Modeling User Behavior and Attention in Search

Jeff Huang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2013

Reading Committee:

Susan Dumais, Co-Chair

Jacob O. Wobbrock, Co-Chair

Eugene Agichtein

Oren Etzioni

Wanda Pratt

Program Authorized to Offer Degree:
Information School

University of Washington

Abstract

Modeling User Behavior and Attention in Search

Jeff Huang

Co-Chairs of the Supervisory Committee:
Principal Researcher Susan Dumais
Microsoft Research

Associate Professor Jacob O. Wobbrock
Information School

In Web search, query and click log data are easy to collect but they fail to capture user behaviors that do not lead to clicks. As search engines reach the limits inherent in click data and are hungry for more data in a competitive environment, mining cursor movements, hovering, and scrolling becomes important. This dissertation investigates how remotely collecting rich user interaction data in the form of mouse cursor activity can help researchers understand fundamental human behavior and improve the design of search engines. Specifically, mining cursor activity can improve upon state-of-the-art methods for scoring and ranking search results, and estimating where users are looking without eye-tracking. Descriptive analyses of cursor movements show how users move their cursor when they search to provide signals of relevance and explain reasons for abandoning a search. User models can be used to infer visual attention on the page to identify what content users are looking at, as well as compute the relevance and attractiveness of search results to the user. This implicit feedback given to the search engine can then inform the layout and content presented on the pages, or improve the ranking of search results.

This dissertation will demonstrate the following thesis: *users' mouse cursor interactions can be collected efficiently on the Web, used to understand users' search behaviors, and can be useful in the design of Web search engines.*

Table of Contents

List of Figures

List of Tables

Acknowledgments

Most students are lucky to have one good advisor, but I am fortunate to have had an amazing series of advisors and mentors. I thank my first advisor, Efthimis Efthimiadis who took me under his wing when I began my Ph.D. and started me on this path, and whose encouragement made anything seem possible. His loss was tragic but his foresight lead me to my advisors, Susan Dumais and Jacob O. Wobbrock. Sue shaped my research voice and gave me insight about experimental design and behavioral data, while Jake's support helped me acquire external funding and his academic advice laid the foundation for my career. I also want to thank Ryen White, whose door was always open and working alongside him taught me how to write and conduct rigorous research. His enthusiasm for cursor interactions provided me opportunities to conduct the experiments found in this dissertation. Finally, I thank Oren for adopting me into his research lab in Computer Science & Engineering and treating me as one of his own students, providing me with access to resources and collaborations from the KnowItAll group.

This dissertation was also made possible by the support provided to me by an amazing community of faculty, colleagues, friends, and collaborators. Many faculty at the University of Washington, both in the Information School and in Computer Science & Engineering contributed to my development as a scholar. Eugene Agichtein and Wanda Pratt provided guidance along the way as members of my committee. I appreciated Eugene's hands-on approach to my research, even while being on the other side of the country. David Hendry who served as the Ph.D. Program Chair was supportive throughout my years in both research and administrative matters. Luke Zettlemoyer's and Andrew Ko's doors was always open to offer advice and feedback. And lastly I want to thank Mike Eisenberg for his leadership in the school and going to bat for me during my job search.

I had the most wonderful cohort of Ph.D. students that started with me. I thank Eun

Kyoung Choe, Sheryl Day, Marisa Duarte, Miranda Belardi-Lewis, John Marino, Kristen Shinohara, and Shawn Walker for sharing these years with me, and making me feel like I was part of a family. I followed in the footsteps of the senior Ph.D.s and was especially inspired by Shaun Kane, Pedja Klasnja, and Peyina Lin. Several students joined me in authoring papers, and I learned a great deal working with them: Anna Kazeykina, Abdigani Diriye, Gifford Cheung, Katherine Thornton, and Thomas Lin. I also had the opportunity to work with enthusiastic undergraduate students, Christian Lee, Kevin Clarke, and Karan Goel. Finally, I thank Lydia Chilton, Anthony Fader, Jin Young Kim, Parmit Chilana, Luis Leiva, and Jessica Tran for helping me out with my work and being friendly faces to be around.

I also appreciate the mentorship from my internship hosts. Craig Miller and Danny Bain were crucial in their support of cursor tracking research, and allowed me to incorporate my experiment into the Bing search engine. Thomas Zimmermann and Nachi Nagappan gave me the opportunity to explore an amazing games dataset that joined my passion in games and data. Maria Katsova at Google provided me with formative career advice. Kuansan Wang helped me come up with a more principled approach to modeling searcher behavior. Thanks also to Georg Buscher to contributed substantially to my work from the Bing side.

During my Ph.D. I had the opportunity to nurture my entrepreneurial spirit, and I thank Ludvig Strigeus, Bo Lu, and Sizhao Yang for their inspiration. I thank Peter Schlichting and Mike Arcuri for joining me in building World Blender, and I thank Arnout Kazemier for working with me on our Observe.it dream.

I want to thank my mom for believing in me and being my number one fan. I thank my dad who gave me someone to look up to and inspiring me to pursue this Ph.D. like him.

Finally, I also want to thank Sarvesh Nagpal for helping me commit the cursor tracking code into Bing when things looked bleak in the last hour. Also a special thanks to Katherine Ye, who volunteered to meticulously review early drafts of many of my papers.

This dissertation was supported by a Facebook Fellowship, a Google Research Award, and work completed during internships at Microsoft. Any opinions, findings, and conclusions expressed in this dissertation do not necessarily reflect the views of those parties.

Dedication

This dissertation is dedicated to Efthimis Efthimiadis (1956–2010), my beloved advisor for the support and guidance he provided when I began this journey.

Chapter 1

INTRODUCTION

1.1 Motivation

People conduct billions of Web searches over billions of documents every day; it is an intrinsic part of our lives. We seek turkey stuffing recipes, names of former middle school acquaintances, clues about a skin rash, our favorite musician's website, and antiquated printer drivers. Getting the right information to the user can educate and improve productivity for billions of people.

From the Web searcher's perspective, searches are issued through a familiar system of typed queries and search results, via interactions with an intermediary Web browser. From the search engine's perspective, the goal is to find relevant Web pages that contain information that will satisfy the searcher. It is this "conversation" between a searcher and search engine that allows them to learn from one another.

Minimally, one of the search engine's primary jobs is to score documents using a multitude of signals, then return a ranked list of the highest scoring documents to the user. The ranking signals in Web search can be classified into three categories: content signals, structural signals, and Web usage [6]. The first search engines drew from techniques in traditional information retrieval, matching the user's query and the Web page using *content* signals. Scoring methods such as tf-idf [87] and its descendant Okapi BM25 [82] analyzed the document content in Web pages, scoring each Web page based on its content and how well this content matched the query.

As Web search engines evolved, signals from the *structure* of the Web gained importance, supplementing content signals through techniques such as mining the links between Web documents, namely 'link analysis.' PageRank [10] emphasized the incoming links to a page, scoring the page based on its relative importance, while HITS [60] computed two metrics for a website: its authority, the value of the page content, and its hub score, the value of its

outgoing links. Links thus became akin to reputation votes, so when one document linked to another, it was like vouching for it. Search engines adopted PageRank and HITS based algorithms, which became useful factors in ranking the documents.

Since then, Web search has evolved past document-query matching and hyperlink analysis (e.g., PageRank), to arrive at a stage where mining user interaction data is crucial to search result ranking and user assistance features. The past several years have been fruitful for using *interaction data* as implicit feedback from the user, providing relevance judgments for search results that can improve the ranking for future searchers [57]. One popular type of search interaction data is query logs, which includes people's search queries and often, what results they clicked. This information is easy to capture server-side and provides clean signals that relate to user interest that can be used for ranking search results. For example, if past users for a given query clicked on the third result more often than the second result, perhaps this is a signal that the third result should be ranked higher. Other applications assist the user, such as using past query reformulation to provide spelling corrections or query suggestions. Finally, these interactions can also be used for analytics to better understand their users and to learn about information seeking behavior (e.g. [54], [86]). More generally, search engines can use user behavior to infer the situational context which can improve search quality in later search sessions.

Query logs are commonly used as ranking signals by providing information about what people were searching for through the query text, and what they found appealing through the click records [56]. These logs have also been shown to be useful for a multitude of applications outside of ranking results. Query logs used in personalized search [78] customize search results to the preferences of the current user; they can help with spelling correction [2, 24], searcher modeling [18, 31, 37, 93], autocomplete queries, and offer query suggestions [9, 59].

1.2 Beyond Query Logs

Query logs provide useful information about searchers' queries and what they clicked, but some limitations of query logs have been noted in the literature [35, 36, 61]. One obvious limitation is that query logs do not typically contain interaction data besides the query

and click. In the time between the search query and the result click, users still interact with the page by scrolling and moving their mouse cursor around the page. Furthermore, a large proportion of queries do not result in clicks, possibly because the information the user seeks is already on the search results page, so clicks are unnecessary. Another possibility is that the query is infrequent so the search engine has not had a chance to collect much data in the past. A substantial portion of queries are infrequent so there are very few clicks or even no clicks from those queries [34]. Researchers know that search engines do poorly for these infrequent queries [28]. Thus, both infrequent and unclicked queries provide little information to the search system about what happened after the query.

In other data-intensive fields such as natural language processing and data mining, there has been evidence that collecting and mining additional data can be more useful than improving algorithms. A study by Banko and Brill [7] looked at various learning algorithms for disambiguating natural language. They showed that increasing the amount of data by 10-fold would boost even the worst algorithm's performance to become better than the best algorithm. A recent article titled "The Unreasonable Effectiveness of Data" highlighted the power of Web-scale data for machine translation [43]. Natural language involves the concept of context, which algorithms have trouble understanding. However, with enough data in the form of words and word sets, machine translation and speech recognition can be accomplished statistically. Rajaraman presented anecdotal evidence to argue that, "adding more, independent data usually beats out designing ever-better algorithms to analyze an existing data set" in an article titled "More data usually beats better algorithms" [79]. In one example, students in his data mining class competed to recommend Netflix movies given a set of previously rated movies from users, a typical machine learning problem. The team that applied a simple algorithm over combined Netflix data and IMDB data performed much better than the team that applied a sophisticated algorithm on just the Netflix data. A similar phenomenon may be occurring in Web search, where growing dependence on user-generated search logs makes it more important going forward to collect more independent data.

Having query logs at scale is essential because it provides good coverage over the queries (which are known to have a long tail), and allow for stratification over variables such as

geography, task type, topic, and user type. Web search already benefits by adding more of the same data—more query logs are generated every second, thereby improving the accuracy of inferences and analyses made from the data; this is akin to adding more Web pages to the index in the early days of ranking using content signals. However, there may be further benefit from adding different sources of data, akin to using structural signals to supplement content signals for ranking, a breakthrough in search ranking quality. New independent data can answer new sets of questions and provide information that cannot be inferred from existing data.

While query logs are the most direct and easily understandable indicators of user intentions and success, they can be further enriched with additional search interaction data. The insight comes from my experience directly watching a person search over their shoulder, in person. When I did this, I could tell a lot more information about the context of their search. And if the search engine could just "see" what they were looking at on their screen, how they were moving their cursor, and scrolling, and how they paused to think, the search engine could know much more about their intentions and underlying thoughts. So a search engine can put itself in the shoes of the users if they had that perspective, but that data is not being captured.

1.3 Mouse Cursor Interactions

Mouse cursor[1] interactions can be recorded with minimal intrusion using JavaScript, which is built into all modern Web browsers. Small snippets of JavaScript can be injected on the search page to record cursor interactions and send the data back to the search system using a GET request or HTML5 WebSockets. This technique can be deployed on a large scale without disrupting the searcher, or requiring any software installation. Thus, tracking cursor interactions can be cheaply instrumented in commercial search engines or in large-scale user studies on the Web.

Cursor movements can be recorded using JavaScript at fine levels of detail; the exact resolution depends on the Web browser and operating system. Cursor data can then be used

[1]From this point forward, I refer to the 'mouse cursor' simply as the 'cursor'.

to diagnose usability issues at an individual level when the session is replayed; furthermore, the data can be analyzed in aggregate to generate heatmaps. Other cursor activity such as scrolling, highlighting text, or non-navigational clicks may be explored. Behavior-centered analysis is also possible—some users move their cursor over text as they read, move the cursor slowly when thinking, or toss aside the cursor to reveal content that the cursor or tooltips were obscuring; these behaviors change the context in which they are interacting with the page. Since cursor location has been shown to correlate with where a user looks in some occasions [19, 22, 41, 83, 84], cursor data can also be used to understand and model search result examination behavior.

Besides tracking cursor interactions on the search page using JavaScript, lab studies provide another way of capturing people's cursor interactions (e.g., [20, 33, 46]). Lab studies involve asking users to enter a lab setting to perform an artificial task. They are useful for qualitatively understanding the user since they give richer insights into what is on the user's mind and control for the situational context. However, mining remotely-collected search data is a more scalable way to record user activity for automatically improving some search engine attributes like ranking and user assistance features. Thus, large-scale cursor data can complement other usability methods such as laboratory studies, surveys, etc. In this dissertation, I use both qualitative and quantitative methods to collect descriptive data about users' cursor behaviors, but also show how models for computing relevance and examination can be built purely from cursor interactions that can be collected remotely at Web scale.

1.4 Research Question

The first step I take is to determine whether cursor interactions can be useful at all in search. People move their cursors in irrelevant and unpredictable ways, and since cursor data is so noisy, we may want to know whether we can even extract useful information from the data. How can richer interaction data comprising cursor activity help us understand search behaviors and improve the design of search engines? This requires an exploratory approach to looking at cursor interaction data, and then some creative ways to build practical applications from the cursor data.

1.5 Contributions

Following from the research question, the contributions presented in this dissertation fall under three main categories. First is a method for tracking cursor interactions at scale (Section 3.2). Having this data at Web scale enables practical applications by providing coverage over many different types of queries and users. Second is a descriptive analyses of how people use their cursor when they search (Chapters 4 and 6). This teaches us about fundamental cursor behavior and informs the third category of contributions. This third category is of practical user models that utilize cursor interactions in two situations: for predicting the content that a user visually examines (Chapter 5), and computing relevance scores for documents in relation to a query (Chapter 7). I show that having cursor data can do these tasks better than current state-of-the-art methods that don't use cursor data.

Put together, these contributions show that users' mouse cursor interactions can be collected efficiently on the Web, used to understand users' search behaviors, and can be useful in the design of Web search engines.

1.6 Ambiguous Terminology

This section does not aim to define all the technical terms used within this dissertation. Instead, the definitions are for terms that can have multiple meanings, to specify how they are used in this document.

1.6.1 Cursor

I will use the word "cursor" as the pointer typically controlled by a mouse on a computer. In other text, authors have simply called it the "mouse" but I feel this is imprecise. The cursor can be controlled by many input devices, such as a touchpad or trackpad. However, referring to it as a pointer is also unsatisfying because that terminology is used only in the Windows operating system, and users of other environments are not familiar with that term. I acknowledge that the term "cursor" is still not ideal, because in other contexts it can also mean the text cursor or caret used for editing text. But I feel it is the best term for referring to the pointer controlled by an input device, typically the mouse.

1.6.2 Query and Click Logs

I will use the terms "query logs", "click logs", and "click data" almost interchangeably. The typical logs collected by search engines are essentially Web server logs, where for example on Google, clicks actually navigate to a URL on the search engine's Web server but are inconspicuously redirected to the target page. This redirection allows the search engine to know which items were clicked. Along with the server logs, the queries are automatically captured as they are part of the URL generated by the search engine after a query. Thus, this data contains queries, clicks, a timestamp, and other metadata together. In other literature, it is referred to as any of the above terms and sometimes even as "transaction logs" [54].

1.6.3 Gaze and Attention

In this document, I will make the assumption that the eye gaze position is also the location of visual attention. In practice, there may be a difference between a person's eye gaze and their visual attention. Since the eye gaze is tracked by an eye-tracking device, a person may actually visually be focused elsewhere; the eye gaze may also include saccades that are irrelevant to visual attention on a page. However, for simplifying the concepts, the two terms are regarded as equivalent, defined in this dissertation as the eye-tracker's estimations of the focal point of the user's gaze.

Chapter 2

RELATED WORK

Three lines of research relate to this dissertation. One focuses on tracking cursor interactions on the Web and in search, mostly in laboratory studies; this involves inferring user interest and intentions directly from the user's interactions with the search engine results page (SERP) or a Web page. The second area explores the relationship between the cursor position and gaze position in order to infer the user's visual attention based on cursor movements; most of these studies seem to suggest that cursor positions may be a good proxy for remotely tracking visual attention (i.e., eye gaze). Lastly, there have been a number of models of searchers' result examination behavior typically using query logs.

2.1 Cursor Tracking on the Web and in Search

The first research projects on cursor tracking involved modifying the Web browser in order to track the cursor interactions. They were done on a small scale in lab studies, and reported qualitative findings. In early work, Goecks and Shavlik modified a Web browser to record themselves browsing hundreds of Web pages [33]. They found that a neural network could predict variables such as the amount of cursor activity on the SERP, which they considered surrogate measurements of user interest. Claypool et al. [20] developed the "curious browser," a custom Web browser that recorded activity from 75 students browsing over 2,500 Web pages. They found that cursor travel time was a positive indicator of a Web page's relevance, but could only differentiate highly irrelevant Web pages. They also found that the number of clicks on a page did not correlate with its relevance, despite the intuition that clicks represent links that users found appealing. Hijikata [46] used client-side logging to monitor five subjects browsing a total of 120 Web pages. They recorded actions such as text tracing and link pointing using the cursor. The findings showed that these behaviors were good indicators for interesting regions of the Web page, around 1.5

times more effective than rudimentary term matching between the query and regions of the page. Shapira et al. [85] developed a special Web browser and recorded cursor activity from a small number of company employees browsing the Web. They found that the ratio of cursor movements to reading time was a better indicator of page quality than cursor travel distance and overall length of time that users spent on a page. More recently, Leiva and Vidal released an open source cursor tracking toolkit called "Simple Mouse Tracking" [67]; this toolkit allows website owners to set up their own cursor tracking systems on their website after installing it. Their toolkit has been used by numerous website developers to track the cursor interactions from their own users. Leiva and Vidal themselves have applied cursor tracking to cluster Web documents based on the cursor interactions on the page.

Some academic studies have analyzed the cursor in usability settings to learn about engagement on Web pages [3, 4, 92]. Atterer et al. investigated the usability of an online form through cursor analytics [4], Arroyo et al. presented visualizations of cursor trails to students who "proposed and prototyped redesigns reorganizing information where it could be easily found, and simpler to navigate," and Torres and Hernando described and offered a tool for websites to conduct usability on their own cursor-based investigations [92].Outside of academic literature, several commercial companies (e.g., ClickTale, Mouseflow, UserFly) have offered cursor tracking analytics for Website operators, typically presenting this data as heatmaps and replays. These services offer cursor interaction data to website developers to apply in usability analysis as they see fit.

In the search domain, Guo and Agichtein [39] captured cursor movements using a modified browser toolbar and found differences in cursor travel distances between informational and navigational queries, as defined by human labelers. Furthermore, a decision tree could classify the query type using cursor movements more accurately than using clicks. Guo and Agichtein also used interactions such as cursor movement, hovers, and scrolling to accurately infer search intent and interest in search results [40]. They focused on automatically identifying a searcher's research or purchase intent based on features of the interaction. In a more recent paper, Guo and Agichtein look at cursor interactions after the click onto the landing page and find that these post-click interactions (e.g., cursor movements, dwell time) correlate with document relevance [42]. They show that a post-click behavior model is more

effective than simply using dwell time for computing document relevance scores. Diriye et al. investigate the use of cursor interactions for classifying the reason why a user abandoned a query, whether it was because they were satisfied because they found the information they were seeking, or dissatisfied at the point of abandonment [27]. They showed that features such as cursor movement distance could distinguish the reason for abandonment, and that multiple additive regression trees (MART) were the best approach to the classification.

2.2 The Gaze-Cursor Relationship

Lab studies involving eye-tracking systems have been commonly used to track searchers' visual attention as they seek information on the Web. Past research has found a correlation between gaze and cursor positions [19, 22, 41, 83, 84] and that cursor movements can be useful for determining relevant parts of the Web page with varying degrees of success [33, 46, 85].

By applying a reading detection method, Buscher et al. [13] used gaze tracking features directly to infer user interest and show that this can yield great improvements when personalizing search. Buscher et al. also demonstrated the value of gaze information for building models that predicting salient regions of Web pages [12] while Cole et al. built reading models operating on gaze tracking data to investigate information acquisition strategies of the searcher for different search tasks [21].

One line of research examines the relationship between eye gaze and cursor positions in general Web browsing. An early study by Chen et al. [19] measures this relationship by recording 100 gaze and cursor positions from five subjects browsing the Web. They showed that the distance between gaze and cursor was markedly smaller in regions of the page that users attended. Liu and Chung [70] recorded cursor activity from 28 students browsing the Web and noticed patterns of viewing behaviors, including reading by tracing text with the cursor. Their algorithms were capable of predicting users' cursor behaviors with 79% accuracy.

Some assume that users look where they point with their mouse and that the cursor is a suitable substitute for eye-tracking. For example, the popular Web analytics service ClickTale notes on their website about their Mouse Move Heatmaps that "By aggregating

the mouse movements of thousands of visitors on a Web page, we create a comprehensive, visual representation of what visitors are looking at and focusing on within the page."[1] I show this assumption is misleading in many situations later in this dissertation.

More recent work has focused on the relationship between cursor and gaze in Web search. In a study involving 32 subjects performing 16 search tasks each [83, 84], Rodden et al. identified a strong alignment between cursor and gaze positions. They found that the distance between cursor and gaze positions was larger along the x-axis than the y-axis, and was generally shorter when the cursor was placed over the search results. Rodden et al. also observed four general types of cursor behaviors: neglecting the cursor while reading, using the cursor as a reading aid to follow text (either horizontally or vertically), and using the cursor to mark interesting results. Guo and Agichtein [41] reported similar findings in a smaller study with ten subjects performing 20 search tasks each. Like Rodden et al., Guo and Agichtein noticed that distances along the x-axis tended to be larger than the distances along the y-axis. They could predict with 77% accuracy when gaze and cursor were strongly aligned using cursor features. Rather than tracking the eye gaze with eye-tracking equipment, Lagun and Agichtein [63] presented a method to estimate gaze position by blurring the SERP and only revealing a region proximal to the cursor. They found that result viewing and clickthrough patterns agree closely with unrestricted viewing of results, as measured by eye-tracking equipment. Their method can also be used for large-scale investigations of search result attractiveness. Buscher et al. [15] also used scrolling to infer user interest and compared it to gaze tracking feedback. They found that scrolling behavior in connection with information on the browser's viewport could be as effective as gaze tracking feedback in during query expansion.

Finally, Navalpakkam et al. extend the work in this dissertation by investigating the gaze-cursor relationship on non-linear page layouts which, in search, may represent cases when information or advertisements are shown in the second column [74]. Furthermore, they predict eye-gaze using a non-linear model and identify particular regions of interest in addition to the coordinates.

---

[1] http://www.clicktale.com/products/heatmap-suite/mouse-move

2.3 Searcher Models from Query Logs

The third area of related research involves modeling the searcher, typically using graphical models such as Bayesian networks. Searcher models attempt to infer the searcher's state as they issue queries, examine the search results, and perhaps click on some results. These models attempt to determine which search result a user has examined, whether a clicked search result was satisfying, and other unobserved states. Searcher models have been developed from two main hypotheses that are commonly used as assumptions in the models. Since users are biased towards clicking search results that are higher ranked [56], the **examination hypothesis** is used to isolate a search result's attractiveness from its position. This hypothesis, originally formulated in Richardson et al. [81], states that the likelihood that a user will click on a search result is influenced only by 1) whether the user examined the search result and 2) its attractiveness. In other words, a user must examine a search result before potentially clicking that result. By making this assumption, a search result's attractiveness can be computed independent of its position in the ranking, i.e.,

$$P(C_i = 1) = P(E_i = 1)P(C_i = 1|E_i = 1),$$

where the term $P(E_i = 1)$ is the position bias and the term $P(C_i = 1|E_i = 1)$ is the search result's attractiveness.

To determine whether a user examined the search result, some searcher models (like the Cascade Model) draw from the **linear traversal hypothesis** [23] that designates which search results a user has examined. The linear traversal hypothesis states that a user always examines search results sequentially and goes from top-to-bottom on the SERP. A user decides whether to click a result before examining the next result, preventing scenarios where the user returns to a higher-ranked search result after passing it by. Therefore, if users do not examine a particular search result, they will not examine any search results below it, i.e.,

$$P(E_1 = 1) = 1,$$
$$P(E_{i+1} = 1|E_i = 0) = 0.$$

While the original Cascade Model stipulated that once a user clicked, they would no longer examine any search results, extensions of this hypothesis have removed this assumption. The Dependent Click Model [38] allows for query sessions to comprise multiple clicks, by using a parameter representing the probability that the clicked document is not relevant and the user returns to examining more search results. The Click Chain Model [37] and Dynamic Bayesian Network Model (DBN) [18] both extend this by adding an additional parameter representing the probability that a user abandons a query session without clicking, thus circumventing the side-effect of the linear traversal hypothesis that users are assumed to examine every search result in abandoned queries.

Other searcher models such as the Partially-Observable Markov Model [93] and User Browsing Model [31] avoid the linear traversal hypothesis entirely, allowing the user to jump between search results non-sequentially in their examination. However, these models have more parameters representing the probabilities of transitions between search result positions which make inference more difficult, especially when there are fewer query sessions with clicks from which to learn.

2.4 Extending Related Work

This dissertation extends the existing literature in several ways. One is by introducing efficient techniques to enable remote cursor tracking at scale on the Web, particularly applied to the search domain, and demonstrating their practicality by deploying the tracking widely on the Bing search engine. A second is going beyond analyzing the relationship between gaze and cursor, and instead using cursor interactions to predict gaze positions (which Navalpakkam et al. [74] build upon). The third is offering a method for extending existing click models built from query logs using cursor interactions, and showing that this can improve the relevance scoring of search results. These methods are scalable, and I show in this dissertation how they can be applied to enhance search systems at Web scale.

Chapter 3

METHODS FOR CURSOR TRACKING[1]

This dissertation presents results gathered through two methods. One is a lab study involving an eye- and cursor-tracking device, and the other is a large-scale deployment of cursor tracking. The lab study provides information about where people are looking when they use the cursor, obtains demographics information, and provides data collected in a controlled setting. The large-scale deployment gives us a larger population from which to compute experimental results and show aggregate patterns of cursor usage.

3.1 Eye-Tracking Lab Study

The gold standard of determining where people are looking is an eye-tracking device. Eye-tracking is a technique for recording where people are looking on the screen using a special eye-tracking camera. The eye-tracking device projects a light into the eye of the user and captures the image at high frequency, which is then processed to determine the direction of the gaze. Eye-tracking allows researchers to figure out what parts of the screen capture a user's attention while they are interacting with the site, which informs how site designers should modify the layout or content of the page. It is a prominent technique in the field of Web usability, and many companies spend lots of money and time conducting eye-tracking studies.

In this dissertation, eye-tracking data was collected using a Tobii x50 eye tracker with 50 Hz tracking frequency and an accuracy of 0.5° visual angle (corresponding to 16 pixels in our setting) on a 1280×1024 resolution 17 inch monitor (96 dpi) and 1040×996 resolution Web browser. Cursor and gaze coordinates were collected from 38 subjects (21 female, 17 male), recruited from a user study pool[2]. They ranged in age between 26 and 60 years

[1] Portions of this chapter are published in CHI 2012 [49] and CHI 2011 [51]

[2] The eye-tracking trials were run by Georg Buscher during his summer internship at Microsoft Research.

($M = 45.5$, $SD = 8.2$), and possessed a wide variety of backgrounds and professions. Two subjects had incomplete data due to technical issues and were dropped from the analysis.

Each subject completed 32 Web search tasks (information needs) on the Bing search engine in a randomized order. Half of the tasks were navigational (i.e., they had to find a specific Web page) and half were informational (i.e., they had to find factual information). Each task started with a description of what subjects should look for on the Web. They had to start searching with a predefined query for each task (e.g., task = "What are some side-effects of Ibuprofen?", predefined query = "ibuprofen side effects") that were generated from the task, but were then free to interact with the search results, browse the Web and search further. The browser cache and cookies were cleared after each subject to prevent subjects from noticing previously viewed pages (from hyperlinks turning purple) and search engine personalization effects. Each subject took about one hour to undertake an eye tracker calibration phase at the beginning, complete all 32 search tasks, as well as fill in a demographics questionnaire at the end.

Gaze and cursor positions were recorded for each SERP as well as subsequent Web pages (i.e., pages visited after clicking on a search result). In total, the eye-tracking data for 1,210 search tasks included 1,336,647 gaze positions and 87,227 cursor positions (whenever the subject moved the cursor). Additional details about the experimental procedure are described in Buscher et al. [14]. Gaze-specific findings on this data set, unrelated to cursor features, have been reported elsewhere [14, 30].

In the logs, the gaze positions were recorded approximately every 20 ms, whereas cursor positions were recorded approximately every 100 ms. Gaze positions are estimated by the eye-tracker; because of the saccades (rapid movements) of the eye, recording at a higher frequency can give more accurate positions. On the other hand, the cursor position is an exact value and recording at 10 Hz is sufficient—the cursor is not radically changing directions at sub-second speeds in a way that we cannot later interpolate its position[3].

Since cursor and gaze events did not necessarily have identical timestamps, a gaze position was interpolated for every cursor position. Interpolation was performed by computing

[3]I tested different frequencies of interpolation and found negligible differences.

gaze x and y coordinates weighted by the coordinates of the nearest gaze coordinates before and after the cursor position. For example, the interpolated x-coordinate for eye gaze is computed as,

$$x_i = x_0 + (x_1 - x_0)\frac{t_i - t_0}{t_1 - t_0} \tag{3.1}$$

where t_i is the time for the corresponding cursor position, x_0 is the gaze's x-coordinate preceding the cursor position, recorded at time t_0, and x_1 is the gaze's x-coordinate following the cursor position, recorded at time t_1. The interpolated y-coordinate was computed the same way, substituting x for y in the above equation. To reduce interpolation inaccuracies due to noise from the eye-tracker, cursor positions were only captured if they occurred between gaze positions that were at most 100 ms apart.

Bringing subjects into an eye-tracking lab creates inherent limitations. Subjects may behave differently in the lab with a camera monitoring their gaze than in a natural setting. The study takes place in an artificial setting where experimenter bias can surface (e.g., the Hawthorne Effect [64]). While every attempt was made to ground the search tasks in realistic information needs, the tasks might not be a representative sample. The users are unlikely to take breaks in the lab or multi-task by doing other things at the same time, factors which may affect cursor behavior outside of the lab. Additionally, though SERPs provide a controlled environment for our studies (and there are already several applications if we focus on that domain alone), more work is also needed to generalize this research beyond SERPs to any Web page.

3.2 Remote Cursor Tracking

Gaze-tracking studies with participants present in the laboratory can provide detailed insights but on a small scale. Studying such behaviors in laboratory settings is limited in terms of what inferences can be made. On the other hand, cursor movements can be collected remotely at Web scale through a different method. Tracking cursor movements across large numbers of users can provide a rich new source of behavioral information to understand, model, and satisfy information needs. While cursor movements correlate with eye gaze [19, 40, 83, 84], and may therefore be an effective indicator of user attention, having

such data at scale can help researchers understand different cohorts of populations and learn models for improving the search experience.

To collect this data, I instrumented a cursor tracking script to collect this new interaction data by inserting a tiny bit of compressed JavaScript into the Bing search engine's search results page. The script embedded in the HTML source would remotely track cursor activity, including movements, scrolling, hovering, the location of the key user interface elements on the page, and clicks. The code was activated whenever the Web page it sits on is viewed, and it ran for every visitor to that page without requiring any software installation and without disrupting the user.

When logging any additional type of user interaction data beyond clickthrough, a trade-off has to be made between: (i) level of detail (e.g., temporal and spatial resolution), (ii) the impact of any additional JavaScript code on page weight, page load time, and therefore the user experience, which can be sensitive to even small increases in load time, and (iii) the amount of data transferred (and hence bandwidth consumed) between the client and the remote server, as well as log volume created on the backend server. I sought to minimize the data gathered and transmitted to avoid adversely affecting the user experience with delays associated with log data capture and data transmission to the remote server. The script negotiated a trade-off between these dimensions by: (i) reasonably coarsening the log resolution, (ii) compressing the JavaScript, and (iii) compressing the log data as well as using a buffering approach for its transferal via AJAX to the backend server.

The embedded script had a total size of approximately 750 bytes of compressed JavaScript, which had little effect on the page load time (it was less than one percent of the search results page file size). The script recorded users' cursor interaction within the Web page's borders relative to the top-left corner of the page. Since cursor tracking was relative to the document, the script captured cursor alignment to search engine results page (SERP) content regardless of how the user got to that position (e.g., by scrolling, or keyboard). Therefore this approach did not constrain other behaviors such as scrolling or keyboard input.

In previous cursor tracking studies, cursor position was recorded at particular time intervals, such as every 50 milliseconds (ms) [41] or every 100 ms [83]. This is impractical

at a large scale because of the large amount of data to transfer from the user's computer to the server. One alternative is to record events only when there is activity, but this is still problematic because even a single cursor movement can trigger many cursor movement events. When a user moves their cursor from left to right, it can generate a hundred coordinates—too many to constantly send over the network in terms of data and bandwidth. My solution was to compress this data but still retain the important points using a technique called pause-based polling (described in more detail in the next section). I recorded all clicks since they were less frequent. The events were buffered and sent to a remote server every two seconds and also when the user navigated away from the SERP through clicking on a hyperlink or closing the tab or browser; this was typically 1–3 kilobytes of data sent with a JavaScript GET request, so overall this approach was highly scalable. It was unlikely that the user would not notice any delay in the page load or bandwidth. The pseudo-code below summarizes the logic in my algorithm.

```
onCursorMove:
  loc = getCursorPos()
  wait(N milliseconds)
  if loc == getCursorPos(): // cursor stable for N ms
    buffer.add(time,loc,getRegion(loc),“position”)
onCursorClick:
  buffer.add(time,loc,getRegion(loc),“click”)
onTick, onPageClose:
  send(buffer)
  clear(buffer)
```

A server-side process aggregated data from multiple pageviews belonging to the same query (e.g., from returning to SERP using the browser "back" button or viewing multiple result pages), to facilitate query-level in addition to pageview-level analysis. I identified regions that the cursor hovers over using attributes in the HTML, and use two such regions in subsequent analyses (result rank, link id). The large volume of data collected enabled me to focus on aspects of how searchers use their cursors on SERPs. For this purpose, I use the query-level data, comprising all clicks and cursor movements for a query session[4]. In

[4]Query session is defined as a query, along with activity taking place following that query including if

addition to the location of cursor positions, I summarize the total amount of cursor activity for a query using cursor trails (i.e., complete contiguous sequences of cursor movements on the SERP). As shown later, these trails are useful in situations where no clicks are observed.

Data were accumulated from two samples of users. One was a random sample of Microsoft employees' searches on Bing between May 12, 2010 and June 6, 2010. In total, this dataset contained 7,500,429 cursor events from 366,473 queries made by 21,936 unique cookies; the actual number of users may have been fewer since multiple cookies could belong to a single user. Although employees of Microsoft may not be representative of the general Web searcher population in some respects (e.g., they were more technical), their interaction patterns can still provide useful insights on how SERPs are examined. A second sample of log data was gathered using the same algorithm over a period of 13 days between May 26, 2011 and June 7, 2011 during an external experiment on a small fraction of user traffic, primarily from English-speaking countries. Samples were drawn by user, storing every query from each user in the dataset. In total, this dataset comprised around 1.8 million queries, averaging eight queries per searcher (median = 3 queries). This second sample was used for the searcher modeling experiment described in Chapter 7.

A summary of the differences between the methods is below in Table 3.1.

	Eye-Tracking Lab Study	**Remote Cursor Tracking**	
		Sample 1	Sample 2
Coordinates Collected	Eye gaze and cursor	Cursor	Cursor
Number of Users	38	21,936	230,000
Period	Summer 2009	Summer 2010	Summer 2011
Number of Queries	1,216	366,473	1,800,000
Sections Using Method	Chapters 4, 5	Chapter 6	Chapters 4, 7

Table 3.1: A summary of the differences between the two methods of data collection used in this dissertation.

they user navigates to a search result link and returns.

3.2.1 Optimal Cursor Sampling

While recording cursor interactions can be done without disrupting the user directly, the overhead of recording the cursor trail and transmitting this data over the network can be substantial. To compress the cursor data, it is important to understand that not every cursor coordinate has equal value. Furthermore, different compression algorithms are more suited to different goals or applications.

Applying compression can optimize for reducing data size and therefore bandwidth between the client and the server, having minimal effect on client-side performance, or reproducing the original data the most accurately. In collaboration with Luis Leiva, I looked at 10 compression algorithms, half of them lossless and the other half lossy. Lossless compression algorithms are capable of reducing the data size while also reproducing the original cursor activity; we found that among lossless methods, LZW [97] performed well. However, lossy compression can offer greater gains in compression as well as improved performance, at the expense of exactly replicating the original trail. A detailed treatise into the exploration and evaluation of the 10 compression algorithms is provided in Appendix A.

We evaluated 5 lossless and 5 lossy compression algorithms over two datasets (one from a live website, another from a lab study), computing client-side performance, space savings, and how well a lossy algorithm can replicate the original cursor trail. The results showed that different compression techniques may be suitable for different goals: LZW offers superior lossless compression, but lossy algorithms such as piecewise linear interpolation and dispersion-threshold identification offer better client-side performance and bandwidth reduction.

Lossless compression like Huffman and LZW encoding is bad for performance, and provides moderate compression. Lossy compression techniques like interpolation, polling, and distance-based sampling are fast but are not necessary the best points to capture. For the same compression levels, the naive method of time-based polling replicated the original trail worse than recently-developed methods. Piecewise linear interpolation and dispersion-threshold identification could reproduce the original trail better at the same levels of compression. Pause-based polling, which has been previously deployed on a large scale, remains

superior in terms of client-side performance; this compression method is especially suitable for wide distribution when some users may be using slower computers.

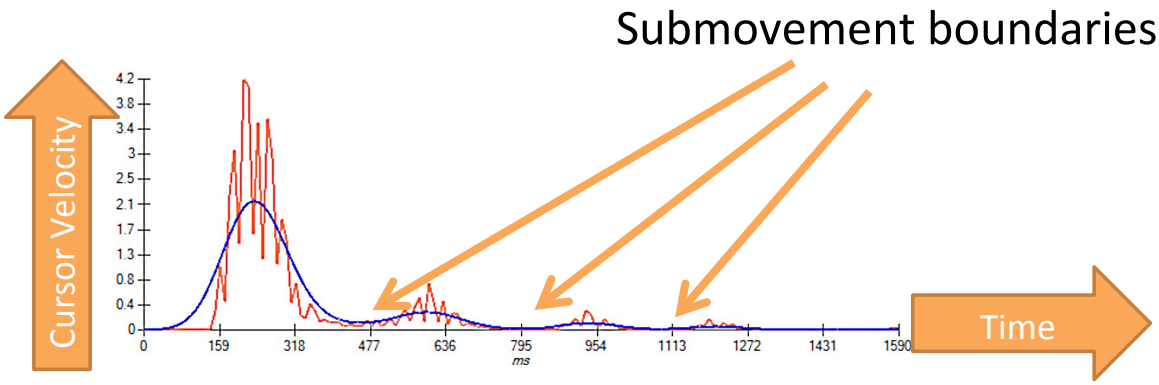

Figure 3.1: Plot of a single cursor movement towards a target generated by FittsStudy [100], illustrating the large initial ballistic movement followed by smaller corrective movements. The short pauses between movements are reasonable places to capture a cursor position.

The pause-based polling approach that provided a reasonable trade-off between compression, performance, and capturing key points is informed from insights about human factors. When humans move pointing devices, we make large ballistic movements and then smaller corrective ones (Figure 3.1). At these submovement boundaries, there are very short pauses, so to record key points during a movement, the script recorded cursor positions after a movement delay, which can be easily interpolated to recreate the intermediate points if needed. Basically, this method captures the key points that bound intentional movements. From experimentation, I found that recording cursor positions only after a 40 ms pause provided a reasonable trade-off between data quantity and granularity of the recorded events. This approach recorded sufficient key points of cursor movement, e.g., when the user changed directions in moving or at endpoints before and after a move. Occasionally, points within a longer movement were also captured if the user hesitated while moving.

Chapter 4

RELATING CURSOR AND EYE GAZE[1]

This chapter provides descriptive analysis of the behaviors people engage in with their cursors in relation to how they visually examine the Web pages. The findings presented are from data obtained through the eye-tracking lab study (Method from Section 3.1).

4.1 Gaze-Cursor Alignment

People navigate the Web as part of their daily lives by looking, finding, reading, pointing, and clicking. Because the flow of using a site is such a fundamental experience for so many users, usability professionals and site designers seek to optimize the experience by analyzing which parts of the page grab a visitor's attention, and what information users read on the page. To achieve this, they conduct laboratory studies using eye-tracking equipment to track users' gaze while they navigate the site. Research studies have shown gaze to be useful in determining what people are reading from their fixation [13, 80], in determining salient regions of Web pages [12], and in identifying the effects of changes to SERPs [25]. But could the cursor be a cheap and scalable alternative to eye-tracking for the same purposes? If so, cursor tracking systems can be deployed on any website, replacing eye-trackers.

Prior work has shown that gaze and cursor are correlated but a further goal is to determine *when* gaze and cursor are aligned. We want to know when the cursor position is a good proxy for gaze position and the effect of various factors such as time, user, cursor behavior patterns, and search task on the gaze-cursor alignment. This analysis will contribute to knowledge of how people use their cursor and how they examine Web search result pages. One starting point is to investigate whether the overall cursor trail can give a good approximation to visual attention with better accuracy than simply using the corresponding cursor position.

---

[1]Portions of this chapter are published in CHI 2012 [49] and WSDM 2012 [16]

In order to understand the relation between attention and cursor position, I will delve into the relationship between eye-gaze and cursor coordinates to find how not only *when* the cursor could be used as a proxy for attention, but later in this dissertation also *where* people were looking (Chapter 5).

The gaze-cursor alignment can be examined from the eye-tracking study of 36 subjects and 32 search tasks. Plotting the distance on the x and y axes between the gaze position and cursor position at each point in time, Figure 4.1 illustrates that there is some correlation between gaze and cursor positions, but with substantial variation. Other studies measuring gaze-cursor alignment [41, 83] have presented evidence of alignment with charts nearly identical to Figure 4.1.

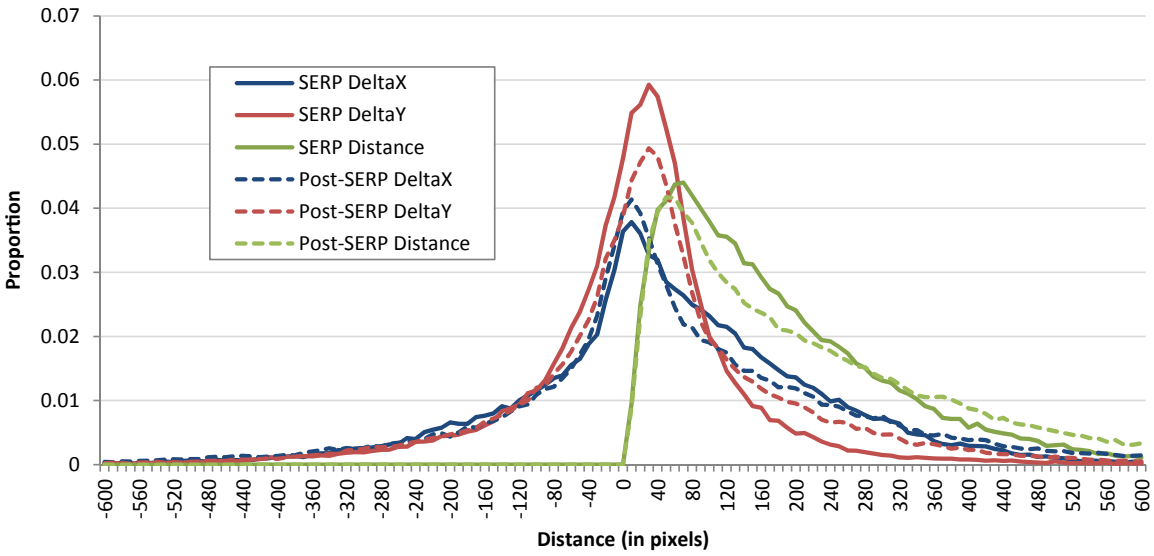

Figure 4.1: Δx, Δy, and Euclidean distance plotted in a frequency distribution for the search results page, and for (Post-SERP) Web pages that users land on after clicking a link on the SERP.

Figure 4.1 shows the frequency distribution for different values of Δx (distance between cursor x-coordinate and gaze x-coordinate), Δy (distance between cursor y-coordinate and gaze y-coordinate), and Euclidean distance between cursor and gaze coordinates, i.e., $\sqrt{(\Delta x)^2 + (\Delta y)^2}$. The solid lines show the distances for SERP pages. As can be seen, cursor and gaze positions are quite similar for both x and y values, their deltas peaking

near 0, when the gaze and cursor positions are in the same place. The mean Euclidean distance between cursor and gaze is 178 px (σ = 139 px) and the median is 143 px. The most common offset for the cursor is +3 px (to the right) for the x-coordinate and +29 px (lower) for the y-coordinate. That is, the cursor is most likely to be just below where the user is focusing with their eyes. In aggregate, the differences are greater in the x than y direction (average 50 px in the x direction and 7 px in the y direction), similar to other studies [41, 83] so this means the cursor tracks the gaze better along the vertical dimension on a screen. Two explanations for the difference between Δx and Δy are: (i) users may place the cursor to the left or right of their gaze to prevent it from obscuring the text as they read up or down, and (ii) computer screens are usually wider than they are tall, offering more horizontal space for the cursor.

The dotted lines in Figure 4.1 represent post-SERP landing pages. Distances between the gaze and cursor on the landing pages were greater than those on the SERP (215 px vs. 178 px), perhaps due to greater variance in the layout and the content of those pages, as has already been suggested by earlier gaze analysis [5]. Thus the cursor is a better proxy for user attention on the SERP than post-SERP pages. This result implies that cursor tracking may be particularly useful technique in search compared to typical Web pages.

4.2 Cursor Behavior Patterns

Cursor interaction spans a variety of behavioral patterns [3, 22, 70, 83, 84] including reading, hesitating, highlighting, marking, and actions such as scrolling and clicking. Identifying these behaviors is a prerequisite to understanding what meaning the cursor interactions convey.

Several studies have reported different behavioral patterns when using the cursor. Rodden et al. observed several types of cursor behavior: neglecting the cursor while reading, using the cursor as a reading aid to follow text, and using the cursor to mark interesting results [84]. Liu and Chung also noticed patterns of cursor behaviors, including reading by tracing text [70]. Arroyo et al. observed users hesitating and reading with the cursor [3]. Mueller and Lockerd found users hesitating before clicking and resting the cursor on white space [22]. Finally, Claypool et al. observed the following cursor behaviors: ignoring the

cursor, examining the page using the cursor, following the text with the cursor, and using it to interact with the page or browser [20]. The taxonomy I have developed categorizes cursor behavior patterns in a manner resembling those reported by Claypool et al.

Informed by prior work and our own qualitative observations of user interactions with the SERP, I separated cursor behaviors into four categories:

Inactive cursors are not moving and are ignored by the user for some time.

Reading cursors are used to follow the text while the user is reading the page.

Action cursors are used when the user is about to perform an action (click on a link, edit the query in the search box, drag the scrollbar, etc.).

Examining cursors move around while the user is examining the page, not including time spent in 'reading' or 'action.'

Each of the previously mentioned studies discusses the cursor behavior qualitatively and most do not have corresponding gaze data. This study involved more subjects and search tasks than the largest prior study, facilitating a quantitative analysis of the cursor behaviors.

4.2.1 Distinguishing Between Cursor Behaviors

A heuristic-based method can classify the different cursor behaviors to determine the current behavioral intent, such as: marking content with the cursor, moving the cursor away to read, using the cursor to perform an action, etc. This method entails iteratively examining replays of the recorded interaction behaviors, deciding which behaviors belong to each category, classifying the behaviors using simple rules, and finally comparing the classified behaviors with the judged behaviors. The process is ad-hoc yet flexible to develop a simple classification scheme that captures the essence of each behavior type.

I developed classification rules informed by watching replays of the interaction in a query session. The cursor is considered inactive if the user leaves it in one location (pausing it) while they examine the page. 'Inactive' is defined as the cursor staying idle for at least one full second. The behaviors occurring when the cursor's position is active can be classified in

three ways. As Claypool et al. noted, users may be using the cursor to help examine or read the page [20]. However, the cursor also serves the purpose of interacting with elements of the page or with the Web browser, typically by clicking on controls on the browser or Web page. I classify these 'action' behaviors as those occurring in the one second preceding a click. The remaining interactions were classified as either 'examining' or 'reading'. Reading was typically defined as users following the text horizontally (an observed pattern in Rodden et al. [84]), since the text in a snippet or advertisements goes from left to right. The movement to the right was not enough to classify as reading, since the cursor may be moved to the right for many reasons. I arrived at three rules for an interaction which, if met, would label all cursor positions within that timeframe as 'reading': the cursor could not have moved more than 50 px vertically (about 3 lines of text); the cursor must have moved at least 150 px to the right (the length of a handful of words); and the cursor must have moved back to the left at least 50 px. The remaining interactions, in which the cursor was neither inactive nor reading nor performing an action, were labeled as 'examining.' The final result was five cursor patterns (including clicking which was treated as an instantaneous action) exhibiting behaviors that warranted further analysis. I studied gaze-cursor alignment in these behaviors next.

4.2.2 Cursor Behavior's Effect on Alignment

I noticed that as people get more engaged with the cursor (as the behavior goes from Inactive to Examining to Reading to Action to Click), the distance between where they are looking and pointing decreases. By identifying when they are more engaged on the page, we can then notice that the eye is looking somewhere near the cursor. This data shows that the cursor works differently from the eye, because we know from human physiology that the eye remains fixated on content it finds attractive [29], but the cursor is fixated on one spot when the user's focus is usually somewhere. So it's the opposite effect—an idle cursor means the user is probably not paying attention there.

For each of the five cursor behaviors, I computed the distance between gaze and cursor. Table 4.1 summarizes the proportion of time spent in each cursor behavior and the

corresponding median distance between gaze and cursor. As expected, gaze and cursor are further apart when the cursor is inactive, since the eye is still roaming the SERP—233 px. Alignment is much closer when the cursor is being actively used to examine, read, or perform an action. The median distance when examining the page using the cursor is 167 px, while the alignment is closer when using the cursor to read—150 px. When the subject is moving the cursor to perform an action involving a click, alignment is extremely close—77 px, and even closer at the actual click—74 px. These findings agree with Hauger et al. that being in motion increases the alignment between cursor and gaze [44].

Behavior	Inactive	Examining	Reading	Action	Click
Total Time	58.8%	32.9%	2.5%	5.7%	-
Distance	233 px	167 px	150 px	77 px	74 px

Table 4.1: The median gaze-cursor distance for different cursor behaviors. The total time is summed across all subjects and search tasks.

During the study, the cursor was inactive a total of 18,554 seconds (58.8% of the time), representing time the subject may have been looking through the page without moving the cursor or just pausing for a few moments to read or think. Inactive time is more than the combined time spent examining, reading, and performing an action on the page, meaning that a substantial period exists in which it is difficult to predict the gaze position. Anecdotally, some people believe they rarely use the cursor while examining Web pages. But in aggregate, a large portion of time is still spent actively moving the cursor, most of which does not produce an action.

Claypool et al. remarked, "Some users move the cursor while reading the window text or looking at interesting objects on the page, while others move the cursor only to click on interesting links" [20]. I quantified individual differences in gaze-cursor alignment and cursor behavior; Appendix B presents the duration of each cursor behavior and gaze-cursor alignment for each subject. The distances were macro-averaged over search tasks for each subject to mitigate the effect of a subject spending more time on certain search tasks.

Leaving the cursor idle from 50% to 79% of the time was common, like Subject 29 who

left the cursor inactive during the majority of the time. Other subjects actively moved the cursor while examining the page; Subject 12 spent 55% of their time examining the page with the cursor, and a mere 29% of their time idling the cursor. Reading behavior comprised 2% or less of total search time for more than half the subjects (22 of 36), who exhibited nearly no reading behavior (per the definition of this behavior that I used). At the other end of the spectrum, Subject 9 spent 8% of their time reading with the cursor. Turning to gaze-cursor distance, some subjects had poor alignment (Subject 33) when not performing an action. While in the aggregate, subjects had stronger alignment when the cursor was active, Subject 18 had essentially unchanged alignment between inactive, examining, and reading behaviors. These differences show that individuals vary substantially in their cursor behavior usage and gaze-cursor alignment. I initially thought that perhaps the variation in users' gaze-cursor alignment could be explained by choice of cursor behavior, but for each cursor behavior, gaze-cursor alignment still varied substantially among the subjects.

4.3 Alignment Effects of User and Task

In addition to the aggregate analysis in the previous section, gaze-cursor alignment can also be viewed through slices of user and search task.

Different people behave differently on the Web in the queries they issue [96], how they gaze at the page [5, 30], and how they interact with the page [41]. People visually examine Web pages differently [12], and have individual styles in their control of the cursor. In essence, there are different types of "personalities" of how people move their cursors. Adopting the terminology from prior eye-tracking research [5] users can be classified as exhaustive, representing users who like to review all the information before making a decision, or as economic users who jump on the first link that looks attractive or potentially satisfying[2].

Past findings have shown that users behave differently depending on search tasks (e.g., between informational and navigational task types [25]). Guo and Agichtein analyzed gaze-cursor alignment [41] on a smaller subject pool and found differences in user (up to twice the average distance between gaze and cursor for some users) and search task type (navigational

[2]In the field of information seeking behavior exhaustive users are similar to Explorers, while economic users are similar to Navigators [98].

vs. informational). Here I explore whether search task or individual differences (including age and gender) have a stronger effect on alignment. Larger variances in alignment would indicate that a search system might have difficulty predicting alignment for users or queries that have not occurred before.

To study alignment differences among subjects, I macro-averaged the gaze-cursor distance across their queries (i.e., took the average of the average for each of their queries, thus giving equal weight to each query). The results in Figure 4.2 show that subjects are fairly distinct in terms of gaze-cursor alignment. Some keep their cursor within about 130 px of their gaze, while others average about 280 px. The standard deviation representing the variation among subjects is $SD = 33.9$. I checked for gender differences in gaze-cursor distance using a two-tailed t-test and found no statistically significant effect ($t(34) = 1.31, p = 0.20$). There was a Spearman correlation of $\rho = 0.22$ between age and gaze-cursor alignment, but this was also not statistically significant ($N = 36, p = 0.18$). Therefore, I conclude that whether the subject tracks their gaze closely with their cursor is more likely to stem from personal habits rather than age or gender.

Next I looked at the average differences in gaze-cursor alignment for different search tasks. Subjects were given predefined queries to begin with, filtering out reformulated queries since reformulations may be reflective of personal style or individual search skills. A search task is therefore represented by a single query in this analysis. For each search task, I first macro-averaged the gaze-cursor distance across subjects to normalize the data (i.e. averaged the distances within each subject for that task), then took the mean of this value to compute the average distance for each task. The search task averages are shown in Figure 4.3, which reveals that there are modest differences between tasks. The alignment distance ranges from about 150 px to about 220 px, and the standard deviation among search tasks is $SD = 20.2$.

The click entropy for each query can provide insight into the attributes of the search tasks as a measure of result click diversity [91, 96]. The click entropy is computed as the Shannon entropy (H) of the click distribution on the search result links (Equation 4.1), which may correspond with gaze-cursor alignment. It is possible that gaze and cursor align differently depending on characteristics of the query; typically, queries with low click entropy are more

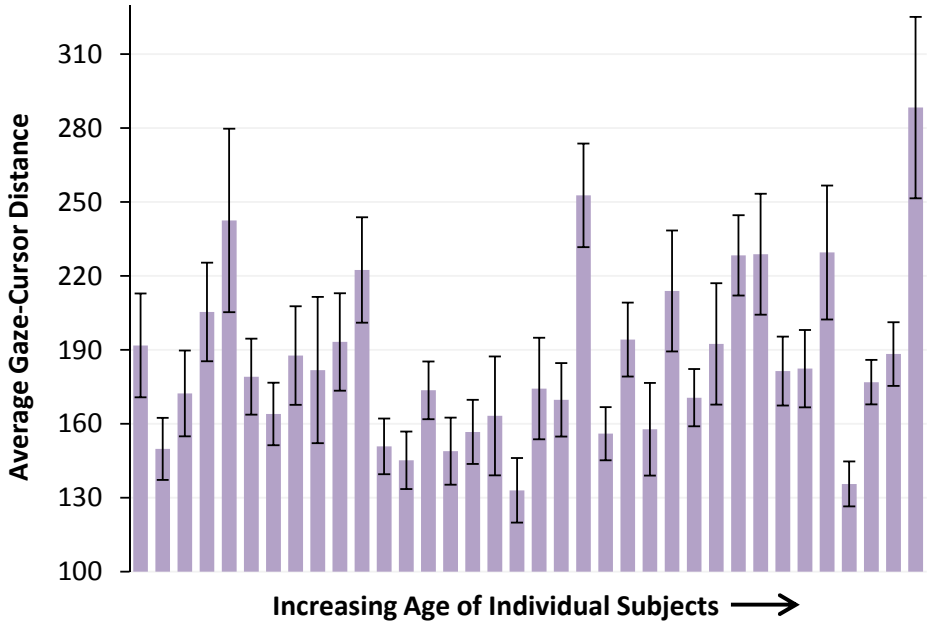

Figure 4.2: The average gaze-cursor distance for each subject with error bars representing the standard error. The variance between subjects is high ($SD = 33.9$). Subjects are sorted by ascending age so that oldest subjects are on the right.

navigational. The past year's search logs of the Bing search engine were used to compute the click distributions for each query. Five queries had not appeared in the search logs in the past year and thus had unknown entropy. From the remaining queries, I found no Spearman correlation between click entropy and gaze-cursor alignment ($\rho = 0.01, N = 27, p = 0.96$).

$$H(q) = -\sum_{u \in R} P(c_u|q) \times \log P(c_u|q) \tag{4.1}$$

where u is a URL in the set of search results R, and $P(c_u|q)$ is the probability that URL u was clicked following query q.

The standard deviation in gaze-cursor alignment across means for different subjects is higher than across means for different search tasks. A Levene's test for homogeneity of variance shows that the differences in variance are statistically significant (Levene statistic[3]

[3]Levene's test is typically used to compare the variance between two sources rather than between two sets of macro-averaged values, so its use here is admittedly a little unusual.

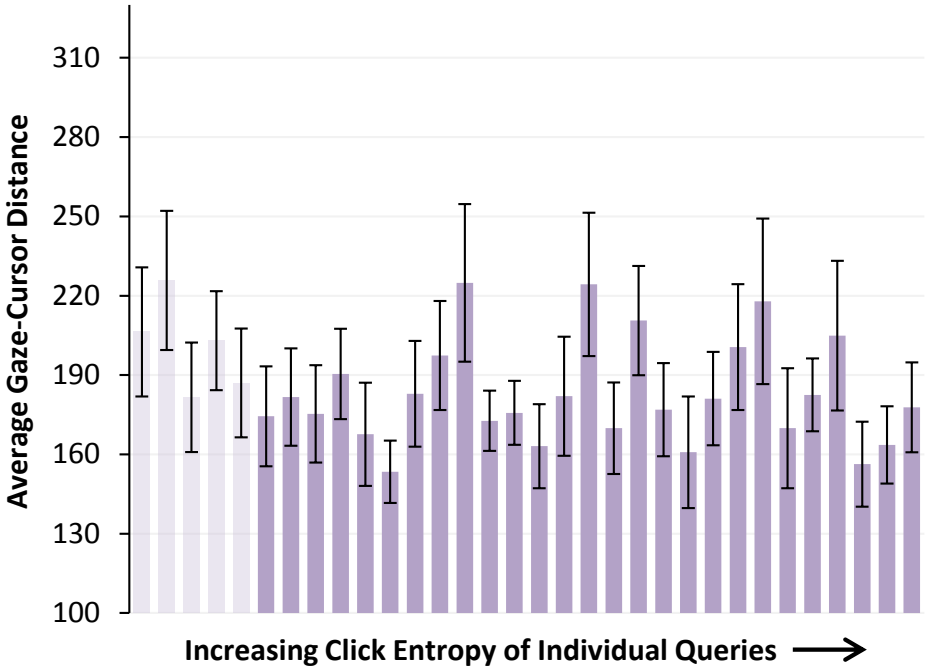

Figure 4.3: The average gaze-cursor distance for each search task with error bars representing the standard error. The variance between search tasks is modest ($SD = 20.2$). Queries are sorted by ascending click entropy so queries with more diverse result clicks are on the right; five queries with unknown entropy are on the left (shaded lighter).

$= 4.529, p = 0.037$). This suggests that users have individual preferences and that these differences are stronger than the differences between search tasks. I experimented with not normalizing the data when computing the mean averages (i.e., micro-averaging), but the differences in standard deviation between search task and subject were similar.

These user and task differences also surface in interaction data collected remotely, i.e. the second dataset sample described in the Methods chapter. In work that I was involved in, Buscher et al. [16] show that there are cohorts of users who examine search results in a similar way, and that the grouping becomes clearer when we consider task effects. The work found pronounced task effects that impact how users engage with the SERP and that can interact with users' typical search behaviors. These findings were established by analyzing logs containing detailed data on user interactions including clicks, scrolls, and cursor movements for millions of search queries and presenting a study of individual

differences in search result page examination behavior. By clustering the data using these interaction features, we identified individual differences in search behavior, and strong effects of user, task and user-task interaction.

Our initial analysis revealed six user clusters. However, we also showed that there are strong effects from the type of search task on users' search behavior, as well as strong interaction effects between task and user. When we focused on non-navigational tasks, we found three distinct user clusters who exhibited different result examination behaviors. Promisingly, users exhibited behavioral patterns similar to those found in previous gaze tracking research [5, 30], especially the presence of exhaustive and economic groups (as mentioned in Section 4.3). Not only do we confirm the existence of these clusters in a naturalistic search setting, but also demonstrate that we can automatically generate them via search engine log analysis. Identifying users with consistent search strategies and patterns is important to understanding how systems are currently being used and create search support.

When we consider task type (by focusing on non-navigational queries) in conjunction with user behavior, three distinct user clusters emerge. These clusters share behavioral traits with those identified in laboratory studies, but we observe these without gaze tracking technology and at scale on the Web, opening up a wealth of opportunity for adaptation of the search experience based on individuals' searching behaviors.

Tailoring the search experience to a user and task are important, but there may be limited practical application to an adaptive model. The reason is that many users and tasks have not been seen by the search engine before, so a personalized model is only helps in a subset of cases, when there is enough historical data from that user or task to make inferences. Global models like the ones presented in the next two chapters help in all cases, and are not restricted to having seen a particular user or task before.

4.4 Temporal Effects on Alignment

In this section, I investigate the possibility that the gaze and cursor may not be most aligned at each point in time, but rather better aligned when one variable is temporally shifted. Researchers have found that there is some visual sequence in how users examine a Web page, which can be predictable [12]. The duration that the search results page has loaded

can affect where users are pointing and where they are looking at each point in time. For example, a user may possess the habit of quickly scanning the page first to see what kinds of items are on the page, then skimming it quickly to see if there is an answer to their information need while neglecting the cursor, then finally reading the text word-by-word. As the time spent dwelling on the SERP increases, the alignment between gaze and cursor may change due to the dynamism of the behaviors.

This change may be analyzed from the eye-tracking data. Averaging gaze and cursor distances for different dwell times, Figure 4.4 shows the relationship between gaze and cursor over time for the first five seconds after the page loads. Indeed, the time since the page has loaded has an effect on alignment. Specifically, the alignment distance ranges from 170 px to almost 240 px. Right when the page loads, gaze and cursor are closely aligned, perhaps from the previous action that led to the page. The peak at 240 px is within one second of the page loading, suggesting that the subject may first scan the displayed page without moving their cursor. The gaze-cursor alignment narrows after about two seconds when the subject may start to examine the page more closely and perhaps prepare to click a link. While the subject's actions at this stage change from query-to-query, the aggregate alignments provide clues of typical examination behavior.

This behavior led us to ask—given that we see the eye moving within a second of the page loading, does the cursor move as quickly? I arrived at three credible hypotheses: a) the user treats the cursor as a reading aid [84] and so the eye follows the cursor, b) the gaze and cursor positions are optimally aligned at each given point in time, c) or the cursor follows gaze because the user looks at something and then moves their cursor to interact with it. The last hypothesis is consistent with findings from early work by Ware and Mikaelian [95] and Zhai et al. [102] that showed that the eye fixation was a faster method for target selection than the cursor. While target selection is not the objective of this study, the difference between eye and cursor speed may cause the cursor to lag behind the gaze.

To investigate this, I used a technique similar to cross-correlation in signal processing that computes when two waveforms are most aligned [53]. I first interpolated cursor and gaze positions at 50 ms intervals again using Equation 3.1. This equalized the periods between positional data points, computing the cursor positions at different time shifts (e.g.,

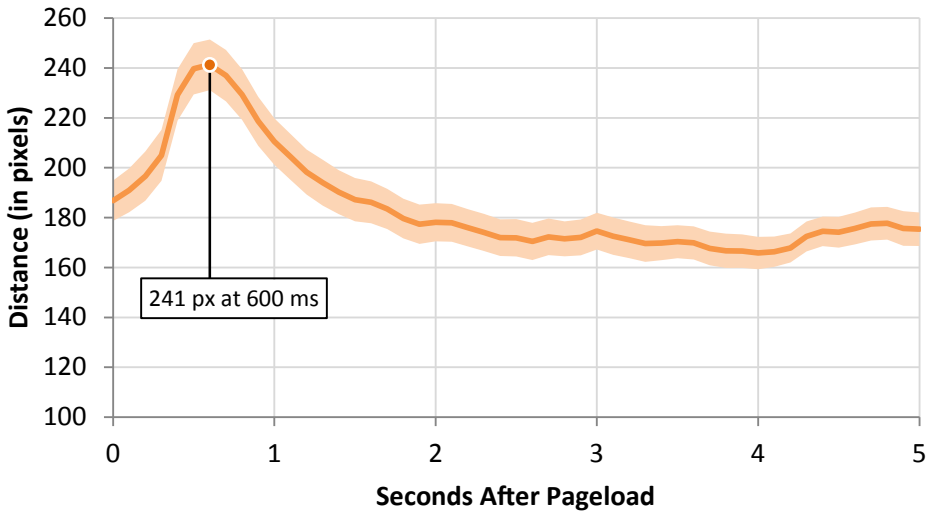

Figure 4.4: The average gaze-cursor distance at 100 ms intervals after the SERP loads, macro-averaged over subjects. The distance is low just as the page loads, increases at 0.5–1 second, then decreases. The shaded area is the region representing the standard error of the mean.

50 ms, 100 ms, 150 ms, ...) after each gaze position. Then I computed the root mean square error (RMSE) between gaze and cursor[4] for each shifted time interval for each subject. The cursor and gaze positions are considered most correlated at the time shift with the lowest RMSE.

Figure 4.5 shows the macro-averaged RMSE values across subjects compared with three example subjects. The first thing to note is that for the macro-averaged values, shifting the cursor positions 700 ms into the future minimizes the RMSE. This means the cursor lags behind eye gaze by about 700 ms, so the user looks at something and then (almost one second) later their cursor is moved to that location on the page. Temporal alignments vary depending on the user and query session; the per-subject RMSE values showed that different people had different delays in moving their cursor to their gaze. In the same manner, the cursor lagged behind the gaze for each individual subject; the inverse situation—gaze lagging behind the cursor—did not occur, refuting the hypothesis that some people lead with the

[4]Note that RMSE differs from the average alignment distance metric used throughout this chapter; RMSE takes the root of the mean of the squared distances which is not the mean distance. RMSE is useful in this instance for comparing between different values of time shift, and penalizing larger distances.

cursor when examining the page. Still, some subjects were quick, moving their cursor to their gaze in 350 ms (one subject had a minimum RMSE at 250 ms shift), while others took over one second.

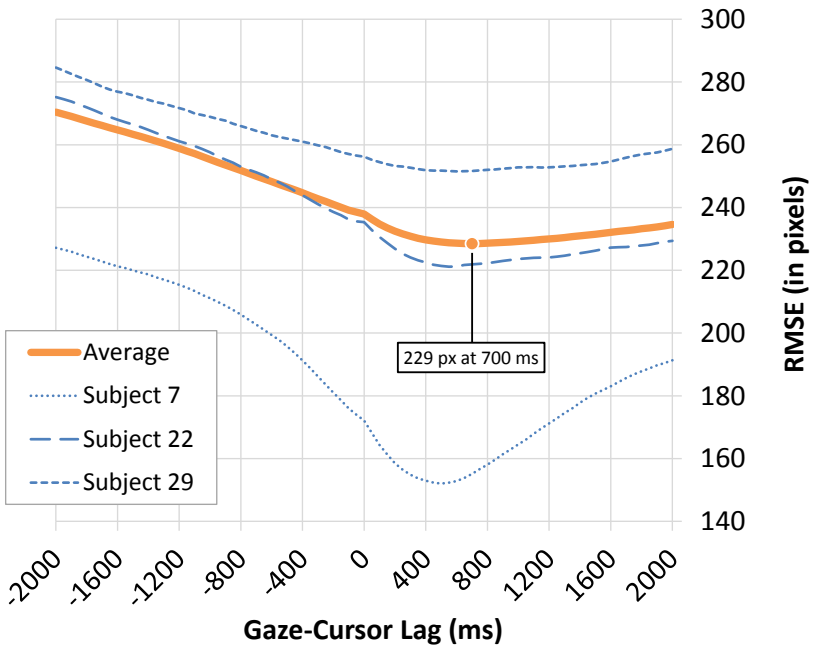

Figure 4.5: The root-mean-square error for gaze and cursor distance at different intervals (in 50 ms increments) of gaze-cursor lag, representing how well gaze positions correlate with future and past cursor positions. The thick solid line plots the RMSE macro-averaged over subjects, and thin dashed lines plot the RMSE for three example subjects.

So far, the findings show that while subjects vary their gaze-cursor alignment substantially, they lag their cursor behind their gaze by at least 250 ms and on average by 700 ms. Both dwell time and the user's personal style affect the distance between the cursor and gaze positions. Finally, as shown earlier in Table 4.1, this distance is longer when the cursor is inactive, shorter when the cursor is used to help examine or read the page, and even shorter when the user is performing an action.

Chapter 5

PREDICTING GAZE WITH CURSOR INTERACTIONS[1]

While studying how eye gaze aligns with the cursor position is useful to understanding behaviors (Section 4.1), knowing only the alignment has but limited application. A more practical application is being able to actually predict the eye-gaze position based solely on cursor interactions, so that visual attention can be determined at any point in time, and at scale. By reducing the barrier of time and effort in acquiring knowledge about visual attention on Web pages, I can enable a better understanding of searcher behavior.

5.1 Motivation

This chapter illustrates that cursor interactions can be used to predict eye-gaze better than simply using the current cursor coordinates, thus estimating user attention more accurately. In other words, rather than simply determining how likely the gaze and cursor are aligned, I propose a technique to select different points on the screen that are more likely to be where the user is looking than the current cursor position. The technique uses features from the cursor interactions and uses the lessons from the previous chapter to build a model for predicting where people were looking on a page, all while the user browses naturally from home.

The goals of this chapter can be illustrated with an analogy. *Imagine you are a spy plane, monitoring a ship sailing in the ocean. The ship sails around, moving fast or slow, sometimes stopping, sometimes for a purpose and other times for seemingly no reason at all. One day, you are lent x-ray goggles and see a submarine which is normally hidden, protecting the ship. The submarine tends to follow behind the ship, but varies greatly in how well its position aligns with the ship. There are also patterns like when the ship is stopped, the submarine often goes elsewhere. You learn these patterns, and after you return*

[1]Portions of this chapter are published in CHI 2012 [49]

the x-ray goggles, you still have a reasonable idea where the submarine may be based on the ship's movements. In our case, the ship is the cursor, and the submarine is the eye-gaze. The data from the eye-tracking study reveals how the gaze and cursor are aligned, but the eye-tracking data can be used to predict gaze even when an eye-tracker is not available by using the model built from the study's data.

My first attempt at this is to use a multiple linear regression which allows point prediction. A later section will discuss potential next steps for predicting gaze using a hidden-markov model, which discretizes features to do region prediction. Predicting the point where a user is looking is similar to eye-tracking, while region prediction requires knowledge about the layout of the page in order to be useful.

The findings from this work can be applied to many practical situations. Predicting gaze will improve state-of-the-art techniques for approximating visual attention with the cursor. Current commercial cursor-tracking systems merely present heatmaps of the cursor movements and claim that the user attention is proximal to the hot areas of the map. My findings will inform these and other applications that remotely collect large amounts of cursor data at scale by showing how to more effectively use this data. More broadly, this work supports collecting user attention data in situations where eye-tracking equipment is unavailable or costly, such as for large numbers of Web users.

5.2 Cursor Features

To acquire the features, I extract four types of features from the raw interaction data, aiming to select features that seemed to influence the gaze-cursor alignment, as informed by the analysis in the previous chapter.

At each time t that we want to predict the gaze position, we have the **cursor position** at that time, represented by a tuple (x, y). The cursor position alone is suggested for approximating gaze position in some prior literature and current Web analytics services. This approach is the baseline against which I will compare our performance, but I also use cursor position as a feature in our gaze prediction model. In addition to the cursor position,

the model uses dwell time[2], temporal features relating to cursor behavior, and future cursor positions to determine the current gaze position.

Cursor behavior has a strong effect on gaze-cursor alignment, as shown in Section 4.2.2. Both our earlier analysis and a study by Hauger et al. [44] showed that active cursors are better aligned with gaze than inactive cursors. A long idle time is an inactive cursor with larger distance between gaze and cursor, while little or no idle time between cursor movements indicates an active cursor. To generalize this, I use the idle time following the last movement before t as the **behavior** feature, representing activity level. Idle time is an easy value to compute and provides a signal of the potential strength of alignment into the model.

Each recorded interaction on the Web page has a corresponding timestamp, allowing us to deduce the length of time since the SERP has loaded. Our analysis has shown that the time since the SERP loaded influences the gaze-cursor alignment. This time ($\text{current}_t - \text{pageload}_t$) is the **dwell** feature, which I incorporate into the linear regression model. Guo and Agichtein also use this feature to predict gaze-cursor alignment [41]. This feature has the potential to influence both x-axis and y-axis predictions because of the visual sequence in which users examine Web pages.

Our analysis showed that for every subject in our study, the cursor position lagged behind the gaze position, i.e. there is a stronger correlation between a future cursor position and the current gaze position. I refer to the globally most likely later cursor position for the current gaze as the **future** feature. The future cursor positions were only used if the last movement was within 10 seconds of the target future time. However, including future cursor positions in the model prevents the ability to predict gaze in real-time (see the Discussion section for detail).

The model also incorporated the interaction effect of cursor position with dwell and cursor position with behavior, since we know that behavior and dwell affect alignment. The user or query were not treated as features for two reasons. First, in practical situations, a user's gaze data is not available to train the model. Second, there is unlikely to be enough

[2]Dwell time here is the page dwell time as typically measured in information retrieval literature, rather than cursor dwell time.

gaze data for most queries to train the model, and the analysis described in an earlier chapter found that query only has a modest effect on gaze-cursor alignment. Our current features are all global and so require fewer training examples in the form of eye-tracking data (i.e., actual gaze positions).

5.3 Experiment

I predict the subject's gaze position using a linear model of interaction features to see if the predicted position is closer to the ground truth than the simple cursor position. The ground truth is the gaze position measured by the eye-tracking system during the lab study. The x- and y-coordinates of the gaze position were predicted separately (i.e., univariate prediction). To compute the weights (coefficients) for each feature, I performed a multiple linear regression. Figure 5.1 illustrates the value of gaze prediction in an example query session with cursor positions, gaze positions, and predicted gaze positions overlaid on the SERP.

The model for the regression for the x-coordinate is:

$$g_x \sim c_x + \log(t_d) + \log(t_m) + c_x \times \log(t_d) + c_x \times \log(t_m) + f_x \tag{5.1}$$

where t_d is the dwell time, t_m is the time since a movement, c_x is the x-coordinate of the cursor position, g_x is the x-coordinate of the gaze position, and f_x is the most likely x-coordinate of the gaze based on future cursor positions. The regression equation for the y-coordinate was similar, but substituting x for y in Equation 5.1.

The evaluation was a 36-fold cross-validation, where each fold was an individual subject; this is essentially a leave-one-out evaluation for each subject. By testing each subject separately with the training data of all 35 other subjects, we achieve a practical method of predicting the gaze position for users that have not been seen before (which would be the typical case in a real deployment unless there was a per-user training phase). Essentially, a multiple linear regression on the gaze and interaction data of 35 "training" subjects were used to compute the weights (coefficients); then those coefficients were used to predict gaze positions from just the interaction data for the "test" subject. This process was repeated for every subject being the "test" subject. For each subject tested, I computed the RMSE

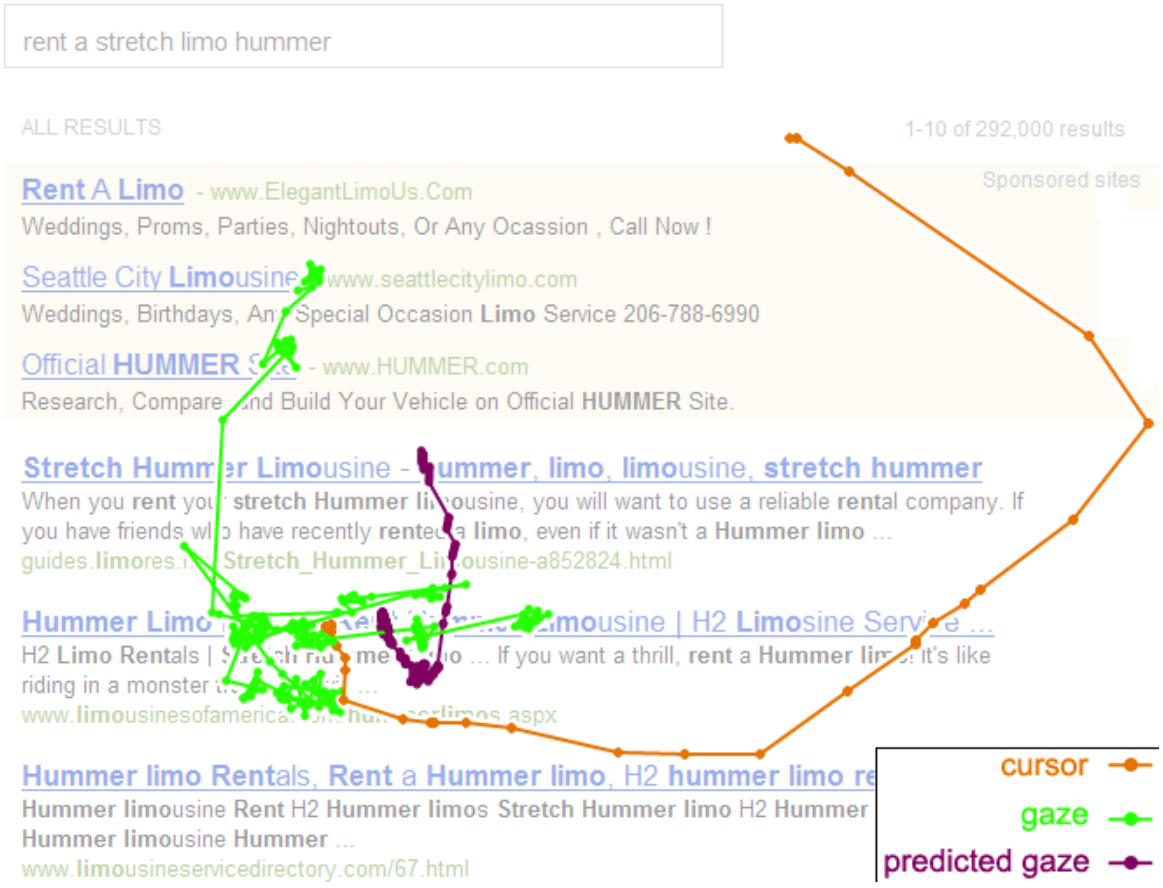

Figure 5.1: The cursor position (orange), the gaze position predicted by the linear regression model (purple), and the gaze position as determined by the eye-tracker (green) are drawn over the SERP presented to the subject following the query "rent a stretch limo hummer" from our study. The figure omits the right and left columns of the SERP.

for their predicted gaze with and without the future feature, as well as the RMSE for just the cursor positions. Without using the future feature, there is a potential application for real-time gaze prediction, while with future cursor positions, the gaze prediction must be performed offline (after the query session, e.g., for analytics).

Table 5.1 shows the results of the prediction experiment. When using only the cursor position for prediction, the distance RMSE is 236.6 px. But using a multiple linear regression with cursor position, behavior, and dwell time, the predicted gaze position is significantly closer to the actual gaze position—186.3 px, a 21.3% decrease in RMSE. Adding future

	RMSE_x	RMSE_y	RMSE_d
Cursor (baseline)	185.0 px	145.0 px	236.6 px
Predicted Gaze with Cursor+Behavior+Dwell	125.2 px	137.1 px	186.3 px
Predicted Gaze with Cursor+Behavior+Dwell+Future	125.1 px	129.9 px	181.1 px

Table 5.1: The computed accuracies in cross-validation evaluations of estimating the gaze position using the cursor position, the cursor position along with behavior and duration using multiple linear regression, and the cursor position along with behavior, duration, and future cursor positions using multiple linear regression. The accuracy is measured by root-mean-square error for the x-axis, y-axis, and Euclidean distance.

cursor data from that query session to the model reduces RMSE further to an overall 23.5% decrease in RMSE compared to just using the cursor position. The RMSEs in the x- and y-coordinates alone were similarly improved by the linear model. An ANOVA shows that the RMSE differed significantly between gaze-cursor alignment and alignment between gaze and predicted gaze along the x-axis ($F(2, 105) = 59.72, p < 0.001$) and Euclidean distance ($F(2, 105) = 41.31, p < 0.001$).

The Lindeman, Merenda and Gold (LMG) metric [69] (R^2 partitioned by averaging over orders) determined the relative importance of the features for predicting the x-coordinate. This metric differs from regression weights which depend on the unit of measurement for the feature. Using LMG, the features were in descending order of importance: $\log(t_d)$, c_x, f_x, $c_x \times \log(t_m)$, $\log(t_m)$, and $c_x \times \log(t_d)$; the relative importance of the features for predicting the y-coordinate in descending order: c_y, f_y, $\log(t_d)$, $\log(t_m)$, $c_y \times \log(t_m)$, and $c_y \times \log(t_d)$.

5.4 Discussion

This analyses in Chapter 4 has shown that gaze-cursor alignment is situational, as it depends on the time spent on the page (Section 4.4), personal browsing habits (Section 4.3), and a user's current cursor behavior (inactive, examining, reading, action) (Section 4.2.2). A model using these features could predict the subject's gaze significantly better than using the cursor position alone. These findings have implications for using large-scale cursor data

more effectively, which has already been demonstrated to be efficiently obtainable at scale, both in prior work [4] and in this dissertation.

The findings suggest that for certain circumstances, it may be possible to predict the actual gaze position on SERPs using only cursor features. This extends previous work [41] which attempted the binary prediction task of whether we can be confident of gaze-cursor alignment within a threshold. Using a linear model, I show that there is room for improving gaze prediction by using other factors on top of solely the cursor position. The predictive model reduces the RMSE by 60 px in the x-direction and is 15 px more accurate in the y-direction. 15 px is around two lines of text on a SERP, and 60 px is around 10 characters of that text. These gains could be significant for differentiating between engagement with regions on the SERP, especially when the cursor is near the boundary of two or more regions. However, since the linear regression fits using least squares, the predicted gaze tends to be conservative and often stays around the center of the screen. This leads us to believe that gaze prediction may be improved further by a more complex combination of the features, perhaps in non-linear models, since interaction effects of time, behavior, and cursor position exist.

The gaze prediction results in Table 5.1 suggest a counterintuitive finding. While several past studies [41, 83] and our own analysis from Section 4.1 agree that the cursor is better correlated with the gaze in the y-direction than the x-direction, the x-coordinate of the gaze is easier to predict than the y-coordinate. This undermines the expectation that better correlation leads to better prediction for gaze. This phenomenon may be an artifact of the prediction model, but perhaps left-right eye movement is less surprising than up-down movement. Scrolling may cause the cursor to move down relative to the page, making it a good estimate of vertical attention, whereas most SERPs require little horizontal scrolling.

The results show that one can predict gaze more accurately when using past and future cursor movement data, rather than only past data. However, the gain in accuracy is quite small, whereas using past data alone would allow us to build applications that could respond to user attention in real-time, such as a focus-plus-context view of the SERP, in which the context could update dynamically with new content based on where the user had already attended during their engagement with the SERP. Still valuable however, is inferring gaze

positions after the user has left the SERP. This would allow us to accurately model where on the SERP the user examined and use this data for applications such as building richer searcher models (improving on existing models, e.g., [18, 37]), usability assessments [3, 4], or profiling users [4]. Thus, whether or not to use the future cursor positions in practice may depend on the application, since the accuracy gain is minimal.

Cursor tracking, deemed the "poor man's eye tracker" [22], may approximate gaze tracking without the eye tracker depending on the accuracy required. Although cursor features would allow us to model many aspects of user attention *in situ* as they browse the Web from home, they cannot completely replace gaze. For example, eye-gaze fixation is a positive signal of interest because the user pays more attention to that position, but prolonged cursor fixation may not be since given this study's findings, the user's attention is probably elsewhere. Here I elected to focus on understanding different cursor behaviors rather than gaze fixations to support cursor-tracking applications that can be remotely deployed on a Web site. More work is needed to study in more detail the relationship between cursor and gaze fixations, especially to determine if and when there are cases in which cursor fixations can be reliably interpreted as attention.

5.5 Further Extensions: Non-Linear Methods

The relationship between gaze and cursor is not necessarily linear. In fact, I observed many instances where there was a non-linear relationship between the two trails. For example, the inactivity of the cursor affects the vertical movement of the eye on the page because the user is likely scanning up and down the page after moving their cursor aside. However, making non-linear predictions is inherently more challenging because it is unclear how exactly to define the relationship. The space of possible relationships between the features is intractable, but models may be constructed using an understanding of how the cursor features affect eye gaze. This section describes two potential methods for predicting gaze using non-linear models, one developed by myself and the other [74] by Navalpakkam et al. They represent potential future directions to pursue if getting the best predictions of gaze positions is the ultimate goal.

5.5.1 Using Hidden Markov Models

Because of the sequential nature of the eye-gaze movements, the movements map well to a hidden-markov model (HMM). In the HMM, the gaze positions can be treated as the hidden states with transitions between different positions. The cursor interactions are the observed features that are output depending on the gaze positions and the intent of the user (Figure 5.2). Using the eye-tracking data that includes both gaze and cursor features to train the HMM, it is possible to estimate where the eye is looking.

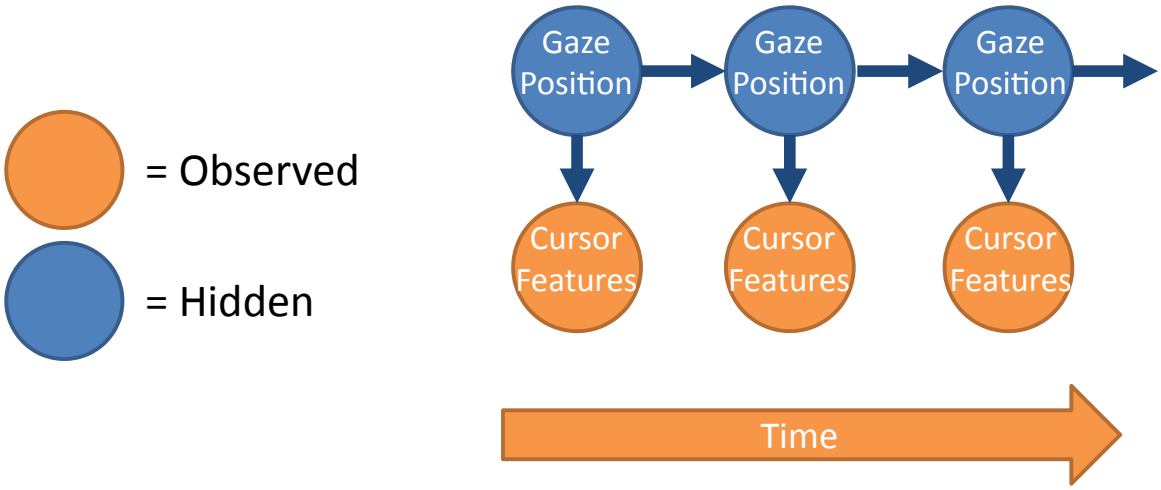

Figure 5.2: An example hidden markov model for predicting gaze positions.

Because the HMM is a discrete model, the features and gaze positions have to also be discretized. Rather than prediction the actual position of the eye-gaze, it is useful to instead predict the gaze position relative to the cursor position. That is, using polar coordinates of the gaze relative to the cursor. Instead of predicting the x and y coordinates, predicting the angle and distance between gaze and cursor may make sense in an HMM, as variables that are more meaningful to a user. The angle and distance can be discretize to form regions emanating from the cursor's position, with larger regions existing further from the cursor. However, further research is needed to explore empirically how well an HMM can predict the region a user is looking in.

5.5.2 Extensions of Gaze Prediction Models

Navalpakkam et al. extend my methods of gaze prediction in two ways [74]. One is by exploring gaze prediction on 2-column page layouts, which is less typical in search but may represent cases when information or advertisements are shown in the second column. Secondly, they make gaze predictions using a non-linear model of cursor interactions through a kernel method. They also conduct experiments to show that using the historical behavior from a particular user can improve the gaze prediction for that user as well. Similar to the gaze prediction methods presented in this chapter, their methods perform point prediction as well as region prediction, and take prediction accuracy one step further.

5.6 Summary

In the previous chapter, we have seen the effects of user, query, dwell time, cursor behavior, and future cursor positions on gaze-cursor alignment. These features can provide guidance about whether the alignment is stronger or weaker, but predicting the strength of alignment is only useful in determining the confidence of the gaze position from only interaction features. Claiming that the cursor approximates the gaze is misguided, as I showed before, because this assumption is often not true depending on time and behavior. Instead, a more practical prediction task is to find the position of the user's attention when an eye-tracker is unavailable by using interaction features.

This chapter shows that it is possible to improve upon using cursor position alone to predict the gaze position by using several cursor features. Note that while the evaluation uses data from an eye-tracking lab study, a pre-trained model can predict gaze at Web scale in any situation where cursor interactions are collected. Cursor movements, scrolling, and other client-side interactions are easy to collect at scale, which many Web analytics services offer to do. The linear regression technique predicts the gaze position using cursor features to achieve 23.5% more accuracy than simply using the corresponding cursor position alone. Non-linear methods are also possible to reach higher levels of prediction accuracy. Chapter 7 focuses on a more focused task of computing attractiveness and relevance of search results specifically.

Chapter 6

PATTERNS OF CURSOR MOVEMENTS DURING SEARCH[1]

A first step to making use of cursor interactions is to explore the nature of the data to find out how people are actually using their cursors when they search, specifically how their move their cursor when they search. One aspect to this exploration is learning what results they hover over, and what this may mean for result relevance and examination. This chapter reports analysis of the first data sample gathered through remote cursor tracking (see Method from Section 3.2).

6.1 Hover on Search Results

6.1.1 General Cursor Activity

Aggregate analysis of general cursor activity can provide an overview of how users actually use their cursor when examining results. This subsection determines where on the SERP users click and move their cursors. The coordinates of the recorded clicks and movements offer some initial insight into differences between click and movement data.

The positions of where a user has clicked or moved their cursor can be overlaid on the SERP, and a simple heatmap illustrating these cursor positions can be beneficial for website designers or usability professionals. Figure 6.1 shows heatmaps for clicks and cursor movement activity for the same query aggregated over all instances of the query [lost finale explanation] (in reference to the final episode of the US television series "Lost") observed 25 times from 22 different users in our data. Heavy interaction occurs in red/orange/yellow areas, moderate interaction in green areas, and light interaction in blue areas. Most of the clicks occur on results 1, 3 and 7, and this is also seen in the cursor activity. However, there are some interesting differences as well. For example, there is considerable cursor activity on result 4 even though it is not clicked. The cursor heatmap also shows some activity on

[1]Portions of this chapter are published in CHI 2011 [51]

Figure 6.1: Heatmaps of all click positions (left) and recorded cursor positions (right) for the query *lost finale explanation*. Heavy interaction occurs in red and yellow areas, moderate interaction in green areas, light interaction in blue areas.

query suggestions (on the left rail) and advertisements (on the right rail) although there are no clicks on these regions. Across all queries, cursor positions are more broadly distributed over the SERP than clicks. Thus, cursor movement can provide a more complete picture of interactions with elements on the SERP. Such information may be useful to search engine designers in making decisions about what content or features to show on search result pages.

6.1.2 Search Result Examination

In addition to monitoring general cursor movement activity on the SERP, we can also summarize cursor movements that reflect how people examine the search results. Previous work on gaze tracking demonstrated differences in the length of time that users spend reading each of the results based on its position in the ranked list [25]. In a similar way, it is interesting to determine whether the time participants spent hovering over the search results was related to the position in the ranked list. Using the data about the location of the SERP elements in conjunction with the cursor positions, I could determine when each search result title was hovered over by the cursor. Figure 6.2 presents a graph of the average

time spent hovering over each search result title (shown as bars; corresponding scale shown on the left side), and the average time taken to reach each result title in the ranked list (shown as circles connected with lines; corresponding scale on the right side). To reduce noise caused by unintentional hovering, I removed hovers of less than 100 ms in duration. Error bars denote the standard error of the mean (SEM).

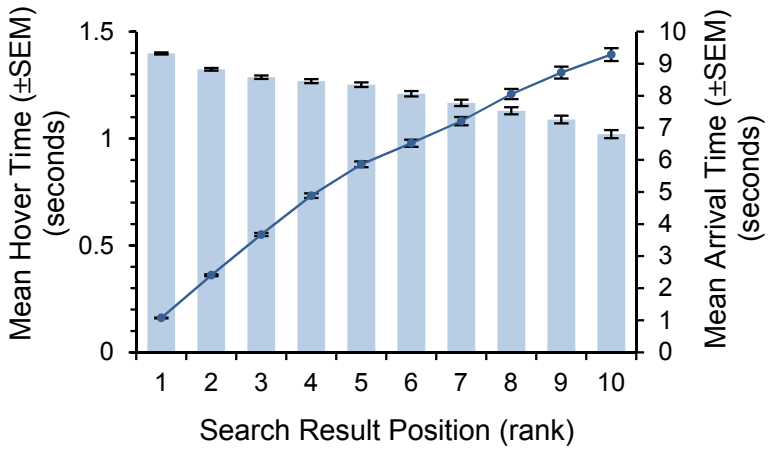

Figure 6.2: The mean title hover duration (bars) and the mean time for the user's cursor to arrive at each result (circles).

The figure shows that time spent hovering on the results decreases linearly with rank and that the arrival time increases linearly with rank. The results are similar to gaze tracking findings reported in previous literature [14, 25, 57]. Hover time decreases with rank as was previously reported; however, cursor hover time drops off less sharply than gaze duration. This difference may be due to some missing data of rapid skimming behavior on low ranks that has been observed previously [14, 25, 57], since the data only comprised hovers after a 40 ms pause (to reduce data payload) and filtered out hovers of 100 ms or less (to reduce cases of capturing accidental hovers). As expected, search results that are lower ranked are entered later than higher ranked results due to the typical top-to-bottom scanning behavior [23]. The arrival time is approximately linear, suggesting that users examine each search result for a similar amount of time.

I also examined which results were hovered over before clicking on a result, re-querying,

or clicking query suggestions or advertisements. This analysis provides further information about how searchers are using their cursor during result examination and again allows us to compare the findings with prior eye-tracking research from Cutrell and Guan [25]. Figure 6.3 summarizes the findings, showing the mean number of search results hovered over before a click as blue lines, and clicks as red circles. The data are broken down by result position (1–10), and separately for clicks on query suggestions, clicks on ads, and re-queries.

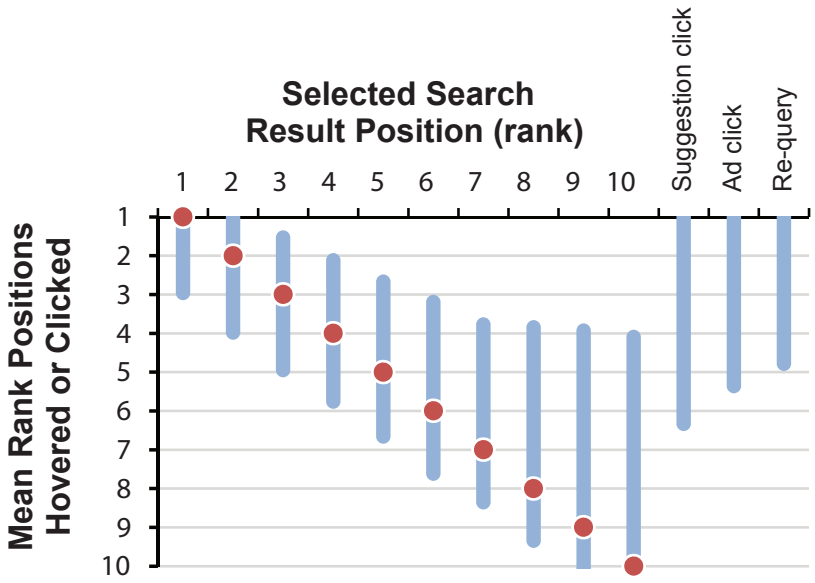

Figure 6.3: The mean number of search results that users hovered over before clicking on a result (above and below that result). Result clicks are red circles, while result hovers are blue lines.

Figure 6.3 shows that prior to clicking on a search result, people consider the surrounding search results. For example, before clicking on result 1, searchers also hover on results 2 and 3 on average; when they click on result 2 they also hover on results 1, 3, and 4; etc. The findings are similar to those reported by Cutrell and Guan [25], but differ in that the search result hovers do not appear to extend as far above the clicked search result in cases where a result is clicked on far down the list (in positions 6–10). This difference may be because queries where low-ranked clicks are observed may have clearly irrelevant results in top ranks, and by excluding hovers of less than 100 ms we miss rapid skims over such

irrelevant results.

The findings also show that users consider many results prior to turning their attention to the additional SERP features: on average, six results in the case of query suggestions, five results in the case of advertisements, and around four results prior to re-querying. This behavior is similar to that reported in Cutrell and Guan [25], at least in terms of re-querying, which is examined in both studies. Cutrell and Guan do report inspection further down the list (up to rank position 8) prior to re-querying, whereas the findings here show that users hover over approximately four results. One explanation for the difference is that the cursor does not track well with eye movements in situations where users rapidly skim low-ranked search results. An alternative explanation is that in naturalistic non-laboratory settings, users may only consider the top-ranked search results prior to trying another query by clicking on a query suggestion or re-querying.

The next section compares the distributions of search result clicks and search result hovers, expanding on this analysis of search result behavior.

6.1.3 Comparing Click and Hover Distributions

Prior studies have presented data on click distribution [57, 77] or gaze distribution for the search results [14, 57]. These distributions reveal on average how much attention a user gives to each result because of its rank and other features such as snippet content [25]. Some theoretical models of behavior depend on accurate assumptions of these distributions, e.g., Huang and Kazeykina [48] assume the frequency with which users review a search result is a power law of its rank, while Wang et al. [94] assume the frequency with which a search result is clicked follows a geometric distribution of its rank.

Click and hover distributions can be analyzed using the cursor positions and layout of the SERP collected from the remote cursor tracking dataset. This analysis presents a cursor hover distribution over search results and compares it with the corresponding click distribution. Figure 6.4 shows both the number and proportion of cursor hovers and clicks that occur on each of the top ten search result links. Bars representing absolute counts are scaled by the primary y-axis (on the left), e.g., there are approximately 240,000 occurrences

of the cursor hovering on the first search result. Circles representing percentages are scaled by the secondary y-axis (on the right), e.g., 50% of result clicks occur on the first search result.

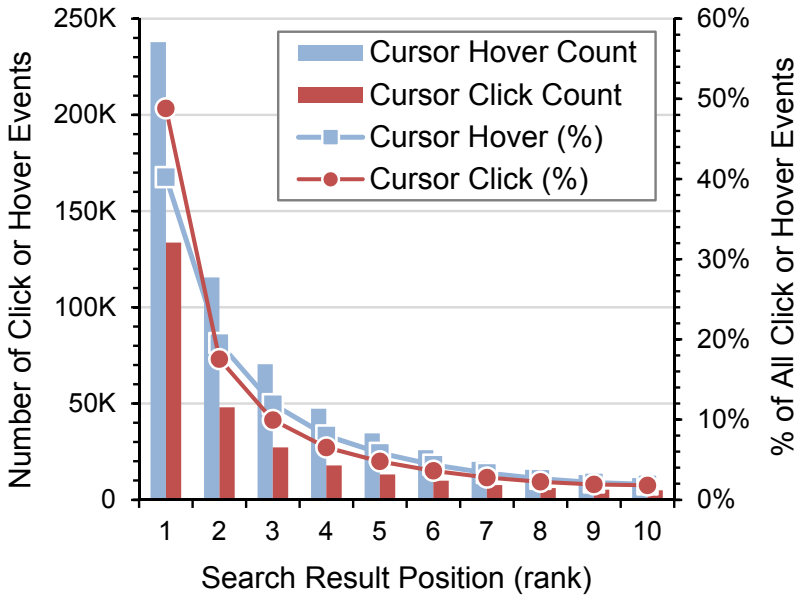

Figure 6.4: Frequencies and percentages of cursor hovers and clicks occurring on the search results. The percentages reflect the proportion of hover or click events over all ten results.

As is typical in SERP interactions, users interact more with top-ranked search results because they are in a more accessible location and are presumed to be more relevant. However, Buscher et al. reported that the distribution of clicks does not always reflect the relative distribution of visual attention (measured by gaze in their study) [14]. Similarly, we find that hovers are more uniformly distributed across the top-ten results than clicks, and the hover rate is higher than clickthrough rate for all ranks beyond the first position.

6.1.4 Unclicked Hovers

In cases where there is no click, cursor interactions are particularly important as they become the only interactions that can be captured. This subsection investigates whether hovering over a result but not clicking on it can be a useful signal of user interest. To

examine this, I define an unclicked hover as an instance of the cursor hovering over a link but not clicking that link before being moved to another location on the page. Table 6.1 shows the number of unclicked hovers on a search result and the percentage of times that it was subsequently clicked by the same user. Result clicks can occur without an unclicked hover in two scenarios: when the user does not hover over the result for at least 100 ms, or if they don't move their cursor to another location on the page before clicking the result.

# unclicked hovers	Results clicked
0	7.0%
1	16.7%
2	19.0%
3	22.4%
4	23.3%
5	25.2%

Table 6.1: The percentage of unclicked hovers for which the hovered search result was eventually clicked.

When there are no unclicked hovers, the result is not likely to be clicked (only 7% of the time). Observing one or more unclicked hovers dramatically increases the likelihood of a result click, perhaps because it demonstrates that the user has attended to it. The trend appears to be that the more occurrences of unclicked hovers, the more likely the user will ultimately return to the result and click it. The Pearson correlation between the number of unclicked hovers and the percentage eventually clicked is strong ($r = 0.84$), when considering up to 10 unclicked hovers. Thus the number of unclicked hovers on a result may help predict result clickthrough or perhaps result relevance.

Segmenting the unclicked hovers by the search result rank shows that result rank significantly affects unclicked hover behavior. Figure 6.5 shows the proportion of each result rank that is eventually clicked after an unclicked hover.

The blue squares show that a search result is eventually clicked after an unclicked hover around 25% of the time for the top-ranked result and less than 15% for low-ranked results. However, when we consider that low ranked results typically have a low clickthrough rate,

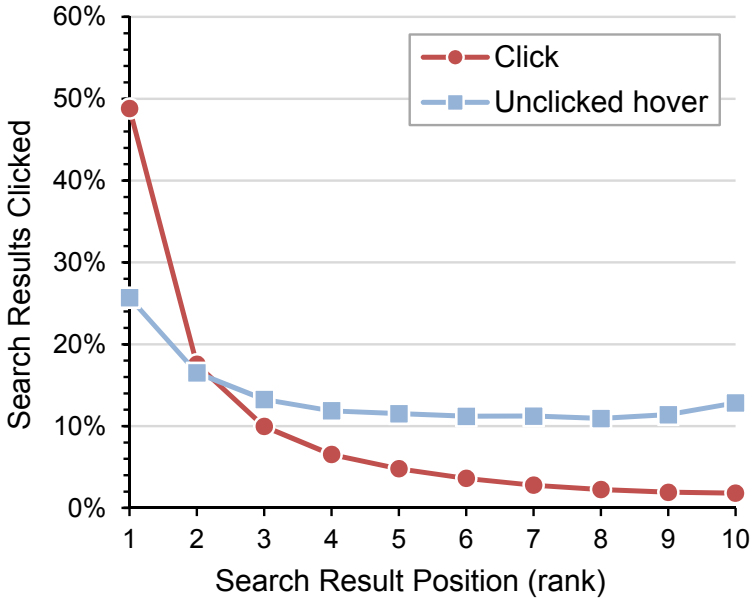

Figure 6.5: The proportion of search results that are eventually clicked after an unclicked hover, plotted against the click distribution from Table 6.1.

an unclicked hover on a low ranked result may actually provide implicit feedback that the result is relevant. To illustrate this, I overlay the click distribution on the chart to compare the probability that an unclicked hover results in a later click (blue squares) with the original probability that the search result will be clicked (red circles). The chart shows that whether an unclicked hover is a positive or negative indicator depends on result rank. To quantify the degree of this effect I compute the phi coefficient (ϕ) across all queries[2]. For the first search result, the presence of unclicked hovers negatively correlates with result clicks ($\phi = -0.47$), but for results at lower ranks, unclicked hovers positively correlate with clicks ($\phi = 0.59$). Thus, this is evidence that an unclicked hover can have different meanings of result attractiveness depending on the rank of the result.

[2]The phi coefficient measures the association between two binary variables, a specific case of the Pearson correlation coefficient which measures the association between two continuous variables.

6.2 Estimating Search Result Relevance

There are a broad range of possible signals that can be extracted from large volumes of cursor tracking data, from signals about the query type to indicators for potential search interface enhancements. The next two sections demonstrate that capturing cursor activity could give insights about two elusive metrics in search: relevance and abandonment. The results showed that a user hovering their cursor over a search result was a better indicator of its relevance than a user clicking, and their cursor movement speed and distance related to whether they would abandon the search.

One useful signal that we attempt to extract from cursor movement data is whether the search result is relevant to the user or not. At scale, these data could be used as an additional data source to train search engine ranking algorithms and boost retrieval performance. As part of this study, we gathered human relevance judgments for query-URL pairs, and examined the correlation between features of the cursor movements and the human relevance judgments. In addition, we examined the signaling value that cursor movements provide compared with search result clicks, the more traditional source of behavioral data used to estimate search result relevance.

As a first step, the human relevance judgments were obtained for thousands of queries as part of an ongoing evaluation of search engine quality. Trained judges assigned relevance labels on a five-point scale—*Bad, Fair, Good, Excellent, and Perfect*—to top-ranked pooled Web search results for each query. The judges provided hundreds of relevance judgments for each query. From here, we intersected the judgment data with our cursor data, resulting in 1,290 query-result URL pairs for which we had both explicit relevance judgments and cursor activity. These pairs formed the basis of our analysis. We computed the following features for each pair:

Clickthrough rate: Fraction of the times that URL was clicked when the query was issued (and URL returned).

Hover rate: Fraction of times that URL was hovered over when the query was issued (and URL returned).

Number of unclicked hovers: Median number of times for which the query was issued and the URL was hovered over but not clicked, as per the earlier definition. The number of unclicked hovers were selected as a feature because it was correlated with clickthrough in our previous analysis (Section 6.1.4).

Maximum hover time: The maximum time that the user spent hovering over the result per SERP instance. We take the maximum as this indicates the point where the user was most interested in the result.

As stated earlier, the clickthrough rate is commonly used to estimate the relevance of a URL to a query from behavioral data [57], and is included in this analysis as a baseline.

We computed the Pearson correlations between each feature and the human relevance judgments (represented numerically as a five-point scale ranging from 0 to 4, inclusive) independently and in combination using linear regression. Table 6.2 summarizes the findings, grouped by whether results were clicked for the query. All correlations and differences between correlations are significant at $p < 0.02$ using Fisher's z-transformation where appropriate.

The results of this analysis show that the use of cursor tracking data can improve estimates of search result relevance. Result hover features correlate better with human relevance judgments than clickthrough rates (0.46 vs. 0.42), and they lead to an improved model when combined with clickthrough (0.49 vs. 0.42). In addition, even when there are no clicks for a query, hover features show a reasonable correlation with human judgments (0.28). This is particularly important since many queries occur infrequently, resulting in little or no clickthrough data. Further analysis on the impact of query-URL popularity shows that hover features provide most value over clickthrough rate when query-URLs are less popular. There are large and significant increases in the correlation for query-URL pairs with fewer than five instances in our cursor data (0.45 hover vs. 0.35 click) and small and not significant for pairs with five or more instances (0.59 hover vs. 0.58 click). Thus cursor data appears to be especially useful when click data is less plentiful, which allows relevance estimates to be made for a much larger range of queries.

Result clicks or no clicks	Feature source	Correlation with human relevance judgments
Clicks (N=1194)	Clickthrough rate (c)	0.42
	Hover rate (h)	0.46
	Unclicked hovers (u)	-0.26
	Max hover time (d)	-0.15
	Combined[1]	**0.4**9
No clicks (N=96)	Hover rate	0.23
	Unclicked hovers	0.06
	Max hover time	0.17
	Combined[2]	**0.2**8

[1] y = 2.25 – 0.1c + 1.38h – 0.08u – 0.12d; [2] y = 0.36 + 0.80h + 0.22u + 0.30d

Table 6.2: Correlations between click and hover features and relevance judgments for queries with and without clicks.

The correlations between human judgments and unclicked hovers and hover time are interesting as well. For clicked queries, unclicked hovers and hover time are negatively correlated with relevance judgments. This result appears to contradict previous findings which suggested that hesitation over a result is a positive indicator of relevance [70, 73]. One explanation may be because clicks often occur on top-ranked results, where unclicked hovers are negatively correlated with clickthrough (as shown in Figure 6.5). For unclicked queries, we find small positive correlations between judgments and all measures. Unclicked queries have fewer relevant results, leading to more exploration lower in the search results (where unclicked hovers are positively correlated with clicks).

In this section, I showed that the correlation between explicit relevance judgments and search activity increases when cursor actions are added to clicks, especially when clicks are infrequent or unavailable.

6.3 Distinguishing between Good and Bad Abandonment

A second signal drawn from cursor information is the likely reason the user abandons a search. Abandonment occurs when searchers visit the search engine result page, but do not click. As noted in previous research [68], abandonment can suggest that users were not attracted to any of the search results (bad abandonment) or that they have found the answer directly on the SERP (good abandonment). For example, for queries like [Vancouver weather] or [WMT stock price], answers are typically shown on the SERP so there is no need for people to click through to other pages. We now examine whether features of SERP cursor behavior can distinguish between good and bad abandonment.

As reported in Li et al. [68], it may be straightforward to estimate good or bad abandonment for queries where search engines offer special interface treatment (e.g., weather updates or stock quotes). A more challenging scenario is determining whether observed abandonment for other queries is good or bad. To study this we selected queries from our log data that were short questions (ending in a question mark) which could be answered by SERP snippets. A similar query class was also studied in earlier abandonment research [68]. To identify examples of likely good abandonment in such cases, we performed some hand labeling.

Whether these short questions were answered on the SERP was determined by a human judge reviewing the results returned to users and identifying whether an answer appeared in the snippet text of results. Judgments were made for results which were hovered over for at least 100 ms, indicating that they had been attended to but not clicked on. Of the 859 queries for which the SERPs were visually inspected, 184 (21%) contained the answer in the snippet content and hence were identified as likely examples of good abandonment. The remaining 675 queries were classified as bad abandonment.

We computed summary measures that reflect how the cursor was used on the SERPs. Specifically, we looked at cursor trail length, cursor movement time, and cursor speed for each SERP, defined as follows:

Cursor trail length: Total distance (in pixels) traveled by the cursor on the SERP.

Movement time: Total time (in seconds) for which the cursor was being moved on the SERP.

Cursor speed: The average cursor speed (in pixels per second) as a function of trail length and movement time.

Feature	Abandonment Type			
	Good		*Bad*	
	M	SEM	M	SEM
Cursor trail length (px)	1084	98	1521	71
Movement time (secs)	10.3	0.9	12.8	0.6
Cursor speed (px/sec)	104	9	125	5
Number of queries	184		675	

Table 6.3: Features of cursor trails for queries associated with likely good and bad abandonment.

Table 6.3 shows the mean (M) and SEM for each measure. As can be seen from the table, the preliminary analysis reveals differences in trail length, movement time, and the speed with which users moved their cursor in good and bad abandonment queries. Cursor trails were shorter in cases where good abandonment was likely, compared to instances of bad abandonment. Searchers also spent less time moving the cursor, and moved the cursor more slowly when answers were in the snippet (good abandonment). All differences between the measures for good and bad abandonment were significant using independent measures t-tests (trail length: $t(857) = 2.58$, $p = .01$; movement time: $t(857) = 2.20$, $p = .03$; cursor speed: $t(857) = 2.17$, $p = .03$). It appears that when the answer appears on the SERP, users need to consider fewer results, and move the cursor more slowly as they examine snippet content in detail. These findings show that features of cursor trails, such as length, duration, and speed, are different for good and bad abandonment.

Diriye et al. build off this work [27] by using cursor features for classifying the reason why a user abandoned a query. Cursor movement distance was also an important feature

in distinguishing between abandonment reasons in their experiment. Like the findings presented here, Diriye et al. showed that there some some ability for cursor interactions in distinguishing between good and bad abandonment. Knowing the true reason for abandoning a query allows search engines to learn from the feedback, as it provides a clearer signal of which result snippets are attractive for a given query.

The last two sections presented ways in which cursor interactions could signal two search metrics: estimating search result relevance and distinguishing good abandonment from bad abandonment. The first signal was based on features from search result hovers, whereas the second came from features derived from cursor trails. In the first application, cursor interaction data captured for uncommon queries where strong indicators of relevance such as result clicks may occur less frequently or not at all. Analyzing click logs for a query that has been issued several times but never clicked may provide limited relevance information, but cursor behavior on the SERP associated with the query provided insight about relevance. In the second application, in cases of so-called good abandonment [68], where the content on the SERP satisfied the user's information need directly, a search result click may be unnecessary. Thus the lack of a click should not always be interpreted as a search failure; instead, cursor movements in the form of trails can help distinguish between good and bad search abandonment.

This chapter has presented correlations between cursor movements and hover and interesting search metrics. The next chapter takes large-scale cursor interactions on search results further by applying them to a searcher model to better infer document relevance, a key search metric.

Chapter 7

EXTENDING SEARCHER MODELS WITH CURSOR INTERACTIONS[1]

While a good understanding of user behavior can inform search engine designers how do develop search interfaces or lay out the information, it does not directly show how the cursor interactions can be used to improve the search engine, a central goal of this dissertation. This chapter presents work extending a dynamic Bayesian network model of clicks to include cursor hovering and page scrolling to model user search result examination behavior. With the right data, this approach enables search systems to compute more accurate parameters of result attractiveness and document relevance. These models could be used to better label Web documents for relevance to a query, improving the ranking of search results. For the top 5 search results, the extended model incorporating cursor interactions was about twice as likely to outperform a standard click model than vice versa.

7.1 Motivation

Being able to compute relevance scores from implicit feedback allows a search engine to better rank the search results for future queries. Clicks (in the aggregate) provide a clear signal that users were attracted to the search result, and numerous studies have used click data in searcher models to infer relevance scores. These searcher models (e.g., [18, 23, 93]) track the user's state as they examine search results and use the observable events (e.g., clicks) to infer search result attractiveness and document relevance. However, query logs possess inherent limitations as I noted earlier: they are unable to reveal actual user intent, provide little data about uncommon queries, and omit many interactions. Furthermore, they are uninformative for queries that have no clicks, i.e., abandoned queries.

In this chapter, I will introduce how the cursor interaction data can be used to sup-

[1]Portions of this chapter are published in SIGIR 2012 [50]

plement query and click data. Cursor movements and scrolling can be additional implicit signals of relevance. These interactions can be captured at scale and can be recorded without disrupting the user, as was shown in Section 3.2. Actions such as cursor hovering and scrolling can be translated into implicit relevance feedback when overlaid on the SERP. Here, I explore techniques to extend searcher models by using cursor hovering and scrolling activity to reveal latent variables in these searcher models to more accurately infer search result attractiveness and document relevance. To date, this is the first study that explores the potential of cursor and scrolling interactions for use in searcher models.

The contribution in this chapter is the experiment extending a popular searcher model by adding hover and scroll data, informed by our analysis of replays of user interactions on the search results page. Qualitative evidence shows from a human observer's perspective that hovering and scrolling provide insight into the user's intentions and attention as they examine the SERP. Searcher models can be improved further by estimating whether a search result was viewed based on cursor hover and scroll behavior.

7.2 Cursor Data

We recorded interaction data directly on the SERP of the Bing Web search engine using the large-scale cursor tracking method described in the Methods chapter, as the second sampled dataset, between May 26, 2011 and June 7, 2011 (see Table 3.1 for a description). The sample was drawn by user, storing every query from each user in the experiment.

To obtain a detailed understanding of user interactions with the SERP, we measured and recorded a variety of interactions with the page as well as page characteristics, such as the layout of elements on the page. The data comprised information on cursor movements, clicks, scrolling, as well as bounding boxes of certain components on the SERP and the browser's viewport size.

Figure 7.1 presents a fictional query along with the corresponding click data and client-side interactions. In this and many other cases, the cursor and scrolling data reveals additional information about the user's intent. In the above scenario, the query logs only show that a query was issued, and that some time later, the fourth result was clicked. This is useful information, but the interaction data supplement this by showing that the user was

← Time	
	Query: lady gaga concert tickets
	Cursor moves from top to hover over 3rd search result
	Cursor pauses for 3 seconds
	Text "Tour Dates Only" is hovered with the cursor
	Cursor moves to the 4th search result, pausing 1s
	User scrolls to the 5th search result, pausing 3s
	Cursor returns to the 4th search result and clicks
	Click: Result 4 [http://gaga.com/tix/]

Figure 7.1: A user searches for "lady gaga concert tickets", examines the first page of results, and clicks the 4th search result. Typical query logs contain only query and click data (bold).

active the whole time examining several results and that the user likely examined the 5th result and returned to the 4th result, indicating the 1–3 and 5th results may have been less relevant than the 4th result.

Figure 7.2 presents a fictional query that has no clicks. In typical query logs, the only recorded data would be the query text itself. The richer cursor and scrolling data here shows that the user did indeed scroll all the way to the bottom. We also see that the user paused to read through the results[2]. In this particular case, it seems reasonable to assume the user abandoned the query because they did not find what they were seeking. Thus, this query can be labeled as unsatisfying in a user-centered analysis of the logs that include cursor interaction data.

Based on these insights, we recorded several fields into the logs with the following the techniques. A detailed treatise of the different types of data is included here to allow other researchers to recreate the searcher model.

7.2.1 Cursor Positions

The JavaScript function for logging cursor positions periodically checked the cursor's x- and y-coordinates within the Web page relative to its top-left corner of the page every

[2]Query logs can compute the dwell time of a click, but only if another recorded event occurs after the click.

	Query: flourless cake recipe
	Cursor moves to the bottom-right over whitespace
	No activity for 4 seconds
	Cursor moves over to the scrollbar
↓ Time	User scrolls down half a screen
	No activity for 2 seconds
	User scrolls down half a page
	Cursor makes left-right motions over the 6th result
	User scrolls to the bottom of the page
	User quickly scrolls back up to the top
	Cursor moves to the top-right over the page
	User closes the page

Figure 7.2: A user searches for "flourless cake recipe" and scrolls to the bottom of the page, then scrolls back up and closes the window. Typical query logs contain only query and click data (bold).

250 milliseconds. As this was the largest sample yet, new coordinates were sent to the backend server only when the cursor had been moved more than eight pixels away from its previously logged position. Eight pixels correspond to approximately half a line of text on the SERP. Since cursor tracking was relative to the document, I captured cursor placement to SERP content regardless of how the user got to that position (e.g., by scrolling, or keyboard). Therefore, this approach was compatible with other behaviors such as scrolling or keyboard input.

7.2.2 Clicks

Clicks were recorded using the JavaScript onMouseDown event handling method. Thus, the backend server received log entries with location coordinates for every click, no matter whether the click occurred on a link or elsewhere on the page (even on white space containing no content that appears adjacent to or between SERP elements). In order to identify clicks on hyperlinks and differentiate them from clicks on inactive page elements, we also extracted and logged unique hyperlink identifiers that were embedded in the SERP, along with the

corresponding URL of the hyperlink. The URL helped identify the actual search result because different query sessions could have different search results or the same search results ranked differently.

7.2.3 Scrolling

The current scroll position was also recorded, i.e., the y-coordinate of the uppermost visible pixel of the SERP in the browser viewport. This coordinate was checked three times per second and was recorded whenever it had changed more than 40 pixels compared to the last logged scroll position. Forty pixels correspond to the height of about two lines of text. From this coordinate I was able to gain a number of insights into scrolling activity, including whether the user was scrolling up or down, and the maximum scroll depth in the result page, in order to understand how far down the page the user had scrolled.

7.2.4 Page Layout

Simply logging the text of what was displayed on the SERP is insufficient for reconstructing its layout since SERPs vary per query (depending on what kinds of SERP elements are shown, etc.), font sizes, and other browser preferences. To reconstruct the exact SERP layout as it was rendered in the user's browser, we recorded the positions and sizes of certain regions. The specific regions in which we were interested in were as follows: (i) top and bottom search boxes, (ii) left rail and its contained related searches, search history, and query refinement areas, (iii) mainline results area and its contained result entries, including advertisements and instant answers, and (iv) right rail.

For each region bounding box, we determined and logged the coordinates of its upper left corner as well as its width and height in pixels. Using this information, we could later map the positions of cursor positions and clicks to specific regions of the page. The recorded data also contained the size of the user's Web browser window, which combined with the scrolling activity could deduce information about the parts of the page that were visible at a particular time during the query session.

Figure 7.3 presents a screenshot of the page layout of a reconstructed SERP taken from

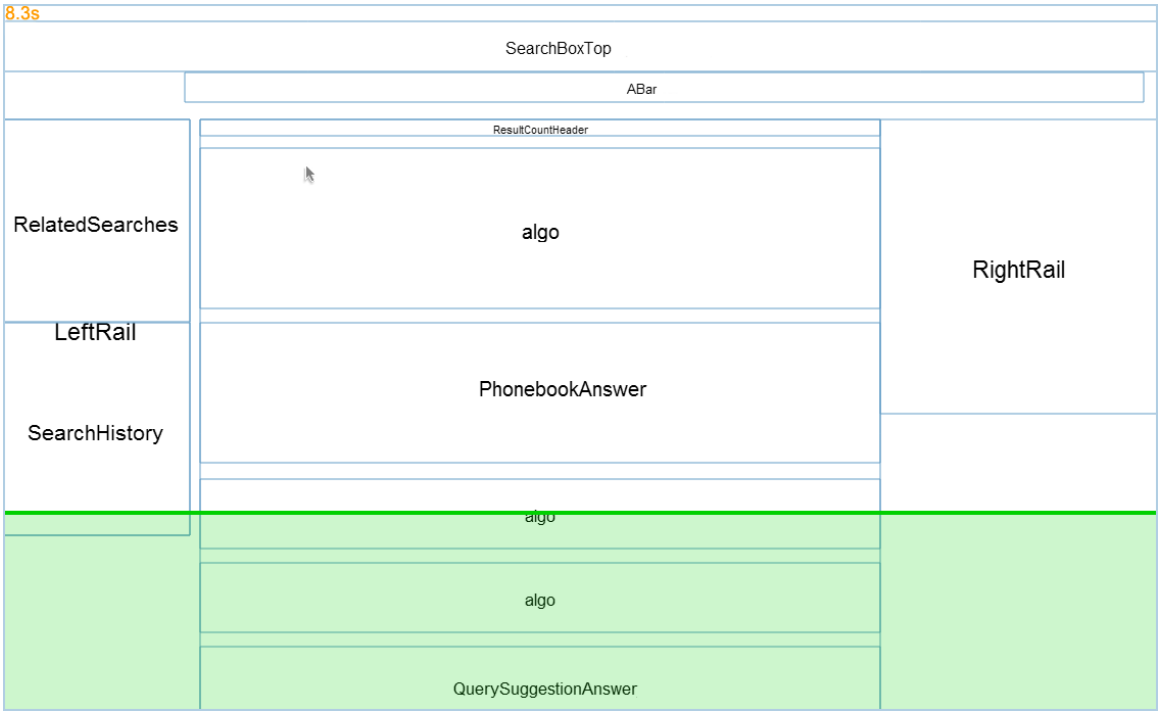

Figure 7.3: The reconstructed SERP during a query session replay. Light blue boxes outline important components, a gray pointer represents the user's cursor position, and the green area overlays off-screen portions of the Web page. The number in the top-left is the time elapsed since the start of the query session.

a query session replay. Important components are outlined in light blue boxes; the user's cursor position is shown as a gray pointer; and the green area represents parts of the Web page not visible in the browser window on the user's screen at the current time.

7.3 Exploratory Analyses

Before constructing any searcher models, I sought to obtain a deeper understanding of the recorded cursor and scrolling activity since this type of data was relatively unexplored. This included both a qualitative perspective and a more traditional quantitative analysis of the data. The findings here informed the approach I took in enhancing the searcher model.

7.3.1 Qualitative Observations

I began by reconstructing the SERP layout from the recorded logs, and developed a tool to replay the entire sequence of cursor interactions on the page in great detail. This included an outline highlighting the viewable area of the Web page (based on the dimensions of the Web browser viewport), since this would change according to the users' screen resolution and their scrolling. I then visually investigated a random sample of over a hundred replays of interactions on the search result pages made by real users. During the replays, I put myself in the users' place to determine their intent. My judgments of their intents were informed by the cursor behaviors described in prior literature. These qualitative observations were a rich way of understanding the data and provided a number of key insights that were difficult to quantify.

First, I saw that many users could only view a small portion of the Web page initially, which only displayed ads or an "Answer" element (such as the PhonebookAnswer, which shows local results and contact information, in Figure 7.3); these users would often scroll down a bit to view at least a couple of search results. The time spent pausing after a scroll suggested that they indeed examined those newly revealed search results. Thus, the newly reveal search results are highly related to an examined result. I was less confident that the user had examined all the visible search results if they did not scroll, since they often clicked a link or abandoned the query immediately after the page loaded.

Second, while I could not see where the user was actually looking, the cursor would commonly move around the page from top-to-bottom while hovering over particular areas, and then move to the scrollbar to reveal more search results, corresponding nicely with the *linear traversal hypothesis* [56]. This behavior seemed to suggest that whenever a user hovered over a search result, they had at least examined that result and the search results above it.

Third, I observed some users moving their cursor back-and-forth horizontally which I believed was evidence of them following the cursor as they read text; some users would do this quite frequently in a single query session. This corroborates previous work that observed this behavior in lab settings [70, 84] including our earlier analysis in Section 4.2,

and suggests that this behavior is specific to individual users.

Finally, I observed many sessions in which the user would move their cursor quickly and directly from the search box to the first search result, without scrolling down to view any of the lower-ranked search results. This happened often in navigational queries, so this provoked the question of whether interaction data would be more or less useful in navigational queries; I explore this later in Section 7.5.2.

7.3.2 Quantitative Summary

The raw interaction events comprised cursor positions, clicks, window scrolls, and page layout. Statistics specific to the different types of interaction data logged include:

Cursor: Users hovered over multiple search result captions (mean = 2.6, median = 2), even for navigational queries when it became clear that a single search result would suffice. This pattern of behavior has been observed in previous studies of eye tracking [25], as well as previous work on large-scale cursor tracking.

Clicks: Clicks and their corresponding targets were collected in the logs regardless of whether they were navigational clicks (clicks on a hyperlink) or interactive clicks on controls in the page (17.1% of clicks). 64.7% of all clicks were hyperlink clicks and 35.2% were non-hyperlink clicks, including re-query events (estimated from clicks on the upper or lower search boxes) totaling around 11% of all queries.

Scrolling: Window scrolling is a client-side interaction that is rarely captured in the context of Web search. Of the queries in our set, 29.7% contained at least one scroll event. 61.8% of logged interaction sequences for a query ended on a downwards scroll. As expected, there were more downward scrolls than upward scrolls, and the majority of scrolled queries (54.8%) comprised only downward scrolls. This suggests that most queries do not result in the user returning to the top of the SERP to examine search results that may be hidden following scrolling.

Figure 7.4 contrasts queries in which the user has scrolled with queries where the user did not scroll. As expected, when the user has not scrolled, the click distribution is significantly more skewed towards higher-ranked search results. The difference between the

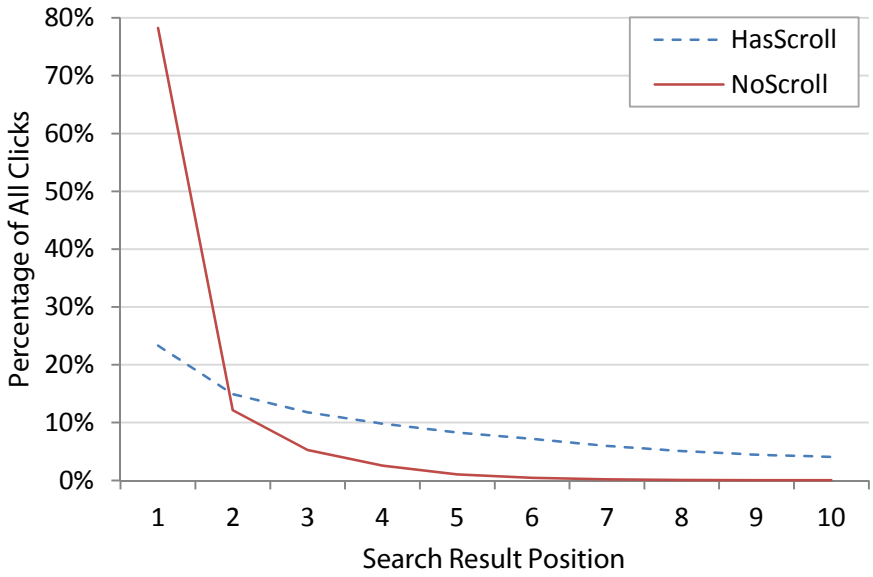

Figure 7.4: The click distributions for cases in which the user does not scroll, and when there is at least one scroll event during the query session. The distribution is heavily skewed when there is no scrolling, and almost linear when the user has scrolled.

click distributions is quite drastic: in query sessions where the user has scrolled, search results in lower-ranked positions have a fairly good chance of being selected. This is consistent with the examination hypothesis mentioned earlier, and supports a hypothesis that scrolling towards a set of search results makes it likely the user has examined those results.

The observations informed the following two hypotheses that could be applied to the searcher models: 1) when a user scrolled down, they have already examined the search results in their viewing area and those above it, and 2) when a user hovered over a search result, they have examined it and the search results above it. In the next section, I validate these hypotheses by implementing them in a traditional searcher model that uses only clicks.

7.4 Extending a Searcher Model

Searcher models are structured based on theoretical knowledge of a user's search examination process. Their internal parameters are inferred from observable data, which in turn can be applied to compute relevance label scores for search results. Label scores are position-independent and computed from the model for every query $\times$ search result. The

search results can then be re-ranked using these labels for future occurrences of the same query. Thus, a better searcher model can compute more accurate relevance labels for search ranking.

The Dynamic Bayesian Network (DBN) model [18] was the baseline model to which the extended model was compared against. The DBN model is the most cited searcher model since the Cascade Model [23] (which compared favorably to all models that came before). It compares favorably to the Cascade Model in recent evaluations [18, 103], and fares well compared to other models (e.g., [103, 104]). Thus, the DBN model serves as a solid baseline for our purposes; it provides an example searcher model in which we can focus on whether cursor data can improve a model, rather than outperforming all models, i.e., more of an analysis of the value of cursor data than strictly model development.

The DBN model is a graphical model where the nodes represent states of the user examining the search results. The model is represented formally as follows:

- E_i: the user examined the search result
- C_i: the user clicked the search result
- A_i: the search result attracted the user
- S_i: the landing page satisfied the user (relevance)

$$A_i = 1, E_1 = 1 \Leftrightarrow C_i = 1$$
$$P(A_i = 1) = a_u$$
$$P(S_i = 1 | C_i = 1) = s_u$$
$$C_i = 0 \Rightarrow S_i = 0$$
$$S_i = 1 \Rightarrow E_{i+1} = 0$$
$$P(E_{i+1} = 1 | E_i = 1, S_i = 0) = \lambda$$
$$E_i = 0 \Rightarrow E_{i+1} = 0$$

In this model, users examine search results from top to bottom, assessing at each result whether or not it is attractive enough to click (linear traversal hypothesis), which depends

only on the attractiveness of the link a_u (examination hypothesis). If they click, there is some probability s_u they will be satisfied and stop the search process; if they are not satisfied, they either return to the search results page to examine the next search result with probability λ, or abandon the search. Figure 7.5 enumerates the user states and decisions in the DBN model; the "hover above and scroll towards" state was a new observable event generated from the cursor data. Examining a search result could emit this event, but the events are not a precondition of examining a result.

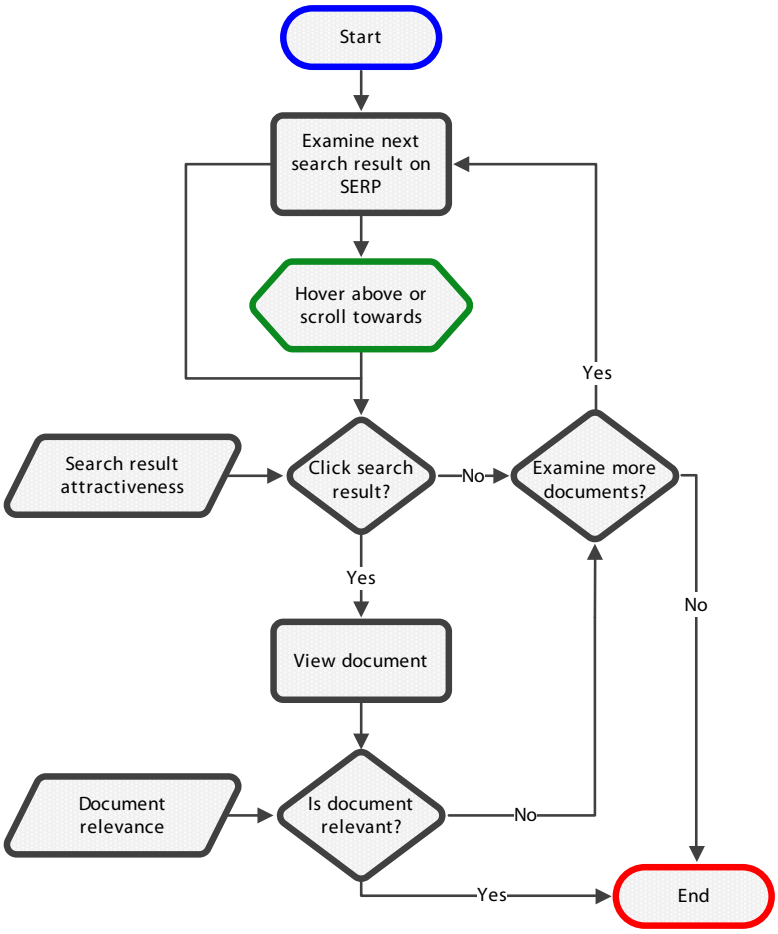

Figure 7.5: Flow diagram of the users' states in the modified Dynamic Bayesian Network model enhanced with cursor hover and scrolling data. The hexagon represents the new potentially observable events that can be captured in interaction logs.

During the exploratory analysis phase, I saw that scrolling towards a set of search results

led to a higher chance of those results being examined. Additionally, hovering over a search result similarly suggested that result and those above it were examined. These assumptions were incorporated into the searcher model by adding the following constraint:

$$(\exists h \in H : i \leq h) \vee i \in V \Rightarrow P(E_i = 1),$$

where H is the set of search result positions the user hovered over, and V is the set of all search results shown when the user scrolled. These events would reveal that the user had examined the search results, but a user examining a search result would not necessarily emit a corresponding hover or scroll event.

I reimplemented the DBN model with $\lambda = 1$, labeled *Algorithm 1* in Chapelle and Zhang [18], to simplify the inference of latent variables. Then I incorporated the additional examination constraint into the model to validate the observations in the exploratory analyses.

7.5 Experiment

Next came an experiment to see whether this extended model truly depicted the users' internal states more accurately than the existing DBN model. This section describes an experiment comparing the baseline DBN model with the modified DBN model incorporating cursor data for computing relevance labels. I define the click perplexity metric used to evaluate the model and report the experiment and results.

7.5.1 Evaluation

While the unobserved events in a searcher model cannot be directly evaluated, the model can be tested by how well it predicts clicks, the observable events. Click perplexity was evaluated in a number of other searcher model studies [31, 37, 101, 103, 104] as a measure of predicting clickthrough rates. Our evaluation used a similar methodology as the past studies in literature: query sessions were divided evenly into training and test sets, each comprising at least 5 query sessions; we only accepted one query session from each user for a particular query to prevent a small number of users from dominating the data. There were 7,341 unique queries in which at least 10 distinct users issued the query; this filtered out queries with insufficient data.

I compared *the DBN model with only click data*, as it is implemented in the literature, with *the DBN model with click and cursor data* from our logs. These data were used to train the searcher model, and the trained model was used to predict clicks in the test set[3]. Better prediction of clicks in the test set implies that the searcher model (and its inferred parameters) better reflects the result examination process. The click perplexity quantifies how much the test data surprises the trained model; it is computed for each combination of query and position as,

$$p_i = 2^{-\frac{1}{N}\sum_{n=1}^{N}(C_i^n log_2 q_i^n + (1-C_i^n) log_2 (1-q_i^n))} \tag{7.1}$$

where p_i is the perplexity in the ith position, N is the number of links, and q_i^n is the predicted click probability for the nth query session. The exponent represents the cross-entropy estimated from a probability distribution. The lowest perplexity is 1, meaning the trained model perfectly predicted the test data, while a larger perplexity means the model was less accurate in predicting the test data. Because the lower bound of the perplexity depends on the clickthrough rate of the query, the perplexity varies substantially depending on the position of the search result. Therefore, a separate perplexity value was computed for each of the top ten rank positions.

7.5.2 Results

This subsection reports on the results of my experiments. Figure 7.6 shows the computed perplexities for each position on the SERP. The baseline searcher model comprising only click data did not perform as well as the searcher model incorporating both click and scrolling data. The latter model was further improved when incorporating hover data as well, although the improvement was small since there is an overlap between the search results the user scrolls to, and the search results at or above that which the user hovers above.

It is clear that the additional cursor data improves relevance labels for search results in positions 2–5, but the prediction is the same or slightly worse for search results in positions

[3]The cursor data was only used for training the searcher model, and not for testing, i.e., I did not try to predict cursor movements and scrolling.

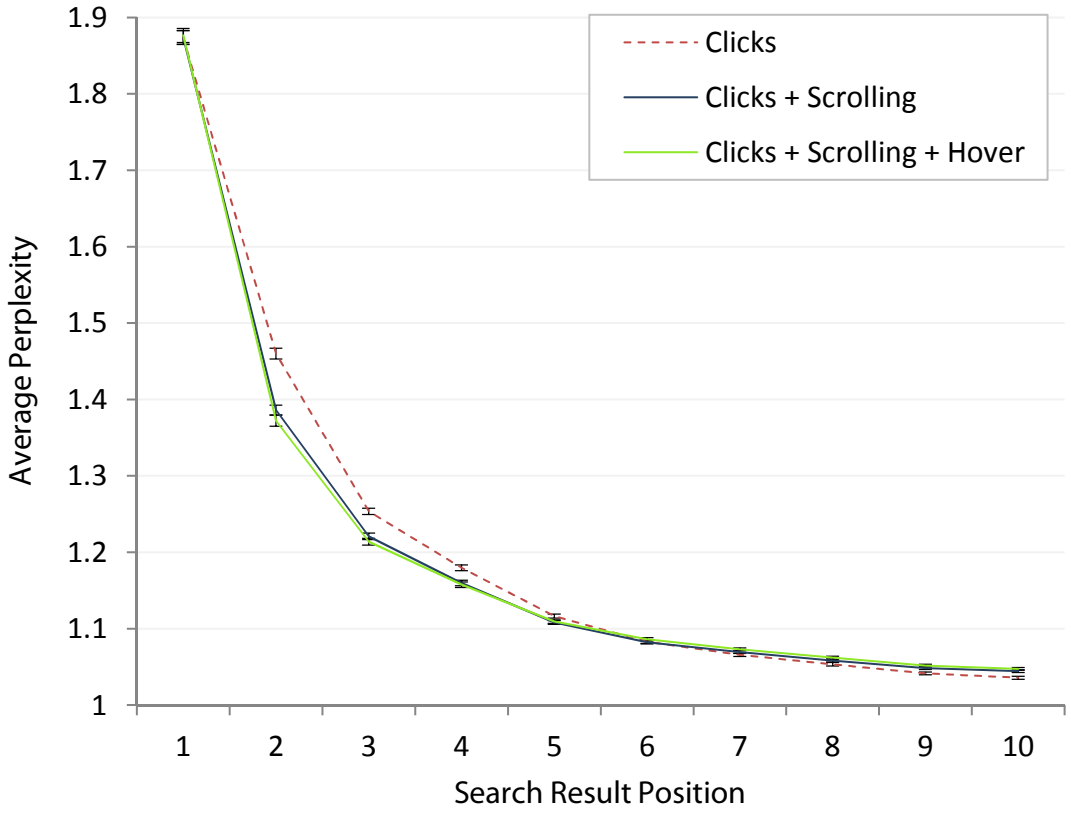

Figure 7.6: A comparison between 3 variations of the DBN model: 1) the baseline model using only click data, 2) a modified model also incorporating scrolling data, and 3) a modified model incorporating clicks, scrolling, and hover data. Lower click perplexity represents better prediction. Error bars represent the standard error of the mean.

6–10. However, users benefit more from the better prediction accuracy for the search results in higher positions since users consider them more important, so there is an overall improvement. More accurate click predictions signify that the document relevance and search result attractiveness labels are more likely to be close to true objective values of these parameters. For search results in positions 6–10, there appears to be a slight decrease in accuracy for predicted clicks in the models incorporating cursor data. I am unsure whether this is due to overfitting or noise in the data. The difference in perplexity for using clicks only compared to clicks + scrolling was significant at the $t(7340) = 8.26, p < 0.001$ level at positions 2–5. The difference in perplexity for using clicks + scrolling data compared to clicks + scrolling

+ hover data was significant at the $t(7340) = 3.07, p < 0.002$ level at positions 2 and 3, as the perplexity drops from 1.46 to 1.37 in position 2 and from 1.25 to 1.21 in position 3. Both differences were significant even after applying the Bonferroni correction.

With my collaborators, we performed two additional sets of analyses of our results designed to better understand the nature of our gains. We studied the distribution of gains and losses across the top 10 rank positions, and studied the effect of query types (navigational versus non-navigational) on our click prediction accuracy. I now report the findings of each.

Gains and Losses

For each of the 7,341 queries in our dataset, and for each rank position, we determined whether the DBN model with full cursor data (clicks + scrolling + hovers) outperformed the model with only clicks. We then computed the percentage of queries for which the model with cursor data attained a perplexity value above, below, or equal to the clicks-only model. Note that to simplify the analysis, we ignored the magnitude of difference between the models for a query. Figure 7.7 highlights the change in click prediction performance for different positions.

The findings reveal a number of things. First, there was generally no change for the first rank position. Second, the biggest gains from the cursor model came at position 2, where over 60% of queries were benefited by using the additional cursor data. Third, the fraction of queries for which the cursor model performed best decreased fairly rapidly with rank, ending with only 5% of queries benefiting from cursor data for at rank position 10. One possible explanation is that since users only scroll for a third of queries, we possess less hover and scrolling evidence from which to learn user preferences at lower ranks. Finally, for the second rank position onwards, the fraction of queries for which there is no change remains fairly constant in the 20–35% range (increasing gradually with rank). Although there were no immediately noticeable patterns in the queries with unchanged click perplexities, they need more analysis since they may represent an opportunity for additional gains, especially further down the ranking, where they represent a sizable fraction of queries.

Figure 7.7: The percentage of queries whose click perplexity was helped or hurt by adding cursor hover and scroll data to the DBN searcher model.

Effect of Query Intent

To investigate the dependence of relevance scoring on search task, which has shown to affect cursor behavior (Section 4.3), we segmented the queries into navigational and non-navigational query types to see if performance differences existed between different query intents in the models. Teevan et al. adopted a metric of click entropy as a threshold to classify navigational and non-navigational query types [91]. They showed that navigational queries classified in this manner exhibited differences in user behavior. We used the same method to segment the queries in our set, and identified 2,407 navigational queries and 3,509 non-navigational queries. The remaining 1,426 queries had a click entropy value between the navigational and non-navigational thresholds, and were removed from this part of the analysis.

Originally, we hypothesized that cursor and scrolling data may be less useful for nav-

igational queries, since the clicks can be determined more easily. However, the findings (summarized in Figure 7.8) showed that click prediction improved when cursor data was added in both navigational and non-navigational queries, particularly in higher-ranked positions; the click predictions were almost evenly improved in navigational queries as in non-navigational queries. Differences in click perplexity between all four combinations—navigational and non-navigational queries, with and without cursor data—were statistically significant at the $t(5915) \geq 7.63, p < .001$ level after applying the Bonferroni correction. We also inspected the individual queries for which the additional cursor data helped and hurt click prediction; the queries exhibited no discernible pattern. The improvements appeared to be uniform and not particular to one type of query.

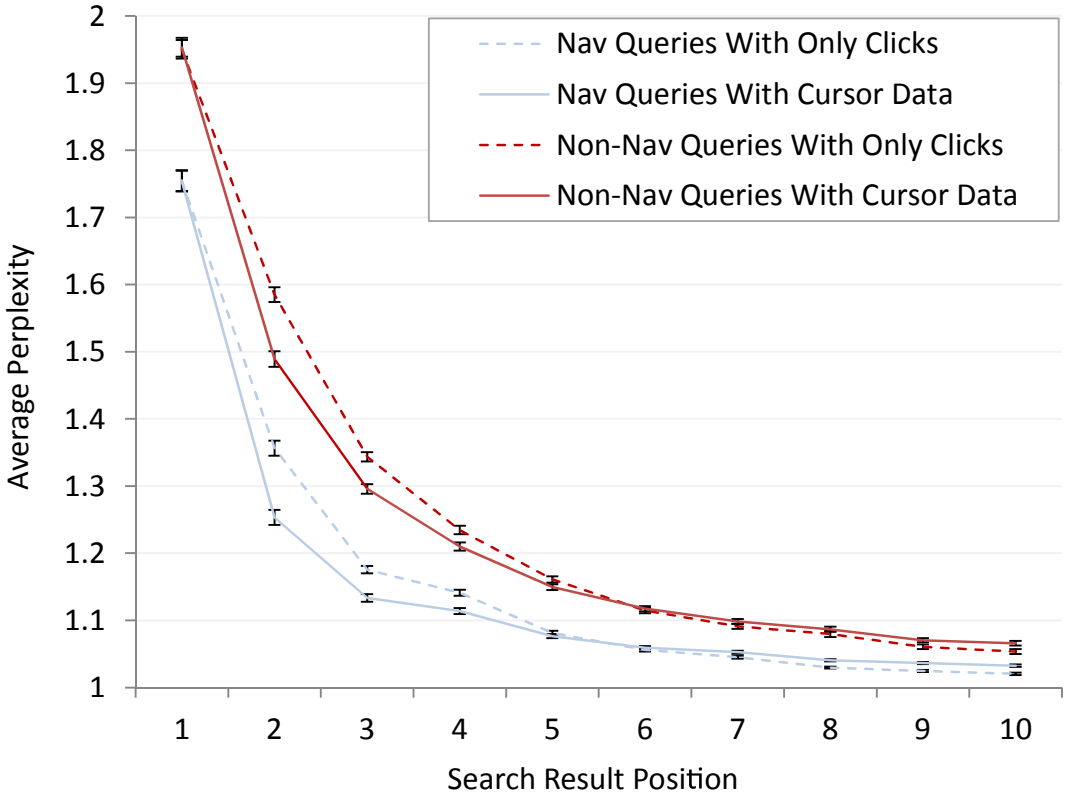

Figure 7.8: Comparison of click prediction between navigational and non-navigational queries, with and without cursor data. Lower click perplexity represents better prediction.

7.6 Limitations

The searcher model presented in this chapter was fairly simple because my goal was to demonstrate the value of new interaction data rather than develop the best overall searcher model. My searcher model does not take advantage of several metrics that have been shown to improve searcher models: click order [55], the duration between clicks [45], and temporally changing relevance of search results [90]. Another limitation is that SERPs in commercial search engines are becoming richer and more interactive. While SERPs have greatly evolved from 10 blue links, features such as *hover preview* and interactive customized displays for many types of search results have changed how users behave. Finally, the extensions I made to the searcher model were informed by an exploratory analysis of replays of user sessions; the results show that there can be benefit to incorporating non-click data in searcher models, but I have not investigated what other forms of data may similarly improve them. These limitations can be addressed in future extensions, perhaps with searcher models beyond the DBN model, incorporating previously mentioned factors such as click order, duration, and temporal relevance.

7.7 Summary

Initial exploratory findings have provided key insights about user behavior. Exploratory analyses of recorded user interactions on the SERP with both qualitative and quantitative approaches (replays and aggregate analyses) suggested that users seem to examine the search results they scroll towards, and the search results they hover over. These observations were converted into constraints in a dynamic bayesian network model, a popular searcher model originally described by Chapelle and Zhang [18]. Adding these interactions as constraints in a popular searcher model allowed it to infer new attractiveness and relevance labels for search results.

An experiment comparing the relevance labels from the original DBN model with only click data to the newly computed relevance labels in the DBN model with hover and scrolling as examination signals, showed that the additional data helped predict future clicks, implying that the labels were more accurate. After further disentangling the queries into naviga-

tional and non-navigational query intents, little difference was found between the two types of queries, and visually inspecting the queries themselves sustained our confidence that better relevance labels could be inferred across all types of queries. The experiments in this chapter have shown that by augmenting query logs with richer interaction data attainable at scale (in our case, cursor hover and scrolling), search systems can realize improvements in existing searcher models. By incorporating hover and scroll events into searcher models, the models can compute more accurate attractiveness and relevance labels for pairs of query $\times$ search result, which in turn lead to position-independent search result scores that can be ranked.

This work has implications for the design of search systems. Search companies have been processing query logs for some time now, and the amount of query logs that can be collected is limited—they cannot obtain more query and click data from a fixed number of users. However, companies can scalably and efficiently collect more search data such as cursor movements and scrolling. These can be used to improve searcher models by generating more accurate attractiveness and relevance labels for search results. Search engineers can then leverage these labels to supplement existing scoring factors for ranking the search results, such as document and link analyses algorithms. Additionally, the labels can be used for analytics and to answer questions such as, "Which search results are highly attractive but are not actually relevant?" or, "which queries are users likely to abandon because of unattractive search results?" Overall, having cursor interactions in current state-of-the-art models can potentially improve existing search engines via these methods.

Chapter 8

ADDITIONAL CONSIDERATIONS[1]

8.1 Touch Interactions

A recent survey puts mobile usage at 12.1% of all Web browsing in September 2012, near doubling from 6.7% a year ago [1][2]; and a talk by Google at a mobile convention in February 2011 notes "Roughly one in seven searches, even in the smaller categories, are happening on a mobile phone, ..."[89]. While search activity on desktops and laptops are still likely to be dominant for the near future, the growing portion of searches on mobile devices is becoming increasingly important. This evolution is an opportunity for Web search engines and other websites to begin using user interaction behavior on mobile devices for usability analysis and to inform their own design.

In touch-based interfaces such as tablets and smartphones, a cursor is not available to the user, and touch events have different meanings from cursor events. Additionally, viewing a Web page effectively on a small screen requires extensive panning and zooming, actions that can provide meaningful information. There has been little work on the utility of user interaction data in a touch-centric environment—interactions with the potential to affect Web search and other online websites. In this section, I propose methods to best take advantage of these interactions, and note the challenges in pursuing this line of research.

Touch interactions have been used for a few applications. Leiva used touch interactions to make adjustments to the stylesheet of a Web page such as element and font sizes [66]. However, there has not yet been evidence that users benefited from these adjustments. Speicher goes beyond CSS metrics and tracks a greater portion of user interactions to adapt a Web page for mobile [88]. The users in Speicher's study found the adapted mobile Web interface to be better than the baseline page. Carta et al. record individual touch

[1]Portions of this chapter are published in HCIR 2012 [47]

[2]Note that mobile usage has remained steady at around 14% between December 2012 and April 2013.

	Interaction	Description
Cursor	Scrolling	Changes the page content being shown in the browser to a general region of interest. Often, this is vertical scrolling, which can mark what the user has or has not read.
	Highlight text	The selection of text on a page. This can identify terms or phrases of significance.
	Hover	Hovering is when the cursor idles over a region on the page. Hovering over search results may be interpreted as a signal that the user examined that result.
Touch	Pan	Changes of the visible content on a device. Panning towards a new area of the page shows user attention shifting from the previous region to the new one.
	Zoom	Magnifying or shrinking specific regions on a page, often performed by double tap or pinch gestures. Zoom can indicate degree of interest based on zoom level.

Table 8.1: Example Cursor and touch interactions that can be recorded by a website, and their potential usage for identifying content of interest to the user.

interaction events *touchstart*, *touchmove*, and *touchend* to incorporate into visual timelines [17]. Finally, Nebeling et al. investigate the use of mis-touches on links (touches that don't lead to clicks on the links they intended) to automatically increase the spacing between links to decrease future errors [76]. They are able to show that the touch interactions facilitate an adaptation of Wikipedia to a mobile version that mobile users prefer over the standard Web version, but not as good as a hand-made mobile version of Wikipedia. Overall, these studies show some potential in using the touch interactions.

8.1.1 Substituting Cursor with Touch

An immediate reaction may be to simply replace cursor interactions with touch interactions for the same applications on the Web. Recording cursor coordinates becomes recording touch coordinates by changing *onmousemove* to *ontouchmove* [75]. The recorded data could include the centroid coordinates from the touches (Figure 8.1), along with timestamp. By doing so, the same applications follow—aggregating the coordinates result in heatmaps, and

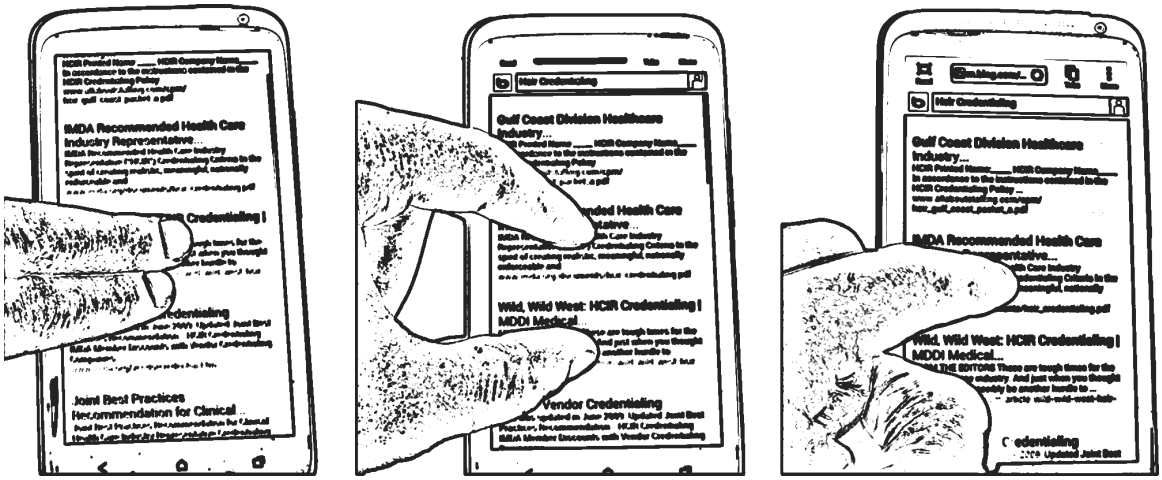

Figure 8.1: Touch interactions on a mobile device could potentially be recorded by the loaded Web site.

the individual interactions played back over time in an animation become replays. In fact, ClickTale has recently began testing a mobile version of their analytics service[3], explaining, "Businesses can now visualize what's working and what's not on their websites by seeing every swipe, pinch, tilt and tap."

Despite the potential of recording touch interaction events, using them in practice is problematic. The primary reason is that there is no evidence or rationale that the touched coordinates on the page relate to user interest or attention. While there is some justification for this in cursor coordinates, touch-enabled devices do not have a cursor that users can possibly use as a marker to aid in reading text or mark interesting parts of the page. The "cursor" on a touch-enabled device is the user's finger and unlike the cursor, it is not tracked when not clicking on the page. When users are performing gestures on the touchscreen, the area beneath a gesture such as pinching does not necessarily relate to the area the user wants to see more of; in fact, the user may be performing this gesture somewhere that does not obscure looking at the region of interest. The assumption that the touch coordinate is where the user's attention is focused is unfounded.

Additionally, current browsers do not reliably report touches. Speicher, who spent a

[3] http://research.clicktale.com/ClickTale-Mobile-Beta.html

considerable amount of time working with these events remarked, "When trying to recognise zooming gestures based on streams of touchdown, touchmove and touchup events using jQMultiTouch [a library specifically built for touch event tracking], we found out that the browsers used for testing showed different and partially unreliable behaviors in firing the corresponding touch events, ..." [88]. Although there is a W3C working draft on "Touch Events Specification", it is still up to the browser vendors to implement the specification. Additionally, the resolution in which the touch coordinates are recorded is variable; the W3C's specification offers, "the rate at which the user agent sends touchmove events is implementation-defined, and may depend on hardware capabilities and other implementation details."

Reproducing the resulting effect of a gesture is also difficult. When tracking a simple gesture such as a flick, browsers each record different points and many sample the gesture which makes it difficult to later reproduce the users' earlier action. Recreating different zoom levels from the pinch or double tap gestures is also difficult, since the zoomed level is device dependent and not all touch events trigger the JavaScript *ontouchmove* event.

8.1.2 Focusing on the Viewport

A website that tracks the viewport can store the bounding boxes of these viewing areas and how long the user spent in each. These bounding boxes are easily recorded as they are relatively small in size (bytes), and aggregate well once collected. Generating heatmaps from the aggregated bounding boxes produces an easy visualization of which parts of the Web page the users focused on (Figure 8.2). When generating the heatmap, greater weight can be assigned to areas where there was a higher degree of zoom. Besides the information that users spent in a viewport, the act of moving the viewport away from an area can be useful information as well. A short dwell time may indicate that a user did not find an area on the page interesting after glancing at it, while a long dwell time indicates the user has read the contents in that region. It may also be useful to analyze which parts of the screen the user is often attending, since after zooming, the user is probably not equally likely to be looking at each part of the page.

Figure 8.2: An illustrative example of a heatmap on a Web page generated from mobile device users' viewport data.

In search, where results are often presented as vertical lists, the viewport plays a key role. Search engines can determine which results are on the users' screens at any given time, and how much of the snippet they can see. Typically, Web search engines like Bing and Google can only present two or three search results at a time on a mobile device, and like ViewSer [63], touch-enabled smartphones can determine which results the user is looking at and how much time they spend examining the snippet. Using this information, websites can then use this information in traditional information retrieval models, such as searcher models [18] or learning to rank systems [11].

8.1.3 Looking Forward

As a portion of online user activity such as Web search moves to touch-enabled mobile devices, online services will begin thinking about recording these user interactions as they do in traditional cursor-based systems such as desktops. While the initial reaction may be to simply replace cursor coordinates with touch coordinates, this is impractical because touch coordinates do not hold the same meaning as cursor coordinates. Touch events typically are region-free methods for navigating the page, and there is no rationale behind the assumption that users are attending to the specific touch coordinates. Additionally, technical difficulties in recording fine-grained touch coordinates prevent applicability of these events.

However, tracking the viewport coordinates can be tremendously useful in noticing where on the page a user is attending to, especially on small screens, where zooming and panning is a necessity. I believe this is the data that should be recorded and analyzed, and as a usability tool, can potentially be used to improve the design of websites and search engines.

8.2 User Privacy

Conducting research involving behavioral data requires careful consideration of users' privacy. Recording cursor interactions adds to the click and query data already being collected, which may reveal users' additional intent that they prefer to keep private. Because people use the cursor differently, and due to the level of detail at which this data is recorded, it may be possible for cursor movements to unintentionally identify particular users. While my research does not pursue this route, it is important during the research to separate personal

identifiable information from the behavioral data. Notions of k-anonymity [65] or l-diversity [71] can be potential guidelines on how to validate the anonymity of the behavioral data. However, it is difficult to strictly follow those guidelines with query logs because queries are often sufficiently unique that some individuals can be identified through numerous types of attacks [58], rendering the data difficult to anonymize correctly.

The privacy concerns raised by this work centers on whether there are additional identifying information in cursor interaction data that are not present in query and click data—data generally accepted to be collected and used to improve the design of search engines. Currently, the cursor interactions are unlikely to contain compromising information about a user. But one concern is that while some users are aware that clicks and queries can be captured because the browser indicates when data is uploaded, my method of remotely capturing cursor interactions does not cause such notifications. Certainly, websites can prompt the user and ask for informed consent before collecting this information, which would alleviate this concern. However, what data users may or may not be aware can be collected from them is a question of education as well as policy in how websites and browsers should negotiate data transmission. It may be useful to explore how interfaces can balance being informative about potential cursor interaction data collection, while not being disruptive.

Another question may be what model of opt-in or opt-out is best; a simple choice of opt-in might not to garner enough data to be useful, and thus search engine results may suffer from the tragedy of the commons, since to a single person, opting out has very little effect on their own experience but reduces the data available to other users. An opt-out model is less likely to suffer from the same consequences but many users will probably not be aware of the option. Regardless of the initial default, one reasonable arrangement could be to allow only users who opt-in to benefit from the cursor interactions collected from other users in their search experience; users who opt-out receive an experience not affected by the interaction data from other users.

In any case, user privacy remains a controversal topic in any situation where additional user information is being collected without explicit consent, as cursor interactions could be. There is no simple solution that satisfies all stakeholders today, and this area continues to warrant further study.

Chapter 9

CONCLUSION

Understanding how people interact with search engines is important in improving search quality. Popular Web search engines currently analyze queries and clicked results, but these actions provide limited signals regarding search interaction. In this dissertation, I have explored how cursor data, which can be easily collected at scale, can be used to augment more commonly-used click measures to provide a richer picture of how searchers interact with search result pages. Cursor interaction data can supplement existing query and click data, especially in tail queries where query logs are sparse. Cursor movements, scrolling, and hesitations are noisy but can provide detailed information if collected efficiently and analyzed properly.

I have described how people use their cursors in relation to their eye-gaze, clarifying the relationship between cursor positions and visual attention. The findings show that the alignment between gaze and cursor is affected by the factors of user, task, and time. Cursor behavior is also a factor, and five distinct behaviors are drawn from the literature and my study. Additional analysis reported in this dissertation has examined interactions on search results to see how users behave with their cursor by hovering or moving it while they search, and how cursor movements can improve estimates of search results relevance and distinguish good from bad search abandonment.

I have also demonstrated how cursor interactions can be used to build user models that have practical applications. This involves understanding the behaviors surrounding the cursor interactions, such as whether the user is using the cursor to examine the page, read text, click, preparing to interact with Browser controls, or leaving it inactive. These behaviors and other factors such as the user, task, and time aspects relate to how well the cursor is a proxy for measuring visual attention. Features from cursor interactions like the behavioral pattern and time variables can be used to predict where a user is looking better

than simply using the current cursor position; I have demonstrated this in a multiple linear regression model, and I discussed possible non-linear models with the same goal.

Specific to search, scrolling and hovering were found in observations to relate to search result relevance, and I have developed a way to incorporate these interactions into an existing searcher model. Hover and scroll are additional observable events that can more accurately infer the unobserved variables to compute more accurate relevance and attractiveness scores for the search results (evaluated through more accurate click prediction). In both the gaze prediction and the relevance labeling tasks, I showed that models incorporating cursor interactions beat the state-of-the-art methods that did not include cursor data.

Overall, cursor interaction data can be useful in search for queries where click data is available by supplementing it (there are more hovers and scrolling data than click data) and in instances where no clicks are available, cursor data can substitute for some of its applications. But cursor data has qualitative uses as well. Usability tools that use cursor behavior (e.g., [3]) may be useful to search quality analysts. For aggregate analysis of cursor movements, heatmaps (such as those in Figure 6.1) can show where the interaction took place for different SERP features or queries, by allowing analysts to quickly see aggregate behavior across multiple query sessions or queries. They may be useful for determining whether users notice new features and how cursor behavior changes following their introduction. The work presented in this dissertation adds to the general understanding of how users examine search results, which is typically difficult to study in naturalistic settings on a large-scale, and demonstrates utility of these data in search-related applications. There may be broader applications beyond Web search: commercial Web analytics companies offer cursor tracking services, but are limited in how they use this data (typically heatmaps and session replays). Search is one domain where I have shown numerous applications beyond heatmaps and replays, and the same may be possible for different types of websites.

I have demonstrated that *users' mouse cursor interactions can be collected efficiently on the Web* by successfully deploying tool for cursor tracking at scale and describing an efficient method for doing so. In Chapters 4 and 6, I have shown how cursor interactions can be *used to understand users' search behaviors.* And finally, I have shown that cursor interactions *can be useful in the design of Web search engines* by extending existing tech-

niques used in designing search engines with cursor interactions and showing that adding cursor interactions into the models improve these state-of-the-art techniques for predicting visual attention and computing relevance scores for ranking results. Overall, I have provided evidence that cursor interaction data has the potential to substantially improve Web search. Collecting this new source of data will enable mining and analysis of rich user data that can provide additional context to the user's search.

Alternate forms of user interaction tracking may involve touch devices where the cursor is not the typical input technique, or high-frequency, high-resolution cameras attached to mobile devices may be practical to track where a user is looking, if the user consents. In the medium term, cursor interactions are here to stay and I believe deeper mining of richer interaction data to understand searchers, particularly cursor interactions, is an important research direction to pursue in Web search. Cursor tracking has tremendous potential and I believe this is the right time and the right opportunity for Web search.

Bibliography

[1] Statcounter global stats: Mobile vs. desktop from sep 2011 to sep 2012. Retrieved September 21, 2012 from `http://gs.statcounter.com/#mobile_vs_desktop-ww-monthly-201109-201209`.

[2] Farooq Ahmad and Grzegorz Kondrak. Learning a spelling error model from search query logs. In *Proceedings of HLT-EMNLP*, pages 955–962, 2005.

[3] Ernesto Arroyo, Ted Selker, and Willy Wei. Usability tool for analysis of web designs using mouse tracks. In *Proceedings of CHI Extended Abstracts*, pages 484–489, 2006.

[4] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *Proceedings of WWW*, pages 203–212, 2006.

[5] Anne Aula, Päivi Majaranta, and Kari-Jouko Räihä. Eye-tracking reveals the personal styles for search result evaluation. In *Proceedings of INTERACT*, pages 1058–1061, 2005.

[6] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 2nd edition, 2011.

[7] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of ACL*, pages 26–33, 2001.

[8] Charles Bloom. LZP: a new data compression algorithm. In *Proceedings of DCC*, pages 425–425, 1996.

[9] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, and Sebastiano Vigna. Query suggestions using query-flow graphs. In *WSDM Workshop on Web Search Click Data*, pages 56–63, 2009.

[10] Sergei Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Comput. Netw. ISDN Syst.*, pages 107–117, 1998.

[11] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of ICML*, pages 89–96, 2005.

[12] Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. What do you see when you're surfing?: using eye tracking to predict salient regions of web pages. In *Proceedings of CHI*, pages 21–30, 2009.

[13] Georg Buscher, Andreas Dengel, and Ludger van Elst. Eye movements as implicit relevance feedback. In *Proceedings of CHI Extended Abstracts*, pages 2991–2996, 2008.

[14] 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 SIGIR*, pages 42–49, 2010.

[15] Georg Buscher, Ludger van Elst, and Andreas Dengel. Segment-level display time as implicit feedback: a comparison to eye tracking. In *Proceedings of SIGIR*, pages 67–74, 2009.

[16] Georg Buscher, Ryen W. White, Susan Dumais, and Jeff Huang. Large-scale analysis of individual and task differences in search result page examination strategies. In *Proceedings of WSDM*, pages 373–382, 2012.

[17] Tonio Carta, Fabio Paternò, and Vagner Santana. Support for remote usability evaluation of web mobile applications. In *Proceedings of SIGDOC*, pages 129–136, 2011.

[18] Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of WWW*, pages 1–10, 2009.

[19] Mon Chu Chen, John R. Anderson, and Myeong Ho Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *Proceedings of CHI Extended Abstracts*, pages 281–282, 2001.

[20] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *Proceedings of IUI*, pages 33–40, 2001.

[21] Michael J. Cole, Jacek Gwizdka, Chang Liu, Ralf Bierig, Nicholas J. Belkin, and Xiangmin Zhang. Task and user effects on reading patterns in information search. *Interacting with Computers*, 23(4):346–362, 2011.

[22] Lynne Cooke. Is the mouse a poor man's eye tracker? In *Proceedings of STC*, pages 252–255, 2006.

[23] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of WSDM*, pages 87–94, 2008.

[24] Silviu-Petru Cucerzan and Eric Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of EMNLP*, pages 293–300, 2004.

[25] Edward Cutrell and Zhiwei Guan. What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of CHI*, pages 407–416, 2007.

[26] Laurence Peter Deutsch. Gzip compressed data format specification. Available at `ftp://ftp.uu.net/pub/archiving/zip/doc/`, 1992.

[27] Abdigani Diriye, Ryen White, Georg Buscher, and Susan Dumais. Leaving so soon?: understanding and predicting web search abandonment rationales. In *Proceedings of CIKM*, pages 1025–1034, 2012.

[28] Doug Downey, Susan Dumais, Dan Liebling, and Eric Horvitz. Understanding the relationship between searchers' queries and information goals. In *Proceedings of CIKM*, pages 449–458, 2008.

[29] Andrew Duchowski. *Eye Tracking Methodology: Theory and Practice*. Methods in molecular biology. Springer-Verlag London Limited, 2007.

[30] Susan T. Dumais, Georg Buscher, and Edward Cutrell. Individual differences in gaze patterns for web search. In *Proceedings of IIiX*, pages 185–194, 2010.

[31] Georges E. Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of SIGIR*, pages 331–338, 2008.

[32] Henry A. Feild, James Allan, and Rosie Jones. Predicting searcher frustration. In *Proceedings of SIGIR*, pages 34–41, 2010.

[33] Jeremy Goecks and Jude Shavlik. Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of IUI*, pages 129–132, 2000.

[34] Sharad Goel, Andrei Broder, Evgeniy Gabrilovich, and Bo Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *Proceedings of WSDM*, pages 201–210, 2010.

[35] Cristina González-Caro, Liliana Caldero-Benavides, Ricardo Baeza-Yates, Libertad Tansini, and Devdatt Dubhashi. Web queries: the tip of the iceberg of the user's intent. In *WSDM Workshop on User Modeling on the World Wide Web*, 2011.

[36] Carrie Grimes, Diane Tang, and Daniel M. Russell. Query logs alone are not enough. In *WWW Workshop on Query Log Analysis*, 2007.

[37] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of WWW*, pages 11–20, 2009.

[38] Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of WSDM*, pages 124–131, 2009.

[39] Qi Guo and Eugene Agichtein. Exploring mouse movements for inferring query intent. In *Proceedings of SIGIR*, pages 707–708, 2008.

[40] Qi Guo and Eugene Agichtein. Ready to buy or just browsing?: detecting web searcher goals from interaction data. In *Proceedings of SIGIR*, pages 130–137, 2010.

[41] Qi Guo and Eugene Agichtein. Towards predicting web searcher gaze position from mouse movements. In *Proceedings of CHI*, pages 3601–3606, 2010.

[42] Qi Guo and Eugene Agichtein. Beyond dwell time: Estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of WWW*, pages 569–578, 2012.

[43] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[44] David Hauger, Alexandros Paramythis, and Stephan Weibelzahl. Using browser interaction data to determine page reading behavior. In *Proceedings of UMAP*, pages 147–158, 2011.

[45] Yin He and Kuansan Wang. Inferring search behaviors using partially observable markov model with duration (pomd). In *Proceedings of WSDM*, pages 415–424, 2011.

[46] Yoshinori Hijikata. Implicit user profiling for on demand relevance feedback. In *Proceedings of IUI*, pages 198–205, 2004.

[47] Jeff Huang and Abdigani Diriye. Web user interaction mining from touch-enabled mobile devices. In *HCIR Workshop*, 2012.

[48] Jeff Huang and Anna Kazeykina. Optimal strategies for reviewing search results. In *Proceedings of AAAI*, pages 1321–1326, 2010.

[49] Jeff Huang, Ryen W. White, and Georg Buscher. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of CHI*, pages 1341–1350, 2012.

[50] Jeff Huang, Ryen W. White, Georg Buscher, and Kuansan Wang. Improving searcher models using mouse cursor activity. In *Proceedings of SIGIR*, pages 195–204, 2012.

[51] Jeff Huang, Ryen W. White, and Susan Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *Proceedings of CHI*, pages 1225–1234, 2011.

[52] David A. Huffman. A method for the construction of minimum redundancy codes. In *Proceedings of IRE*, pages 1098–1101, 1951.

[53] J. Ianniello. Time delay estimation via cross-correlation in the presence of large estimation errors. *IEEE Trans. Acoust., Speech, Signal Process.*, 30(6):998–1003, 1982.

[54] Bernard J. Jansen and Amanda Spink. How are we searching the world wide web?: a comparison of nine search engine transaction logs. *Information Processing & Management*, 42(1):248–263, 2006.

[55] Shihao Ji, Ke Zhou, Ciya Liao, Zhaohui Zheng, Gui-Rong Xue, Olivier Chapelle, Gordon Sun, and Hongyuan Zha. Global ranking by exploiting user clicks. In *Proceedings of SIGIR*, pages 35–42, 2009.

[56] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of SIGIR*, pages 154–161, 2005.

[57] 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. *ACM Trans. Inf. Syst.*, 25(2):1–27, 2007.

[58] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. "i know what you did last summer": query logs and user privacy. In *Proceedings of CIKM*, pages 909–914, 2007.

[59] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In *Proceedings of WWW*, pages 387–396, 2006.

[60] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of SODA*, pages 668–677, 1998.

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

[62] Michael H. Kuhn, Horst Tomaschewski, and Hermann Ney. Fast nonlinear time alignment for isolated word recognition. In *Proceedings of ICASSP*, pages 736–740, 1981.

[63] Dmitry Lagun and Eugene Agichtein. Viewser: Enabling large-scale remote user studies of web search examination and interaction. In *Proceedings of SIGIR*, pages 365–374, 2011.

[64] Henry A. Landsberger. *Hawthorne Revisited: A Plea for an Open City.* Cornell University, 1957.

[65] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. IEEE, 2006.

[66] Luis A. Leiva. Restyling website design via touch-based interactions. In *Proceedings of MobileHCI*, pages 599–604, 2011.

[67] Luis A. Leiva and Enrique Vidal. Assessing users' interactions for clustering web documents: a pragmatic approach. In *Proceedings of Hypertext*, pages 277–278, 2010.

[68] Jane Li, Scott Huffman, and Akihito Tokuda. Good abandonment in mobile and pc internet search. In *Proceedings of SIGIR*, pages 43–50, 2009.

[69] Richard Harold Lindeman, Peter Francis Merenda, and Ruth Z. Gold. *Introduction to Bivariate and Multivariate Analysis*, page 119. Scott Foresman, Glenview IL, 1980.

[70] Chen-Chung Liu and Chen-Wei Chung. Detecting mouse movement with repeated visit patterns for retrieving noticed knowledge components on web pages. *IEICE - Trans. Inf. Syst.*, E90-D(10):1687–1696, 2007.

[71] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[72] Erik Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, 2002.

[73] Florian Mueller and Andrea Lockerd. Cheese: Tracking mouse movement activity on websites, a tool for user modeling. In *Proceedings of CHI Extended Abstracts*, pages 279–280, 2001.

[74] Vidhya Navalpakkam, LaDawn Jentzsch, Rory Sayres, Sujith Ravi, Amr Ahmed, and Alex Smola. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. 2013.

[75] Michael Nebeling and Moira Norrie. jqmultitouch: lightweight toolkit and development framework for multi-touch/multi-device web interfaces. In *Proceedings of EICS*, pages 61–70, 2012.

[76] Michael Nebeling, Maximilian Speicher, and Moira Norrie. W3touch: metrics-based web page adaptation for touch. In *Proceedings of CHI*, pages 2311–2320, 2013.

[77] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of InfoScale*, 2006.

[78] James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Commun. ACM*, 45:50–55, September 2002.

[79] Anand Rajaraman. More data usually beats better algorithms. Available at `http://anand.typepad.com/datawocky/2008/03/more-data-usual.html`, 2008.

[80] Erik D. Reichle, Alexander Pollatsek, and Keith Rayner. E-z reader: A cognitive-control, serial-attention model of eye-movement behavior during reading. *Cognitive Systems Research*, 7(1):4–22, 2006.

[81] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of WWW*, pages 521–530, 2007.

[82] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *Proceedings of TREC*, 1994.

[83] Kerry Rodden and Xin Fu. Exploring how mouse movements relate to eye movements on web search results pages. In *SIGIR Workshop on Web Information Seeking and Interaction*, pages 29–32, 2007.

[84] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *Proceedings of CHI Extended Abstracts*, pages 2997–3002, 2008.

[85] Bracha Shapira, Meirav Taieb-Maimon, and Anny Moskowitz. Study of the usefulness of known and new implicit indicators and their optimal combination for accurate inference of users interests. In *Proceedings of SAC*, pages 1118–1119, 2006.

[86] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33:6–12, 1999.

[87] Karen Sparck Jones. *A statistical interpretation of term specificity and its application in retrieval*, pages 132–142. Taylor Graham Publishing, London, UK, 1988.

[88] Maximilian Speicher. W3touch: Crowdsourced evaluation and adaptation of web interfaces for touch. Master's thesis, ETH Zürich, 2012.

[89] Jason Spero. The time for mobile is now. Presented at *thinkmobile with Google*, 2011.

[90] Ramakrishnan Srikant, Sugato Basu, Ni Wang, and Daryl Pregibon. User browsing models: relevance versus examination. In *Proceedings of KDD*, pages 223–232, 2010.

[91] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of SIGIR*, pages 163–170, 2008.

[92] Luis A. Leiva Torres and Roberto Vivo Hernando. (smt) real time mouse tracking registration and visualization tool for usability evaluation on websites. In *Proceedings of IADIS WWW/Internet*, pages 187–192, 2007.

[93] Kuansan Wang, Nikolas Gloy, and Xiaolong Li. Inferring search behaviors using partially observable markov (pom) model. In *Proceedings of WSDM*, pages 211–220, 2010.

[94] Kuansan Wang, Toby Walker, and Zijian Zheng. Pskip: estimating relevance ranking quality from web search clickthrough data. In *Proceedings of KDD*, pages 1355–1364, 2009.

[95] Colin Ware and Harutune H. Mikaelian. An evaluation of an eye tracker as a device for computer input. In *Proceedings of CHI*, pages 183–188, 1987.

[96] Ingmar Weber and Carlos Castillo. The demographics of web search. In *Proceedings of SIGIR*, pages 523–530, 2010.

[97] Terry Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.

[98] Ryen White and Steven Drucker. Investigating behavioral variability in web search. In *Proceedings of WWW*, pages 21–30. ACM, 2007.

[99] Heino Widdel. *Operational problems in analysing eye movements*, pages 21–29. Elsevier Science Ltd., New York, 1984.

[100] Jacob O. Wobbrock, Kristen Shinohara, and Alex Jansen. The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. In *Proceedings of CHI*, pages 1639–1648, 2011.

[101] Yujiu Yang, Xinyi Shu, and Wenhuang Liu. A probability click tracking model analysis of web search results. In *Proceedings of ICONIP*, pages 322–329, 2010.

[102] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of CHI*, pages 246–253, 1999.

[103] Yuchen Zhang, Dong Wang, Gang Wang, Weizhu Chen, Zhihua Zhang, Botao Hu, and Li Zhang. Learning click models via probit bayesian inference. In *Proceedings of CIKM*, pages 439–448, 2010.

[104] Feimin Zhong, Dong Wang, Gang Wang, Weizhu Chen, Yuchen Zhang, Zheng Chen, and Haixun Wang. Incorporating post-click behaviors into a click model. In *Proceedings of SIGIR*, pages 355–362, 2010.

[105] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Appendix A

Techniques for Compressing Cursor Coordinates

While recording cursor interactions can be done without disrupting the user directly, the overhead of recording the cursor movements and transmitting this data to the server can be substantial. For instance, swiping the mouse from left to right can generate a hundred {x, y} cursor coordinate pairs, which over a minute of interaction can lead to nearly 1MB of data being sent to the analytics server. This Appendix presents experiments to compress the trail a cursor makes on a page, taking advantage of the fact that not every cursor coordinate has equal value to the user of the data. This problem is particularly important for situations where bandwidth is limited, such as a mobile device or in locations with a slow Internet connection. And while there are a limited number of services offering cursor tracking features, having efficient cursor tracking methods will benefit all Web users. One Web analytics company may be used by a thousand websites which in turn may serve a million users.

This Appendix presents a rigorous evaluation of 10 compression algorithms for 3-dimensional data (x-coordinate, y-coordinate, and time), evaluated with multiple metrics over for the compressed data. The work was done in collaboration with Luis Leiva. The evaluation is conducted across both a dataset collected from a lab study and a dataset from a live web page with real users. We show that different compression techniques may be useful in different situations; the situations can reflect a desire for consuming less bandwidth, better client-side performance, more accurate replication of the original data, or a combination of all three. By improving the compression methods used to record cusor positions, tracking cursor interactions moves towards being a more practical, scalable technology.

For each combination of compression algorithm and dataset, we measured data size reduction (compression ratio) and performance (compression time). For lossy compression, we also computed the distance between the compressed trail and the original trail from

which it was derived. This measures how well the original trail can be replicated via the lossy-compressed trail, for situations where reproducing the original trail is useful. We define *trail replication* as the per-pixel similarity measure defined by the following distance metric:

$$\tau = \int_0^T \frac{1}{|\vec{g}|} \sqrt{\sum_{i=1}^{d} \left[\vec{c}_i(t) - \vec{g}_i(t)\right]^2} \, \mathrm{d}t \tag{A.1}$$

where $\vec{c}$ represents the vector of compressed cursor coordinates, and $\vec{g}$ represents the vector of the original cursor coordinates. Equation A.1 therefore computes the sum of distances between the coordinates, normalized by the number of coordinates. Given that $\vec{c}$ and $\vec{g}$ have different lenghts, to compute (A.1) we iterate over $\vec{g}$ and find the closest point in time from $\vec{c}$.

While a compression algorithm may not be able to exactly reproduce the entire trail, it may capture the important points in the trail which may be sufficient for many applications. We used the points before a click as a proxy for an important point in the trail. So for a second distance metric, we computed Equation A.1 only for coordinates occurring just before a click. This measure represents how well the compression algorithms perform when only the points before a click matter.

A.1 Data

We used two datasets in our evaluation to achieve generalizbility of the compression methods. One dataset was from a lab study, which we will refer to as LAB while the other was the tracking script deployed on a website accessible to the public, which we will refer to as LIVE.

The LAB dataset was collected in Feild et al. during an eye-tracking study of web searcher frustration in October 2009 [32]. The dataset consists of full interaction logs and sensor readings for 30 participants who were asked to conduct search tasks. During the study, coordinates for the cursor were recorded by a toolbar that captured cursor positions when they changed and stored them in logs. We removed 5 outliers from this dataset for users

whose time at the lab spanned over two consecutive days, leaving 25 participants in the dataset.

The LIVE dataset was collected from a live website with the script described earlier that captured raw cursor events in JavaScript. Events were buffered at regular time intervals and sent to our server. The website was an informational resource listing the best paper awards in computer science; the content on the page was lengthy and included numerous clickable links to navigate the page and to search for the award-winning papers. The data was gathered between June and September 2012, and totaled 12K visitor logs. After removing outliers using inter-quartile range introspection, this dataset contained 10,471 interaction logs from 7,064 unique visitors.

There are some notable differences between the LIVE and LAB datasets. The users in the LAB dataset were brought into a lab, asked to perform search tasks, and the data was recorded with a toolbar. In contrast, the users in the LIVE dataset were browsing *in situ* in their natural environments and the data was recorded using a tracking script on the website.

A.2 Lossless Compression

There are two fundamentally different types of data compression: lossless and lossy. Lossless compression involves a transformation of the representation of a data set such that it is possible to reproduce *exactly* the original data set by performing a decompression transformation. Lossy compression is a representation that allows us to reproduce an approximation to the original data set. In other words, lossless compression allows the server to recreate the exact cursor trail recorded, while lossy compression can typically compress the data more.

Five standard encoding techniques comprise our set of lossless algorithms: Delta, GZip, Huffman, LZW, and LZ4. They are commonly used in various compression utilities for file compression. These algorithms were selected for their popularity and appropriateness for the data.

Delta compression refers to compressing data in the form of a sequence of differences between the original data and the follow-up changes performed to such data. A delta-

compressed cursor point $\vec{c}$ at time t is

$$\Delta\vec{c}(t) = \left\{\vec{g}(t) - \vec{g}(t-1)\right\} \quad \forall t > 0$$

where $\vec{g}$ is a d-dimensional point from the original cursor data.

Huffman compression [52] is a form of statistical encoding, where a binary tree is built from character counts. Then, symbols from the original (uncompressed) string are replaced by their corresponding binary codes. The more frequent a symbol is encoded, the shorter its bit-sequence.

The remaining lossless encoding techniques (GZip, LZW, and LZ4) are based on the milestone algorithm LZ77 [105]. In this encoding algorithm, repeated strings are replaced by back-references linking to the previous location of that identical string. Concretely: *1)* GZip [26] is based on the deflate algorithm, a combination of LZ77 and Huffman, using a sliding window during compression; *2)* LZW [97] is an optimization of LZ78, in which data are encoded by means of explicit dictionary entries; *3)* LZ4 is an improvement over LZP [8], a variation of LZ77, using finite context Markov prediction.

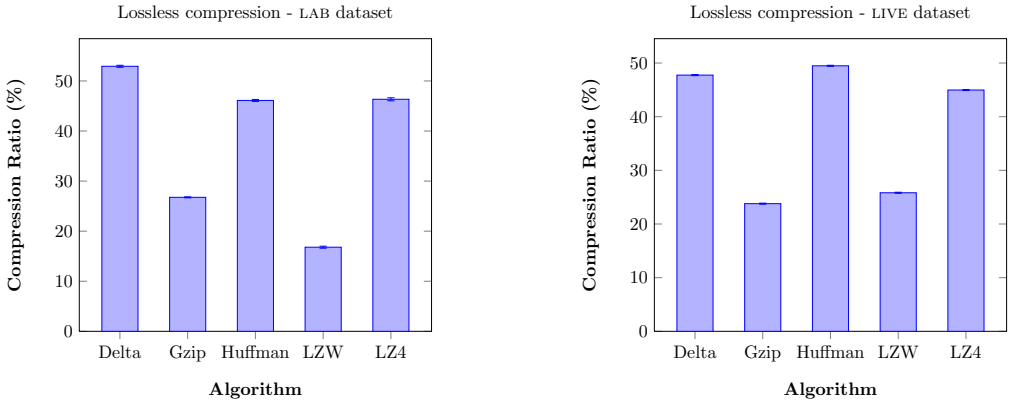

Figure A.1: Compression ratio for lossless algorithms (lower is better). Error bars denote 95% confidence intervals.

Relative compression performance is reported in Figure A.1. The results show that Delta, Huffman, and LZ4 behaved similary, reducing the data size by around 50% in both the LAB and LIVE datasets. Gzip and LZW were able to achieve the most compression, reduced the data size by approximately 80% in both datasets.

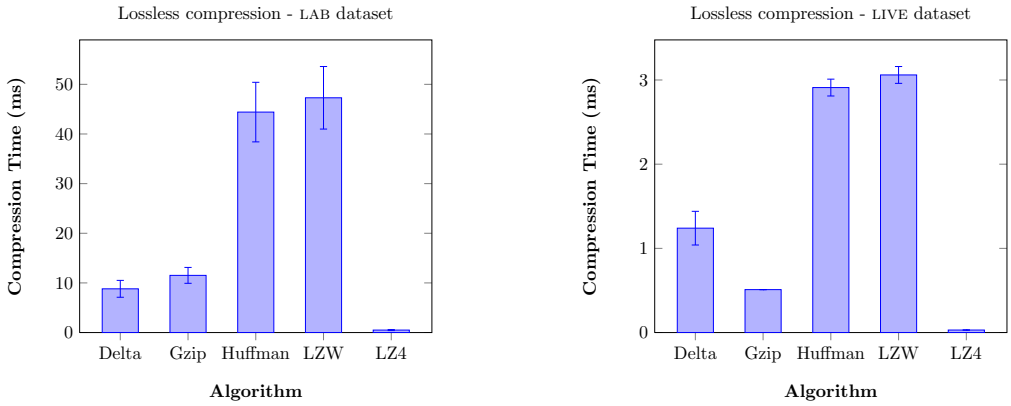

Figure A.2: Compression time for lossless algorithms (lower is better). Error bars denote 95% confidence intervals.

In a comparison of compression time (Figure A.2), LZ4 is shown to be faster than the other methods, with a payload as low as 0.5 ms on average for LAB data and 0.1 ms for LIVE data. Huffman and LZW encoding performed similarly, requiring at least twice the time consumed by Gzip and Delta compression. Gzip performed significantly faster than Delta compression in the LIVE data set.

A.3 Lossy Compression

We investigated five lossy compression techniques, shown visually in Figure A.3. Three were resampling techniques: piecewise linear interpolation (RSL) [72], non-linear interpolation (RSN) [62], and dispersion-threshold identification (IDT) [99]. Two were time-based polling techniques inspired by other researchers: sampling the cursor position after a fixed interval (time-based polling, TBP) [41] and sampling after a pause longer than a fixed amount of time (pause-based polling, PBP) [51].

These techniques operate by removing redundant and non-significant variations in the data. RSL is the simplest reconstruction method, sampling the points from a trajectory uniformly. RSN creates a resampled sequence by using piecewise linear interpolation on point distances accumulated along the original trajectory. IDT is a popular method to sample cursor data [3, 41, 73], and samples coordinates when the cursor moves away from

the previous point by a fixed number of pixels. TBP can be seen as an application of RSL in time instead of spatial dimensions. Similarly, PBP is a derivation of IDT using time rather than spatial constraints.

(a) RSL (b) RSN (c) IDT (d) TBP (e) PBP

Figure A.3: An illustration of a cursor trail compressed with different lossy compression algorithms in a thick red line, drawn alongside the original cursor trail in grey. The compression ratio for each algorithm is set to 50%, removing half of the original cursor coordinates.

For lossy compression algorithms, the compression ratio becomes a configurable independent variable in our experiments. We set the number of cursor points to be preserved as a percentage of the original data points, representing the option to choose an appropriate compression level for the application. We experimented with compression ratios ranging from 10% to 90%. A compression ratio of 10% meant that a cursor trail is reduced to 10% of the original cursor trail length. Therefore, the lower the compression ratio the lower the number of compressed data points. A good lossy compression algorithm achieves good performance and replication results at a low compression ratio.

The lossy compression algorithms had different input parameters to lead to the resulting compression ratio. Both RSL and RSN take as input the number of points to be sampled. IDT takes as input a threshold distance between consecutive cursor coordinates [99], which we chose to be the distance that returned the desired number of points according to the selected compression ratio. For TBP we set the poll interval proportional to the point percentages, which is also proportional to the compression ratio. Unfortunately, for PBP it is impossible to create a similar dependency, as there is no direct way to relate a pause threshold to a desired number of points to be sampled. A pause was counted when the

time between consecutive data points was greater than 250 ms, the same threshold used in Huang et al. [50]. Thus, PBP showed constant behavior for all tested compression ratios.

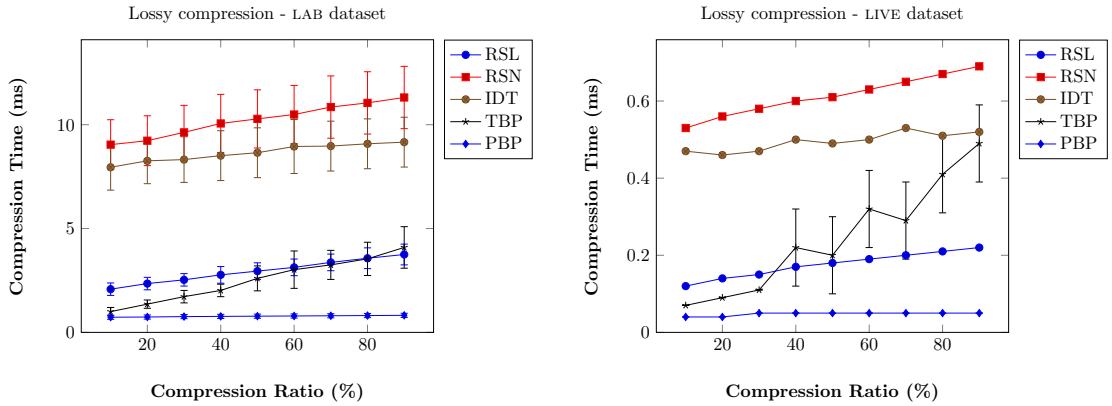

Figure A.4: Compression time for lossy algorithms (lower is better), as a function of compression ratio (higher ratios mean less reduction of cursor data points). Error bars denote 95% confidence intervals.

As expected, the lossy compression techniques except PBP (which used a fixed pause threshold and therefore a fixed compression ratio) performed better as the compression ratio increased. For the LIVE data, PBP performed significantly better than the other techniques considered. The other techniques (which were set to depend on the number of data points) reduced the data size linearly compared to the compression ratio. As a consequence of this linear dependence, a higher compression ratio requires more time to compress the data (Figure A.4), because more points are processed while sampling the original cursor trail.

Figure A.5 shows how well the compressed cursor trails reproduce the original cursor trail. Polling techniques were least able to produce the original trail using the metric from Equation A.1. Their poorer performance compared to other lossy compression techniques may be because the timestamps of compressed points in TBP and PBP are usually assigned to coordinates that are distant from the original cursor position. Therefore, while they achieve good compression ratios, they are less useful for replicating the original cursor data. However, the performance for TBP will change for different polling intervals.

The three resampling techniques had an approximate distance of 50 px when the data

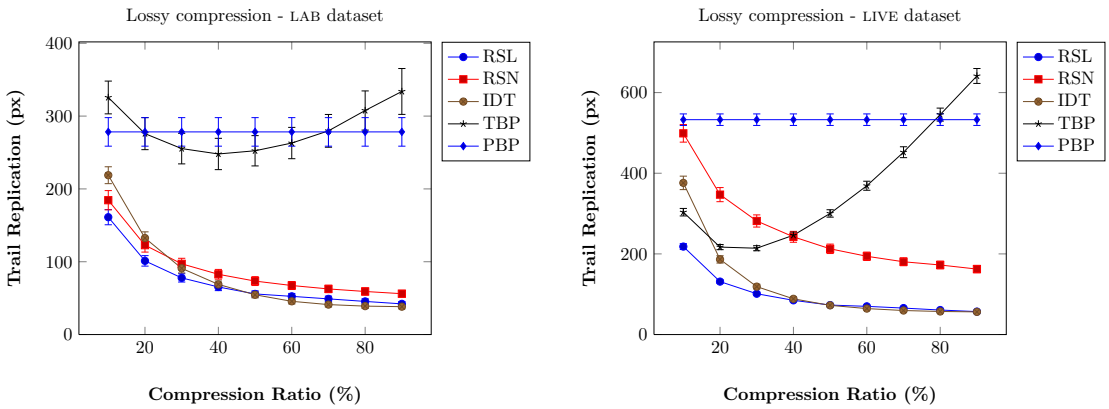

Figure A.5: Trail replication for lossy compression algorithms (lower is better), as a function of compression ratio (higher ratios mean less reduction of cursor data points). Error bars denote 95% confidence intervals.

was compressed by 60% or more in the LAB data. RSN had worse replication accuracy for the LIVE data, where RSL and IDT performed better. From this, we can see that both RSL and IDT seem to be better approaches for preserving the original cursor trail. For the LIVE data, RSL and IDT outperformed the others for compression ratios above 80%. These results suggest that resampling techniques may be a convenient approach for replaying cursor trails, when reproducing the original cursor data accurately is important.

When we look at the distance between compressed coordinates and original coordinates only at points preceding a click (Figure A.6), all lossy compression techniques except TBP performed well for compression ratios above 50% for the LAB dataset. When the data was compressed substantially, IDT significantly diverged and both RSL and PBP retained their ability to reproduce the original cursor data during clicks for compression ratios below 10%. On the LIVE data, resampling techniques performed reasonably well for compression ratios above 30%, with distances below 100 px. RSL had the shortest average distance, and thus is the best compression algorithm for applications where reducing data size and reproducing the original coordinates at clicks are both important goals. Overall, these experiments shows that lossy compression techniques usually preserve important information for web analytics.

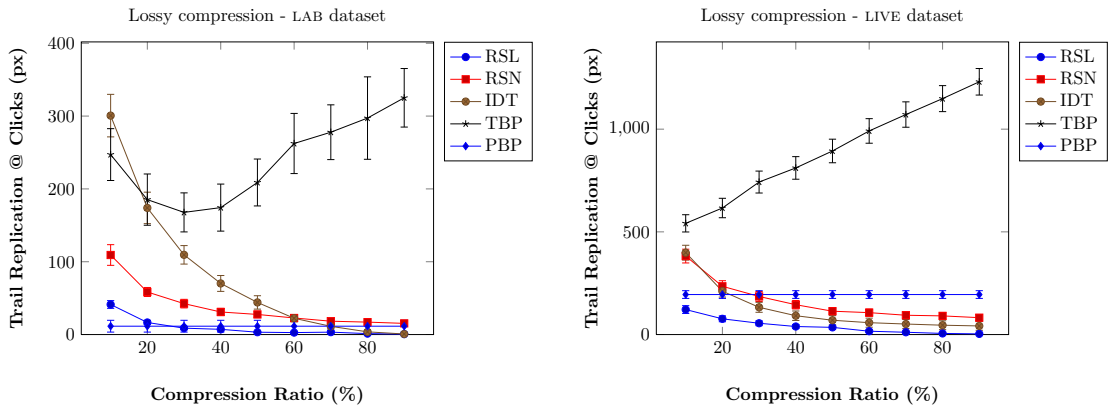

Figure A.6: Trail replication at clicks for lossy compression algorithms (lower is better), as a function of compression ratio (higher ratios mean less reduction of cursor data points). Error bars denote 95% confidence intervals.

A.4 Summary

We looked at 10 compression algorithms, half of them lossless and the other half lossy. Lossless compression algorithms are capable of reducing the data size while also reproducing the original cursor activity; we found that among lossless methods, LZW performed well. However, lossy compression can offer greater gains in compression as well as improved performance, at the expense of exactly replicating the original trail. For the same compression levels, the naive method of time-based polling replicated the original trail worse than recently-developed methods. Piecewise linear interpolation and dispersion-threshold identification could reproduce the original trail better at the same levels of compression. Pause-based polling (PBP), which was the method eventually used for remote cursor tracking in this dissertation, showed the best client-side performance and trail replication. Overall, the experiments evaluate multiple approaches to sampling cursor interactions, and teach us how to build a better mousetrap.

Appendix B

Gaze-Cursor Alignment in Subjects

Subject	Sex	Age	Inactive		Examining		Reading		Action		Click
			Distance	Time	Distance	Time	Distance	Time	Distance	Time	Distance
1	F	46	188 px	68%	116 px	27%	45 px	0%	46 px	4.8%	49 px
2	M	56	201 px	65%	149 px	28%	133 px	0%	79 px	5.9%	73 px
3	F	35	165 px	47%	144 px	47%	167 px	2%	82 px	4.9%	75 px
4	M	47	162 px	61%	218 px	26%	214 px	6%	113 px	7.2%	106 px
5	F	31	246 px	45%	139 px	42%	134 px	6%	76 px	6.7%	89 px
6	M	44	175 px	59%	163 px	32%	211 px	1%	108 px	7.4%	101 px
7	M	57	144 px	60%	100 px	29%	93 px	6%	76 px	5.9%	81 px
8	F	47	309 px	63%	224 px	30%	143 px	3%	58 px	3.7%	43 px
9	F	50	262 px	56%	150 px	31%	146 px	8%	89 px	5.7%	100 px
10	M	59	180 px	65%	165 px	29%	209 px	1%	139 px	4.6%	126 px
11	M	50	247 px	50%	198 px	36%	256 px	4%	106 px	9.9%	81 px
12	F	39	236 px	29%	140 px	55%	125 px	5%	74 px	10.7%	77 px
13	F	45	177 px	55%	155 px	33%	204 px	4%	95 px	7.3%	71 px
14	M	50	206 px	49%	195 px	41%	177 px	2%	91 px	7.4%	93 px
15	F	38	294 px	68%	202 px	23%	168 px	2%	129 px	6.7%	88 px
16	F	26	152 px	61%	131 px	27%	118 px	2%	98 px	8.8%	84 px
17	F	46	218 px	55%	152 px	36%	145 px	4%	69 px	5.1%	65 px
18	M	50	245 px	58%	244 px	34%	251 px	2%	92 px	6.4%	116 px
19	F	50	185 px	55%	120 px	38%	142 px	2%	58 px	5.1%	46 px
20	M	39	370 px	44%	219 px	47%	140 px	3%	97 px	6.1%	70 px
21	M	38	146 px	39%	164 px	49%	139 px	1%	69 px	10.1%	59 px
22	M	47	216 px	69%	160 px	24%	173 px	3%	92 px	4.4%	71 px
23	F	46	218 px	50%	190 px	42%	196 px	1%	76 px	6.5%	64 px
24	F	48	224 px	61%	178 px	34%	89 px	0%	64 px	4.7%	47 px
25	F	54	228 px	56%	184 px	36%	173 px	1%	39 px	6.8%	26 px
26	F	44	166 px	32%	167 px	58%	116 px	1%	78 px	8.1%	109 px
27	F	28	161 px	58%	142 px	33%	112 px	0%	90 px	8.3%	52 px
28	F	51	259 px	56%	195 px	34%	242 px	2%	76 px	7.5%	72 px
29	F	41	218 px	79%	171 px	15%	148 px	1%	89 px	5.7%	119 px
30	M	48	279 px	50%	138 px	37%	87 px	5%	42 px	7.7%	38 px
31	M	59	192 px	58%	175 px	34%	162 px	3%	152 px	4.5%	133 px
32	F	53	236 px	61%	199 px	35%	77 px	0%	52 px	3.3%	77 px
33	M	60	418 px	62%	305 px	27%	319 px	3%	69 px	8.0%	105 px
34	F	47	245 px	59%	214 px	36%	160 px	1%	44 px	4.2%	34 px
35	M	44	187 px	73%	133 px	22%	167 px	1%	68 px	4.7%	85 px
36	F	37	290 px	76%	231 px	20%	147 px	1%	58 px	2.9%	56 px

Table B.1: Basic demographic information for each subject, the proportion of time spent performing each cursor behavior, and the average distance between gaze and cursor while performing that cursor behavior. Clicks are instantaneous events with no duration.