or recommendation engine. Typically logs record:

- The search/recommendation query (be it a text query, source item, or just user identifier).
- As detailed a view of what was shown to the user as practical (*i.e.*, rankings, snippets, any surrounding contextual content).
- As detailed a view of how the user responded to what was shown as practical (*e.g.*, clicks, mousing information).
- Available user identifiers, depending on the application, to permit session level and longer analysis as well as bot detection, aggregate statistics, *etc.*
- Interface metadata (such as the browser used).
- If a randomized evaluation like interleaving is being performed, the input ranking(s) provided to the randomization algorithm.

6.3.5 Establishing Evaluation Parameters

Once an evaluation metric and strategy has been selected, the next key consideration is how large and how long the online evaluation must be. Naïvely, one might assume that we should simply show the online experiment to users while periodically observing the outcome and concluding the experiment when there is a clear outcome using a simple statistical significance test. However, such an approach is invalid, as multiple-testing effects mean that we would instead over-interpret random fluctuations in metric values, and claim statistical significance

when there is none [Johari et al., 2015] (statistical methods for continuous testing are an area of ongoing research, *c.f.*, Section 2.3.3).

Rather, the experimenter needs to start by selecting: (1) The *sensitivity*, i.e. the smallest magnitude of difference in evaluation metric, that can be detected[7]; (2) The *power* of the test, which is the probability that a difference between the systems is detected when one actually exists; and (3) The *confidence* of the test, which is the probability that a detected difference between the systems is real, rather than the result of chance. There is a detailed discussion of this tradeoff in Kohavi et al. [2009], and the authors provide a rule of thumb for estimating sample sizes. Here, we illustrate with an example. Typically, online evaluations aim for a 95% confidence (in other words, if the experimenter runs 20 online experiments, 1 out of these is expected to detect a statistically significant difference purely due to chance), and a power of 80% or more (if there is a true difference of at least the target sensitivity, then there is an 80% chance of actually detecting it). In such a case, one can roughly estimate the sample size needed for the evaluation, following [van Belle, 2008, page 29]:

$$n = 16\sigma^2/\Delta^2, \tag{6.1}$$

where σ is the variance of the metric and Δ is a normalized required sensitivity.

Several statistical software packages support computing either sample size, sensitivity, power, and confidence when three of these are provided. For example, in `R` this functionality is provided in the package `pwr`,[8] and in `python` in the package `statsmodels`.[9] For further details on designing and planning experiments see Section 2.2.2.

The units of n are the experimental unit, such as queries, sessions, or users (*c.f.*, Section 2.2.3). To obtain this many samples, an experiment can be run on more users for less time, or fewer users for more

[7] It is often noted that just because an improvement is statistically significant does not mean that it is substantial enough to be noticed by individual users. Hence depending on the application, it may be useful to consider what differences are also *substantial* enough to warrant considering successful. Also, the tested hypotheses should be meaningful, *e.g.*, from a theory building perspective.

[8] See `https://cran.r-project.org/web/packages/pwr/pwr.pdf`.

[9] See `http://statsmodels.sourceforge.net/stable/stats.html`.

time (assuming the unit of experimentation is at the query or session level). While Kohavi et al. [2012] provide a detailed treatment of some necessary considerations we provide two key observations:

First, while shorter online evaluations may be desirable from the standpoint of experimental agility, natural fluctuations in user needs argue for longer evaluations [Kohavi et al., 2009]. For instance, many information retrieval systems naturally have different distributions of query loads on different days of the week, and at different times of day.

Second, the sample size needed depends on the variance of the metric. Some metrics are unbounded – such as the number of queries per user of an information retrieval system. As an evaluation is run for longer, the variance of this metric may increase, meaning that longer experiments may not have higher statistical significance.

6.4 Recruiting Users for Reliable Evaluation

Now that we have a functioning retrieval system and evaluation strategy, we turn to the next step: Finding users to evaluate the system. This is typically not a challenge for large scale systems with an established user base, hence this subsection focuses on smaller (for instance academic) evaluation settings. The goal is to measure performance on users who genuinely use the experimental system in a way that is consistent with how such a system would be used when ultimately deployed.

There are three basic levels for which users can be recruited. The most reliable is natural in-situ evaluation where the a system evaluation is based on real users performing natural tasks day-to-day. This is possible when the IR system provides clear utility to a sufficient number of people, and recruitment is akin to making people aware of the service, which by its very existence provides value. Provided the system is reliable and natural enough to use (for example, fast enough), recruitment is simply a matter of building sufficient awareness. For example, such an evaluation of an IR system was performed by Matthijs and Radlinski [2011]. It is important to note, however, that for evaluations where multiple approaches are evaluated on the same population of users in sequence, there may be inter-experiment

interactions (see Section 3.5 in Kohavi et al. [2012] for a detailed example). It is worth also noting that the evaluation environment need not match final system usage. For example, evaluation tasks for user interface design may be hidden say within a game to obtain larger user populations more easily [Boll et al., 2011].

Tip for small-scale experiments #16

Gamification, or embedding an evaluation in a different application, is an interesting way to more easily recruit users.

A less generalizable approach is evaluation using recruited users who nonetheless are asked to perform specified tasks in a naturalistic environment. For instance, this is the approach used by common crowd sourcing platforms such as Amazon Mechanical Turk [10]. In an IR setting, this may involve preparing a set of required tasks. It is most effective when the users performing the evaluation have a natural interest and expertise in the tasks being performed that would match that of the users of the final system. An overview of designing such evaluation approaches was recently published by Alonso [2013].

The final approach for evaluation of an IR system is a lab study, where users are recruited to perform pre-specified tasks in a controlled environment. On the one hand, this allows the most control in the evaluation, and the experimenter can see if users behave in unexpected ways. This provides valuable early-stage feedback on IR system design and performance. Additionally, a lab study allows very specific experiments to be designed to address particular questions in the design or implementation of an IR system. On the other hand, the simple act of observing such users affects how they interact with the system being evaluated, for instance perhaps causing them to pay more attention to results being presented than actual users would. This leads to such studies having a higher risk of providing results that are not representative of real-world usage of IR systems in a public setting. Therefore, validating that lab study users behave in ways consistent with natural

[10] `http://mturk.com/`

users is an important part of any such evaluation. For instance, in a web search setting, it has been observed that even imperceptible changes in the time it takes for an IR system to return results can have large impact on user behavior [Kohavi et al., 2013]. As an example of research taking the lab study approach, see Kelly and Azzopardi [2015].

Tip for small-scale experiments #17

Lab studies are usually run at smaller scales by necessity. There is extensive literature on how to run these to obtain the maximum information. [Kelly and Azzopardi, 2015] is a great starting point.

6.5 Validation, Log Analysis and Filtering

An important aspect of online evaluation with unobserved users is data validation. This often takes a number of forms. The most obvious is that, as part of the development process, known actions by the experimenter should be confirmed to be logged correctly and have the correct impact on online metric computation.

During and after the actual evaluation with users, it is important to validate that the results generally make sense,[11] and search for reasons when the outcome may be unexpected. For instance, Kohavi et al. [2014] present seven rules that capture many common experimental observations. The first rule notes that seemingly small changes can have a big impact on key metrics, illustrating this with experiments from small errors causing failures to changing how a website responds to user clicks. However, a second rule points out that changes rarely have a big *positive* impact on key metrics, hence such outcomes should be treated carefully. Such changes in metrics need to be drilled into during validation.

Validation is also important whenever randomization occurs. For instance, if the results presented to search engine users depend on a random number generator (such as when assigning users to the experimental condition, or within an experimental condition with interleaved evaluation), it is important to validate that the random

[11] But be aware of confirmation bias [Nickerson, 1998].

numbers generated do not exhibit unexpected properties. For instance, if multiple servers present search results and these periodically reset, do they always initialize to the same random seed, causing the samples observed to not respect the intended statistical properties? As another example, if performing an experiment where the unit of experimentation is a user, and users are initially divided into experimental conditions at known rate using browser cookies (say, 50% control and 50% treatment), the balance of users at the end of the experiment should be validated. If it changes, it may be that users in one condition are more likely to abandon the IR system (a valid outcome), but might also mean that one of the conditions simply corrupts the cookies used to assign users to experimental conditions.

Another common concern with online IR evaluation is the effect of automated crawlers (also known as bots). In particular, when computing metrics where a mean metric is computed, even one automated crawler that is not eliminated may skew the mean meaningfully. Kohavi et al. [2009] note that using javascript logging is less prone to robots being logged, although in some cases automated crawlers may be malicious. In some situations it may be beneficial to compute medians rather than means to make the metric more robust to outliers.

6.6 Considerations and Tools for Data Analysis

Once an experiment has been completed, and data has been carefully validated, the final step is analyze the data, and to compute statistical significance of any outcomes found. An overview of data analysis using regression analysis was given in Section 2.3. In addition, there has been substantial work published on statistical significance testing for information retrieval (for instance, see Carterette [2013] for an overview). Here, we summarize a few statistical tools that can particularly useful when performing online evaluation of information retrieval systems.

6.6.1 Modeling and Analysis Tools

Many tools and toolboxes exist for analyzing experiment data. Here, we focus specifically on tools that facilitate modeling and analysis

using regression analysis. Standard hypothesis testing (*e.g.*, t-test) is widely supported by statistical software packages. Common choices that are available under open source licenses include `R`, and the `python` package `scipy`.

Approaches for estimating model parameters in more complex models have been developed over the past years. These either approach the task from a frequentist perspective (and fit models using OLS), or from a Bayesian perspective (and estimate model parameters using, *e.g.*, sampling techniques). State of the art implementations for the statistics software `R` can be found in the packages `lme4`[12] [Baayen et al., 2008, Bates et al., 2014] (model coefficients are estimated using OLS) and `MCMCglmm`[13] (estimation using Markov-Chain Monte Carlo methods) [Hadfield et al., 2010]. A subset of this functionality is provided in the `python` packages `scikit-learn`[14] and `statsmodels`[15].

Stand-alone tools for statistical modeling from a Bayesian perspective are `BUGS`[16] and `infer.net`[17] [Minka et al., 2014].

6.6.2 Assessing Statistical Power

When a sequence of online evaluations is to be performed, understanding the statistical properties of metrics of interest is often useful. In general, given non-uniformity in how users arrive and generate information needs, measuring how sensitive the outcome is to experiment length and sample sizes can be non-trivial. One approach to assess the sensitivity of online evaluation is to use bootstrap sampling to resample logs, subsampling experimental units with replacement. For instance, Chapelle et al. [2012] used this approach to measure the relative sensitivity of different metrics.

[12] See `https://cloud.r-project.org/web/packages/lme4/index.html`.
[13] See `https://cloud.r-project.org/web/packages/MCMCglmm/index.html`.
[14] See `http://scikit-learn.org/stable/modules/linear_model.html`.
[15] See `http://statsmodels.sourceforge.net/devel/mixed_linear.html`.
[16] See `http://www.openbugs.net`.
[17] See `http://research.microsoft.com/infernet`.

6.6.3 Maximizing Sensitivity

Finally, a common concern in online evaluation is maximizing the sensitivity of any online evaluation. Depending on the evaluation metric, different approaches may be relevant. With AB evaluations, it is often the case that different user queries are known ahead of time to have different metric values. For instance, we may be known that one word queries are much higher CTR than five word queries. When measuring significance of a metric such as CTR, such a wide range of values for different query classes increases the variance of the metric, requiring larger sample sizes to achieve statistical significance. However, if different queries are separately analyzed, the statistical power of the same size sample can be greatly increased [Deng et al., 2013, Deng, 2015].

In the case of interleaving evaluations, we start with much higher sensitivity. However, as we saw in Section 2.7 that the interleaving metric can also be tuned to directly optimize it for the statistical power of an evaluation. If performing an evaluation where sensitivity is more important than metric interpretability, and if a number of similar experimental evaluations is available to tune the parameters to achieve this, such an approach may be warranted (for instance, Hofmann et al. [2012a], Yue et al. [2010a]).

7

Concluding Remarks

Evaluation of information retrieval systems is among the core problems in IR research and practice. The key challenge is to design reliable methodology to measure an IR system's effectiveness of satisfying users' information need. There are roughly two types of approaches. *Offline* approaches such as the Cranfield paradigm, while effective for measuring topical relevance, have difficulty taking into account contextual information including the user's current situation, fast changing information needs, and past interaction history with the system. The *online* approach, on the other hand, aims to measure the actual utility of a fully functioning IR system in a natural usage environment. In contrast to offline approaches, user feedback in online evaluation is usually implicit, in the forms of clicks, dwell time, *etc.* Consequently, online evaluation requires a rather different set of methods and tools than those used in offline evaluation.

In this survey, we provide an extensive survey of existing online evaluation techniques for information retrieval. In particular, we start with a general overview of controlled experiments, the scientific foundation of most online evaluation methods, in the context of information retrieval. Then, we review the large collection of metrics

that have been proposed in the IR literature for different tasks and research questions. These metrics turn low-level, implicit user feedback into aggregated quantities that characterize different aspects of an IR system. We also cover the important topic of *offline* estimation of online evaluation results using historical data. These techniques are particularly useful when running online experiments are expensive and time-consuming. On the more practical side, we also discuss pros and cons of online evaluation, as well as often encountered practical issues.

Online evaluation for information retrieval is an active area of current research. Much of the recent research has been driven by requirements of industry applications. Now, online evaluation platforms are developed for academic purposes and fruitful collaborations between industry and academia are emerging. As these expand, we expect new insights and new applications that will drive the development of new methods and tools. Some of the most prominent research directions are discussed in this chapter.

The trend towards increased research in, and use of, online evaluation is well illustrated by the introduction of online evaluation in academic evaluation campaigns. The *living labs* challenge first introduced online evaluation to CLEF [Balog et al., 2014]. Following this setup, industry partners provide access to (anonymized) queries to subscribing teams through an API, and provide the opportunity for teams to generate some of the rankings used. Observed clicks and other meta data are then fed back to the subscribing teams, allowing them to evaluate and tune their ranking approaches using real usage data. The success of this model is demonstrated by the recent extension to the TREC OpenSearch track,[1] which focuses on academic literature search. This trend provides an exciting opportunity for information retrieval research to extend beyond static collection and laboratory settings, and investigate retrieval under real usage data.

The living labs setup is one answer to the problem of sharing access to usage data, and setting up repeatable experiments. Problems related to privacy have severely limited access to usage data in the past. The anonymized access to a limited set of queries provided

[1] See `http://trec-open-search.org` for details.

through living labs addresses both privacy issues and the need for access to usage data. Another line of important work studies how log data can be anonymized while preserving information that is crucial for research [Feild et al., 2011, Navarro-Arribas et al., 2012, Hong et al., 2015]. New insights in this area can lead to sharing more information to benefit research while at the same time better preserving users' privacy. An open question is whether part of the setup could be replaced by a simulator. For example, can observed usage data inform a sophisticated simulator that captures the key aspects of user behavior required for valid IR experimentation? Approaches have been proposed in the context of known item search [Chowdhury and Soboroff, 2002, Azzopardi et al., 2007] and to assess properties of online learning approaches under varying user models [Hofmann et al., 2011a, Chuklin et al., 2014]. Developing high-quality simulators for broader classes of user behavior remains an open problem, which may benefit from eye-tracking [Guan and Cutrell, 2007a, Buscher et al., 2010, Hofmann et al., 2014] and mouse movement studies [Diaz et al., 2013].

Many challenges remain to be addressed. First, despite the potential for online experimentation using "big data" — the available amount of data is typically orders of magnitude greater than in laboratory studies or offline evaluation — issues of scale remain an enormous challenge. The increased realism of online data also means increased variance and reduced experimental control. Thus, methods for effectively reducing variance will remain an important area of research. In the extreme, an ideal information retrieval system would provide the most relevant information for each individual user at each point in time with the smallest possible amount of training data or exploration. Moving towards this ambitious goal will require a concerted effort in statistical techniques, user modeling, and machine learning techniques such as one shot learning.

Moving towards online evaluation does not mean that insights gained in offline evaluation should be ignored. The wealth of insights in models and metrics, designed to effectively capture user needs and expectations, can inspire and inform work in online evaluation. A central open question is how to align offline and online metrics. A

reliable alignment would allow practitioners and researchers to benefit from the best of both worlds, *e.g.*, using existing offline data for initial testing before moving to online evaluation. First insights in how to relate online and offline metrics, and how to tune them to achieve better alignment are provided in Carterette and Jones [2007], Ozertem et al. [2011] and Schuth et al. [2015b].

As online experimentation and evaluation mature, automation is becoming an important topic. Instead of setting up each experiment individually, such as a series of AB tests, online learning approaches, bandit algorithm and sequential design of experiments provide routes towards automatically constructing and selecting the most informative experiment given previous observations. Tools like the Metric Optimization Engine[2] apply Bayesian optimization methods to construct informative experiments. Bakshy et al. [2014] present a tool for deploying online experiments that covers a wide range of experiment designs, including factorial designs.

A range of metrics have been proposed for online evaluation, ranging from simple click-based metrics to composite metrics of engagement and user satisfaction (discussed in Chapter 3). So far, no single universal metric has been identified. Most likely, different metrics will be required for different IR applications and needs of the experimenter. New metrics are currently explored for applications that go beyond scenarios like Web search, *e.g.*, for mobile search [Guo et al., 2013b, Lagun et al., 2014] and interaction with conversational (search) agents [Jiang et al., 2015b]. As information retrieval systems move towards more conversational settings, we will need a better understanding of long-term metrics. Compared to the currently dominant query-based and session-based metrics, longer term metrics could assess effects of learning and gaining and remembering information throughout search tasks and interests that last weeks or months. Hohnhold et al. [2015] demonstrate the value of optimizing for long-term metrics in online advertising. Typical problems of measuring long-term effects are illustrated in Kohavi et al. [2012]. User engagement metrics are designed to capture engagement over multiple weeks [Dupret and Lalmas, 2013].

[2] `https://github.com/yelp/MOE`

Finally, online evaluation can benefit from cross-fertilization with other trends in IR evaluation, in particular *crowdsourcing* and *gamification*. Crowdsourcing [Alonso and Mizzaro, 2009, Kazai et al., 2011, Zhou et al., 2014] has become a popular source of relatively cheap labeled data for constructing test collections for offline evaluation. Crowdsourcing workers on platforms such as Amazon Mechanical Turk[3] and a number of similar platforms are paid small amounts of money for completing small HITs – human intelligence tasks. Ease of access and low prices make these platforms a popular alternative to much more expensive expert relevance judgments [Mason and Suri, 2012]. Combining crowdsourcing with online evaluation can lead to novel experiment setups where crowd workers complete simulated or actual search tasks using fully functional search interfaces. Driven by the move towards crowdsourcing instead of expert judgments, gamification has become a popular method for increasing engagement and label quality [Eickhoff et al., 2012]. The boundaries between online evaluation and gamified assessments via crowdsourcing are expected to become blurred. Many of the developed techniques apply to both, and combinations can lead to innovative experiment designs that generate detailed insights into the motivations and behaviors of information seekers.

[3] `https://www.mturk.com/mturk/welcome`

Acknowledgements

This work benefited from input of many colleagues in the field. The editor, Mark Sanderson, and the anonymous reviewers made a number of excellent suggestions that greatly improved the presentation and contents. Miro Dudík and John Langford provided helpful feedback for an earlier draft. Finally, we would also like to thank our colleagues and collaborators, especially Leon Bottou, Wei Chu, Nick Craswell, Miro Dudík, Thorsten Joachims, John Langford, Maarten de Rijke, Rob Schapire, Anne Schuth, Shimon Whiteson, and Masrour Zoghi, who have shaped our work and views of online evaluation for information retrieval over the past years.

References

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Miro Dudík, John Langford, Lihong Li, Luong Hoang, Dan Melamed, Siddhartha Sen, Robert Schapire, and Alex Slivkins. Multi-world testing: A system for experimentation, learning, and decision-making, 2016. Microsoft whitepaper. URL: http://mwtds.azurewebsites.net.

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. Click shaping to optimize multiple objectives. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 132–140, 2011.

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. Personalized click shaping through Lagrangian duality for online recommendation. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 485–494, 2012.

Mikhail Ageev, Qi Guo, Dmitry Lagun, and Eugene Agichtein. Find it if you can: A game for modeling different types of web search success using interaction data. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, 2011.

Rakesh Agrawal, Alan Halverson, Krishnaram Kenthapadi, Nina Mishra, and Panayiotis Tsaparas. Generating labels from clicks. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 172–181, 2009.

Omar Alonso. Implementing crowdsourcing-based relevance experimentation: An industrial perspective. *Information Retrieval*, 16:101–120, 2013.

Omar Alonso and Stefano Mizzaro. Can we get rid of TREC assessors? using Mechanical Turk for relevance assessment. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, volume 15, page 16, 2009.

Natalia Andrienko and Gennady Andrienko. *Exploratory analysis of spatial and temporal data: A systematic approach*. Springer Science & Business Media, 2006.

Olga Arkhipova, Lidia Grauer, Igor Kuralenok, and Pavel Serdyukov. Search engine evaluation based on search engine switching prediction. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 723–726, 2015a.

Olga Arkhipova, Lidia Grauer, Igor Kuralenok, and Pavel Serdyukov. Search engine evaluation based on search engine switching prediction. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 723–726, 2015b.

Arthur Asuncion and David J. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Leif Azzopardi. Modelling interaction with economic models of search. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 3–12, 2014.

Leif Azzopardi, Maarten De Rijke, and Krisztian Balog. Building simulated queries for known-item topics: An analysis using six european languages. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 455–462, 2007.

R Harald Baayen, Douglas J Davidson, and Douglas M Bates. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412, 2008.

Yoram Bachrach, Sofia Ceppi, Ian A. Kash, Peter Key, and David Kurokawa. Optimising trade-offs among stakeholders in ad auctions. In *Proceedings of the ACM Conference on Economics and Computation (EC-14)*, pages 75–92, 2014.

Eytan Bakshy and Eitan Frachtenberg. Design and analysis of benchmarking experiments for distributed internet services. In *Proceedings of the International World Wide Web Conference (WWW)*, 2015.

Eytan Bakshy, Dean Eckles, and Michael S. Bernstein. Designing and deploying online field experiments. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 283–292, 2014.

Krisztian Balog, Liadh Kelly, and Anne Schuth. Head first: Living labs for ad-hoc search evaluation. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, 2014.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*, 2014.

Michael Bendersky, Lluis Garcia-Pueyo, Jeremiah Harmsen, Vanja Josifovski, and Dima Lepikhin. Up next: Retrieval methods for large scale related video suggestion. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1769–1778, 2014.

Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 129–138, 2009.

Susanne Boll, Niels Henze, Martin Pielot, Benjamin Poppinga, and Torben Schinke. My app is an experiment: Experience from user studies in mobile app stores. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 3(4):71–91, 2011.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis Xavier Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research (JMLR)*, 14:3207–3260, 2013.

Justin Boyan, Dayne Freitag, and Thorsten Joachims. A machine learning architecture for optimizing Web search engines. In *AAAI Workshop on Internet Based Information Systems*, pages 1–8, 1996.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Giuseppe Burtini, Jason Loeppky, and Ramon Lawrence. A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757*, 2015.

Georg Buscher, Susan T. Dumais, and Edward Cutrell. The good, the bad, and the random: An eye-tracking study of ad quality in web search. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 42–49, 2010.

Donald T Campbell and Julian C Stanley. *Experimental and Quasi-Experimental Designs for Research*. Houghton Mifflin Company, 1966.

Ben Carterette. Statistical significance testing in information retrieval: Theory and practice. In *Proceedings of the International Conference on The Theory of Information Retrieval (ICTIR)*, 2013.

Ben Carterette and Rosie Jones. Evaluating Search Engines by Modeling the Relationship Between Relevance and Clicks. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 217–224, 2007.

Ben Carterette, Evgeniy Gabrilovich, Vanja Josifovski, and Donald Metzler. Measuring the reusability of test collections. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 231–240, 2010.

George Casella and Roger L. Berger. *Statistical Inference*. Cengage Learning, 2nd edition, 2001.

Sunandan Chakraborty, Filip Radlinski, Milad Shokouhi, and Paul Baecke. On correlation of absence time and search effectiveness. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1163–1166, 2014.

Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research - Proceedings Track*, 14:1–24, 2011.

Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 1–10, 2009.

Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *Transactions on Information System (TOIS)*, 30(1):6:1–6:41, 2012.

Ye Chen, Yiqun Liu, Ke Zhou, Meng Wang, Min Zhang, and Shaoping Ma. Does vertical bring more satisfaction?: Predicting search satisfaction in a heterogeneous environment. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1581–1590, 2015.

Abdur Chowdhury and Ian Soboroff. Automatic evaluation of world wide web search services. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 421–422, 2002.

Aleksandr Chuklin, Anne Schuth, Katja Hofmann, Pavel Serdyukov, and Maarten de Rijke. Evaluating Aggregated Search Using Interleaving. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, 2013a.

Aleksandr Chuklin, Pavel Serdyukov, and Maarten de Rijke. Click model-based information retrieval metrics. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 493–502, 2013b.

Aleksandr Chuklin, Anne Schuth, Ke Zhou, and Maarten de Rijke. A comparative analysis of interleaving methods for aggregated search. *Transactions on Information System (TOIS)*, 2014.

Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search. Synthesis Lectures on Information Concepts, Retrieval, and Services*, volume 7. Morgan & Claypool Publishers, 2015.

Charles L. A. Clarke, Eugene Agichtein, Susan Dumais, and Ryen W. White. The influence of caption features on clickthrough patterns in web search. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 135–142, 2007.

Cyril Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19(6):173–194, 1967.

Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 87–94, 2008.

Alex Deng. Objective Bayesian two sample hypothesis testing for online controlled experiments. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 923–928, 2015.

Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 123–132, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

Fernando Diaz, Ryen White, Georg Buscher, and Dan Liebling. Robust models of mouse movement on dynamic web search results pages. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1451–1460, 2013.

Abdigani Diriye, Ryen White, Georg Buscher, and Susan Dumais. Leaving so soon? Understanding and predicting web search abandonment rationales. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1025–1034, 2012.

Anlei Dong, Jiang Bian, Xiaofeng He, Srihari Reddy, and Yi Chang. User action interpretation for personalized content optimization in recommender systems. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 2129–2132, 2011.

Alexey Drutsa, Gleb Gusev, and Pavel Serdyukov. Future user engagement prediction and its application to improve the sensitivity of online experiments. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 256–266, 2015a.

Alexey Drutsa, Gleb Gusev, and Pavel Serdyukov. Engagement periodicity in search engine usage: Analysis and its application to search quality evaluation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 27–36, 2015b.

Alexey Drutsa, Anna Ufliand, and Gleb Gusev. Practical aspects of sensitivity in online experimentation with user engagement metrics. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 763–772, 2015c.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1097–1104, 2011.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Sample-efficient nonstationary-policy evaluation for contextual bandits. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 247–254, 2012.

Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, pages 563–587, 2015.

Georges Dupret and Mounia Lalmas. Absence time and user engagement: Evaluating ranking functions. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 173–182, 2013.

Georges Dupret and Ciya Liao. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 181–190, 2010.

Georges E. Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 331–338, 2008.

Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman & Hall, 1993.

Carsten Eickhoff, Christopher G Harris, Arjen P de Vries, and Padmini Srinivasan. Quality through flow and immersion: gamifying crowdsourced relevance assessments. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 871–880, 2012.

Henry Allen Feild, James Allan, and Rosie Jones. Predicting searcher frustration. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 34–41, 2010.

Henry Allen Feild, James Allan, and Joshua Glatt. Crowdlogging: distributed, private, and anonymous search logging. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 375–384, 2011.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan T. Dumais, and Thomas White. Evaluating implicit measures to improve Web search. *Transactions on Information System (TOIS)*, 23(2):147–168, 2005.

Hiroshi Fujimoto, Minoru Etoh, Akira Kinno, and Yoshikazu Akinaga. Web user profiling on proxy logs and its evaluation in personalization. In *Proceedings of the Asia-Pacific web conference on Web technologies and applications (APWeb)*, pages 107–118, 2011.

A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press, 2006.

Andrew Gelman et al. Analysis of variance – why it is more important than ever. *The Annals of Statistics*, 33(1):1–53, 2005.

Laura A Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 478–479, 2004.

Artem Grotov, Shimon Whiteson, and Maarten de Rijke. Bayesian ranker comparison based on historical user interactions. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 273–282, 2015.

Zhiwei Guan and Edward Cutrell. What are you looking for? An eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 407–416, 2007a.

Zhiwei Guan and Edward Cutrell. An eye tracking study of the effect of target rank on web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 417–420, 2007b.

Huan Gui, Ya Xu, Anmol Bhasin, and Jiawei Han. Network A/B testing: From sampling to estimation. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 399–409, 2015.

Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael J. Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in Web search. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 11–20, 2009a.

Fan Guo, Chao Liu, and Yi-Min Wang. Efficient multiple-click models in Web search. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 124–131, 2009b.

Qi Guo and Eugene Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 569–578, 2012.

Qi Guo, Shuai Yuan, and Eugene Agichtein. Detecting success in mobile search from interaction. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1229–1230, 2011.

Qi Guo, Haojian Jin, Dmitry Lagun, Shuai Yuan, and Eugene Agichtein. Mining touch interaction data on mobile devices to predict web search result relevance. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 153–162, 2013a.

Qi Guo, Haojian Jin, Dmitry Lagun, Shuai Yuan, and Eugene Agichtein. Towards estimating web search result relevance from touch interactions on mobile devices. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1821–1826, 2013b.

Jarrod D Hadfield et al. Mcmc methods for multi-response generalized linear mixed models: the mcmcglmm r package. *Journal of Statistical Software*, 33(2):1–22, 2010.

Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011.

David Hardtke, Mike Wertheim, and Mark Cramer. Demonstration of improved search result relevancy using real-time implicit relevance feedback. *Understanding the User – Workshop in conjunction with SIGIR*, 2009.

Ahmed Hassan and Ryen W. White. Personalized models of search satisfaction. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 2009–2018, 2013.

Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. Beyond dcg: user behavior as a predictor of a successful search. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 221–230, 2010.

Ahmed Hassan, Yang Song, and Li-Wei He. A task level user satisfaction metric and its application on improving relevance estimation. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 125–134, 2011.

Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. Beyond clicks: Query reformulation as a predictor of search satisfaction. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 2019–2028, 2013.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2nd edition, 2009. 7th printing 2013 edition.

William Hersh, Andrew H. Turpin, Susan Price, Benjamin Chan, Dale Kramer, Lynetta Sacherek, and Daniel Olson. Do batch and user evaluations give the same results? In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 17–24, 2000.

Keisuke Hirano, Guido W. Imbens, and Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

Katja Hofmann, Bouke Huurnink, Marc Bron, and Maarten de Rijke. Comparing click-through data to purchase decisions for retrieval evaluation. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 761–762, 2010.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Balancing exploration and exploitation in learning to rank online. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, volume 6611 of *Lecture Notes in Computer Science*, pages 251–263, 2011a.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 249–258, 2011b.

Katja Hofmann, Fritz Behr, and Filip Radlinski. On caption bias in interleaving experiments. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 115–124, 2012a.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Estimating interleaved comparison outcomes from historical click data. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1779–1783, 2012b.

Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 183–192, 2013a.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16(1):63–90, 2013b.

Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *Transactions on Information System (TOIS)*, 31(4), 2013c.

Katja Hofmann, Bhaskar Mitra, Filip Radlinski, and Milad Shokouhi. An eye-tracking study of user interactions with query auto completion. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 549–558, 2014.

Henning Hohnhold, Deirdre O'Brien, and Diane Tang. Focusing on the long-term: It's good for users and business. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1849–1858, 2015.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(6):945–960, 1986.

Yuan Hong, Jaideep Vaidya, Haibing Lu, Panagiotis Karras, and Sanjay Goel. Collaborative search log sanitization: Toward differential privacy and boosted utility. *IEEE Transactions on Dependable and Secure Computing*, 12(5):504–518, 2015.

Edward L. Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Bernard J. Jansen. Search log analysis: What it is, what's been done, how to do it. *Library & Information Science Research*, 28(3):407 – 432, 2006.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *Transactions on Information System (TOIS)*, 20(4):422–446, 2002.

Jiepu Jiang, Ahmed Hassan Awadallah, Rosie Jones, Umut Ozertem, Imed Zitouni, Ranjitha Gurunath Kulkarni, and Omar Zia Khan. Automatic online evaluation of intelligent personal assistants. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 506–516, 2015a.

Jiepu Jiang, Ahmed Hassan Awadallah, Rosie Jones, Umut Ozertem, Imed Zitouni, Ranjitha Gurunath Kulkarni, and Omar Zia Khan. Automatic online evaluation of intelligent assistants. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 506–516, 2015b.

Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W. White. Understanding and predicting graded search satisfaction. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 57–66, 2015c.

Nan Jiang and Lihong Li. Doubly robust off-policy evaluation for reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *Transactions on Information System (TOIS)*, 25(2), 2007.

Ramesh Johari, Leo Pekelis, and David J. Walsh. Always valid inference: Bringing sequential analysis to A/B testing. *arXiv Preprint arXiv:1512.04922v1 [math.ST]*, 2015.

Gabriella Kazai, Jaap Kamps, Marijn Koolen, and Natasa Milic-Frayling. Crowdsourcing for book search evaluation: Impact of hit design on comparative system ranking. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 205–214, 2011.

Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1–2): 1–224, 2009.

Diane Kelly and Leif Azzopardi. How many results per page? a study of SERP size, search behavior and user experience. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 183–192, 2015.

Diane Kelly and Karl Gyllstrom. An examination of two delivery modes for interactive search system experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1531–1540, 2011.

Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

Diane Kelly, Filip Radlinski, and Jaime Teevan. Choices and constraints: Research goals and approaches in information retrieval. *Tutorial at Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2014.

Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. Using historical click data to increase interleaving sensitivity. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 679–688, 2013.

Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. Optimised scheduling of online experiments. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 453–462, 2015a.

Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. Generalized team draft interleaving. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 773–782, 2015b.

Eugene Kharitonov, Aleksandr Vorobev, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. Sequential testing for early stopping of online experiments. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 473–482, 2015c.

Youngho Kim, Ahmed Hassan, Ryen White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 193–202, 2014a.

Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. Comparing client and server dwell time estimates for click-level satisfaction prediction. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 895–898, 2014b.

Julia Kiseleva, Kyle Williams, Jiepu Jiang, Ahmed Hassan Awadallah, Imed Zitouni, Aidan C Crook, and Tasos Anastasakos. Predicting user satisfaction with intelligent assistants. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, 2016.

Jon Kleinberg. Temporal dynamics of on-line information streams. In Minos Garofalakis, Johannes Gehrke, and Rajeev Rastogi, editors, *Data Stream Management: Processing High-Speed Data Streams*. Springer, 2004.

Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. Controlled experiments on the web: Survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.

Ron Kohavi, Alex Deng, Brian Frasca, Roger Longbotham, Toby Walker, and Ya Xu. Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 786–794, 2012.

Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. Online controlled experiments at large scale. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1168–1176, 2013.

Ron Kohavi, Alex Deng, Roger Longbotham, and Ya Xu. Seven rules of thumb for web site experimenters. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1857–1866, 2014.

Adam D. I. Kramer, Jamie E. Guillory, and Jeffrey T. Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, 2012.

Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 113–122, 2014.

John Langford, Alexander Strehl, and Jennifer Wortman. Exploration scavenging. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 528–535, 2008.

John Lawson. *Design and Analysis of Experiments with R*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2014.

Jane Li, Scott Huffman, and Akihito Tokuda. Good abandonment in mobile and PC Internet search. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 43–50, 2009.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 661–670, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 297–306, 2011.

Lihong Li, Shunbao Chen, Ankur Gupta, and Jim Kleban. Counterfactual analysis of click metrics for search engine optimization: A case study. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 929–934, 2015a.

Lihong Li, Jin Kim, and Imed Zitouni. Toward predicting the outcome of an A/B experiment for search relevance. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 37–46, 2015b.

Lihong Li, Remi Munos, and Csaba Szepesvári. Toward minimax off-policy value estimation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 608–616, 2015c.

Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer-Verlag, 2001. ISBN 0387763694.

Winter Mason and Siddharth Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44(1):1–23, 2012.

Nicolaas Matthijs and Filip Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 25–34, 2011.

T. Minka, J.M. Winn, J.P. Guiver, S. Webster, Y. Zaykov, B. Yangel, A. Spengler, and J. Bronskill. Infer.NET 2.6, 2014. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

Taesup Moon, Wei Chu, Lihong Li, Zhaohui Zheng, and Yi Chang. An online learning framework for refining recency search results with user click feedback. *Transactions on Information System (TOIS)*, 30(4), 2012.

Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 272–281, 1994.

Susan A. Murphy, Mark van der Laan, and James M. Robins. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 96(456):1410–1423, 2001.

Guillermo Navarro-Arribas, Vicenç Torra, Arnau Erola, and Jordi Castellà-Roca. User k-anonymity for privacy preserving data mining of query logs. *Information Processing & Management*, 48(3):476–487, 2012.

Raymond S. Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology*, 2(2):175–220, 1998.

Olivier Nicol, Jérémie Mary, and Philippe Preux. Improving offline evaluation of contextual bandit algorithms via bootstrapping techniques. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 172–180, 2014.

Kirill Nikolaev, Alexey Drutsa, Ekaterina Gladkikh, Alexander Ulianov, Gleb Gusev, and Pavel Serdyukov. Extreme states distribution decomposition method for search engine online evaluation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 845–854, 2015.

Daan Odijk, Ryen W. White, Ahmed Hassan Awadallah, and Susan T. Dumais. Struggling and success in web search. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1551–1560, 2015.

Umut Ozertem, Rosie Jones, and Benoit Dumoulin. Evaluating new search engine configurations with pre-existing judgments and clicks. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 397–406, 2011.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Ashok Kumar Ponnuswami, Kumaresh Pattabiraman, Desmond Brand, and Tapas Kanungo. Model characterization curves for federated search using click-logs: Predicting user engagement metrics for the span of feasible operating points. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 67–76, 2011.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 759–766, 2000.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, 2008.

Filip Radlinski and Nick Craswell. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 667–674, 2010.

Filip Radlinski and Nick Craswell. Optimized Interleaving for Online Retrieval Evaluation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 245–254, 2013.

Filip Radlinski and Thorsten Joachims. Minimally invasive randomization for collecting unbiased preferences from clickthrough logs. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1406–1412, 2006.

Filip Radlinski, Andrei Broder, Peter Ciccolo, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel. Optimizing relevance and revenue in ad search: A query substitution approach. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 403–410, 2008a.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 784–791, 2008b.

Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 43–52, 2008c.

Andrea Rotnitzky and James M. Robins. Semiparametric regression estimation in the presence of dependent censoring. *Biometrika*, 82(4):805–820, 1995.

Mark Sanderson. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.

Falk Scholer, Milad Shokouhi, Bodo Billerbeck, and Andrew Turpin. Using Clicks as Implicit Judgments: Expectations Versus Observations. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 28–39, 2008.

Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. Multileaved comparisons for fast online evaluation. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 71–80, 2014.

Anne Schuth, Robert-Jan Bruintjes, Fritjof Buüttner, Joost van Doorn, Carla Groenland, Harrie Oosterhuis, Cong-Nguyen Tran, Bas Veeling, Jos van der Velde, Roger Wechsler, et al. Probabilistic multileave for online retrieval evaluation. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 955–958, 2015a.

Anne Schuth, Katja Hofmann, and Filip Radlinski. Predicting search satisfaction metrics with interleaved comparisons. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 463–472, 2015b.

William R. Shadish, Thomas D. Cook, and Donald Thomas Campbell. *Experimental and quasi-experimental designs for generalized causal inference*. Wadsworth Cengage learning, 2002.

Fabrizio Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1–2):1–174, 2010.

Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 238–241, 1951.

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: Learning diverse rankings over large document collections. *Journal of Machine Learning Research (JMLR)*, 14(1):399–436, 2013.

Yang Song, Hao Ma, Hongning Wang, and Kuansan Wang. Exploring and exploiting user search behavior on mobile and tablet devices to improve search relevance. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 1201–1212, 2013a.

Yang Song, Xiaolin Shi, and Xin Fu. Evaluating and predicting user engagement change with degraded search relevance. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 1213–1224, 2013b.

Alexander L. Strehl, John Langford, Lihong Li, and Sham M. Kakade. Learning from logged implicit exploration data. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 2217–2225, 2011.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 814–823, 2015a.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 3231–3239, 2015b.

Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudík, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation, 2016. arXiv:1605.04812.

B.G. Tabachnick and L.S. Fidell. *Using Multivariate Statistics: Pearson New International Edition*. Pearson Education Limited, 2013.

Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 17–26, 2010.

Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1587–1594, 2013.

Philip S. Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2380–2388, 2015.

Edward R Tufte and PR Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

Andrew Turpin and Falk Scholer. User performance versus precision measures for simple search tasks. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 11–18, 2006.

Andrew H. Turpin and William Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 225–231, 2001.

Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 329–337, 2013.

Antony Unwin. *Graphical Data Analysis with R*, volume 27. CRC Press, 2015.

Gerald van Belle. *Statistical Rules of Thumb*. Wiley-Blackwell, 2008.

Ellen M Voorhees. The effect of sampling strategy on inferred measures. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1119–1122, 2014.

Ellen M Voorhees and Donna K Harman. *TREC: Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing. MIT Press, 2005.

Dylan Walker and Lev Muchnik. Design of randomized experiments in networks. *Proceedings of the IEEE*, 102(12):1940–1951, 2014.

Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ahmed Hassan, and Ryen White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 123–132, 2014.

Kuansan Wang, Toby Walker, and Zijian Zheng. PSkip: estimating relevance ranking quality from web search clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1355–1364, 2009.

Kuansan Wang, Nikolas Gloy, and Xiaolong Li. Inferring search behaviors using partially observable markov (pom) model. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 211–220, 2010.

Kyle Williams, Julia Kiseleva, Aidan C. Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. Detecting good abandonment in mobile search. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 495–505, 2016.

Dragomir Yankov, Pavel Berkhin, and Lihong Li. Evaluation of explore-exploit policies in multi-result ranking systems. Technical Report MSR-TR-2015-34, Microsoft Research, 2015.

Emine Yilmaz, Manisha Verma, Nick Craswell, Filip Radlinski, and Peter Bailey. Relevance and Effort: An Analysis of Document Utility. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 91–100, 2014.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1201–1208, 2009.

Yisong Yue, Yue Gao, Oliver Chapelle, Ya Zhang, and Thorsten Joachims. Learning more powerful test statistics for click-based retrieval evaluation. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 507–514, 2010a.

Yisong Yue, Rajan Patel, and Hein Roehrig. Beyond position bias: examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 1011–1018, 2010b.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Yuchen Zhang, Weizhu Chen, Dong Wang, and Qiang Yang. User-click modeling for understanding and predicting search-behavior. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1388–1396, 2011.

Dengyong Zhou, Qiang Liu, John C. Platt, and Christopher Meek. Aggregating ordinal labels from crowds by minimax conditional entropy. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 262–270, 2014.

Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 307–314, 1998.

Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 17–26, 2015.

Masrour Zoghi, Tomáš Tunys, Lihong Li, Damien Jose, Junyan Chen, Chun Ming Chin, and Maarten de Rijke. Click-based hot fixes for underperforming torso queries. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, 2016.