

No Search Result Left Behind: Branching Behavior with Browser Tabs

Jeff Huang
University of Washington
Seattle, WA 98195
wsdm@jeffhuang.com

Thomas Lin
University of Washington
Seattle, WA 98195
tlin@cs.washington.edu

Ryen W. White
Microsoft Research
Redmond, WA 98052
ryenw@microsoft.com

ABSTRACT

Today's Web browsers allow users to open links in new windows or tabs. This action, which we call 'branching', is sometimes performed on search results when the user plans to eventually visit multiple results. We detect branching behavior on a large commercial search engine with a client-side script on the results page. Two-fifths of all users spawned new tabs on search results in the timeframe of our study; branching usage varied with different query types and vertical. Both branching and backtracking are viable methods for visiting multiple search results. To understand user search strategies, we treat multiple result clicks following a query as ordered events to understand user search strategies. Users branching in a query are more likely to click search results from top to bottom, while users who backtrack are less likely to do so; this is especially true for queries involving more than two clicks. These findings inform an experiment in which we take a popular click model and modify it to account for the differing user behavior when branching. By understanding that users continue examining search results before viewing a branched result, we can improve the click model for branching queries.

Categories and Subject Descriptors

H.5.4 [Information interfaces and presentation]: Hypertext/Hypermedia—Navigation

General Terms

Human Factors, Measurement, Experimentation, Design

Keywords

browser tabs, examining search results, click models

1. INTRODUCTION

Every Web browser of today supports tabbed browsing, allowing people to comfortably navigate multiple Web pages simultaneously. Browsing flow within a single tab or window

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.
Copyright 2012 ACM 978-1-4503-0747-5/12/02 ...\$10.00.

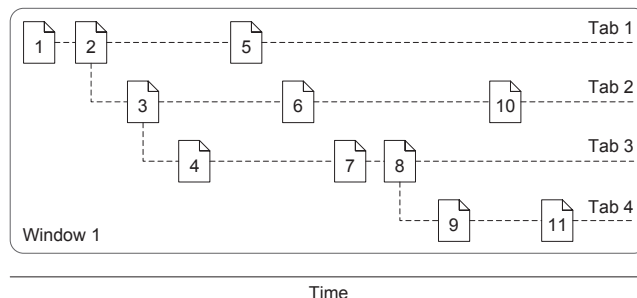


Figure 1: An example Web browsing session with multiple tabs over time. The user starts by viewing a single page and ultimately navigates to 11 pages across 4 browser tabs.

may be interrupted by switching to pages in other tabs and windows. This can be initiated by opening one or more pages from a single page; specifically, opening multiple links on a page in new windows or tabs. We call this 'branching'¹ since the user is spawning a new browsing thread and can continue browsing on either thread. Branching occurs when a user opens a link in a new tab or window, typically achieved with the right-click menu, a middle-mouse click, or by holding the CTRL key before clicking a link. Figure 1 shows the transitions between up to four tabs within a single browser window; the transition from page 2 to page 3 is an example of branching.

Branching can substitute for a classic browser feature—the Back button—which enables backtracking. Rather than sequentially viewing the interesting links on a page by clicking then tapping the Back button to return to the original page, users can simply open all the attractive links on a page in tabs to view later one by one. There are multiple benefits to this approach: opened pages will pre-load, shortening the wait time between viewing pages; there is no risk of being unable to return to the originating page (due to a Web page disabling the Back button or the originating page being inaccessible); and the user may spend as much time on the originating page as desired without having to remember all the links they were planning to view. Indeed, Weinreich et al. [22] note "the usage of multiple windows and tabs has

¹Different authors have used alternate terminology: Chierichetti et al. [3] call this 'control-click' since this action can be performed by holding the CTRL key and clicking; the word 'branching' in Meiss et al. [15] also includes Back button usage (which we distinguish as *backtracking*); Huang and White [10] use the same terminology that we use here.

partly replaced back button usage”.

This behavior affects the analysis of Web search, in which a holistic understanding of search interfaces and browser usability is important. In interviews with users who frequently use tabs in the Firefox browser [5], one user commented, “*If I search for something on Google, I just go ‘right-click, right-click, right-click’—you know, opening all the tabs. And then I would look at them later.*” Another explained, “*most links that I click [...] like when I do a Google search, I hold on ctrl and click on links.*”

On the search engine results page (SERP), browsing behavior is deliberate rather than forced by popups or links that always open in new windows. When studying branching outside of the SERP, these popups and ‘forced branches’ are difficult to differentiate from user initiated branches, which make the findings less meaningful. Branching in Web search is inadequately studied, partly because capturing the required data at scale requires additional effort; they are not captured in typical query logs [11]. However, the prevalence of branching as a new search strategy has prompted our study of branching behavior on search results.

Since Web servers are agnostic to which tab or window is active, they only record clicks and queries in the order they were received. Therefore, switching to a different page to click or opening multiple tabs from a single search result page confuses algorithms that process query logs [19]. Essentially, there are two problems: 1) the challenge of understanding interleaved sessions and 2) loss of detail about branching behavior (when users open links in new tabs or windows). Also, branching has not been captured in click models (e.g., [2, 6, 7, 20]) that treat user clicks as implicit relevance feedback; thus, we evaluate the effectiveness of adapting a click model that understands branching.

Our guiding research questions are 1) which users are branching, when they do it, and to what extent does branching occur on search results? 2) How does branching compare to backtracking in user search strategies, specifically in queries with multiple result clicks? 3) Can we modify an existing click model that predicts clicks on SERPs based on a better understanding of branching behavior?

In our paper, we explore these questions to gain insight about searcher activity, unpack branching behavior as a key interaction in search, and—enabled by better understanding of branching—propose modifications to a click model.

2. RELATED WORK

Several qualitative studies have reported on the use of multiple windows and tabs in Web search. Thatcher et al. [18] observed users opening multiple browser windows to simultaneously conduct multiple searches. A survey of experienced Web users by Aula et al. [1] found that multiple windows and tabs were used to support backtracking or multitasking while one page was loading. To backtrack, they observed users branching and eventually switching back to the originating tab rather than using the Back button in the browser. For multitasking, they observed users interacting with another tab while waiting for the original one to load, i.e., tab switching. More recently, a Web navigation study by Weinreich et al. [21] found that their participants frequently used multiple windows, enabling them to compare search results side-by-side, load pages in the background while they continued browsing, and retain important pages in their sessions for backtracking. Weinreich et al. also found that the

participants who used tabs backtracked less than those who did not. Similarly, those who created more windows tended to backtrack less. Dubroy and Balakrishnan [5] conducted a study on 21 users of the Firefox browser to investigate why tabs were popular when similar functionality is available by using multiple windows. They found that users preferred tabs because they are less cluttered and easier to switch between. Dubroy and Balakrishnan also looked at revisitation and their findings agree with Weinreich et al. that tab usage decreased use of other revisitation methods, especially the Back button. We extend this work by studying *when* and *how often* this behavior occurs in Web search.

Human editors from Jones and Klinkner’s study [13] found that 17% of search tasks are interleaved, potentially from users switching between tabs when browsing. Spink et al. [17] discussed multitasking in Web search, which they defined as searching for multiple topics in a single session. They concluded that users frequently multitasked searches, according to this definition. However, parallel browsing, a user’s use of simultaneous tab sessions [10], better resembles the classical definition of multitasking. MacKay and Watters [14] find in their diary and field studies that certain tasks required multiple sessions to complete. From a cognitive psychology perspective, Miyata and Norman [16] explained that humans have foreground and background tasks: they focus on the former and do not give conscious supervision to the latter. Parallel browsing follows this model because the user does not interact with multiple tabs simultaneously. Instead, the active tab is the foreground task and has the user’s attention while other tabs may be loading in the background or contain information that is not yet needed. Indeed, Zhang and Zhao [23] found a significant difference between revisitation rates measured by pageloads, compared to revisitation rates measured by tab focus.

Viermetz et al. [19] studied the effect of tabbed browsing on the validity of search log studies. They analyzed query logs which did not directly contain information about tabs or windows since they only recorded incoming network traffic. Using this data, they constructed a ‘clicktree’ of possible paths from the pageviews in the logs. They found that parallel browsing occurred 4–85% of the time. In contrast, our search logs directly contained information about branching activity involving tabs on the search engine results page.

Most related to our work is a study of parallel browsing behavior on the Web conducted by Huang and White [10]. They performed a preliminary study of tabbed browsing on logs gathered through a browser plug-in in June 2009. One part of this study included branching in search results, which we believe to be the first large-scale study of branching. They found that while a small percentage of users branched frequently, the majority always clicked results directly. These users only branched 6–15% of the time for queries that were the most likely candidates for branching: non-navigational queries. In this paper, we extend the exploration of branching by delving deeper. Our study examines branching behavior using a pageload event instrumented on the search results page, whereas the previous study by Huang and White used a browser-specific plug-in. We also examine queries involving multiple clicks in order to compare branching with backtracking, which informs our experiment on accounting for branching in a click model.

Some work has improved Web browsing models by incorporating tabbed browsing data. Chierichetti et al. [3] de-

Sequence	Percent	Notes
pageload click	57.4%	click
pageload	19.7%	no clicks
pageload click pageload	5.1%	click, back
pageload click pageload click	5.0%	click, back, click
	3.9%	events lost
pageload click pageload click pageload click	1.4%	3 clicks, back each time
pageload click pageload click pageload	1.3%	click, back, click, back
pageload click click	0.7%	branch or double click
click pageload	0.7%	delayed javascript
pageload pageload	0.7%	page refresh without click

Table 1: Top 10 most common recorded event sequences

Expression	Percent	Notes
$\hat{\text{pageload}}$	95.2%	sequence begins with pageload
$\hat{\text{click}}$	0.93%	sequence begins with click
$\hat{\text{click}} \cap (\text{pageload})$	0.84%	click recorded before any pageload, events out of sync
$\hat{\text{click}} \cap \neg (\text{pageload})$	0.1%	see click but not pageload (i.e., JavaScript off)
(click click)	1.5%	has branching or double click
$\hat{\text{pageload}}.*\text{click}.*\text{click}.*$	11.7%	JavaScript on, in sync, ≥ 2 clicks
$\hat{\text{pageload}} \cap \text{click}\$$	66.7%	begins with pageload, ends with click
$(\text{click pageload}) \cap (\text{click click})$	0.5%	(branch or double click), backtrack on same SERP

Table 2: Some interesting regular expressions on click sequences

veloped a new model of browsing based off Pagerank, but allowed tabs to be spawned from a page. Unfortunately, the toolbar logs they used could not distinguish between branching and backtracking, but they found that adding branching and backtracking data made the model better at predicting user browsing data. Meiss et al. [15] incorporated bookmarks, Back button usage, and tabbed browsing to develop a model that predicted user entropy, session size, and session depth better than the BookRank baseline. Our work focuses on the search domain and understanding branching behavior on the SERP, applying the findings to click models.

3. METHOD

Our method is based around collecting search logs with the necessary data for analyzing branching behavior. Studying branching behavior at scale through logs allows us to make generalizations and observe uncommon phenomena, which may not surface in small-scale studies. Search logs possess statistical power in the diversity of tasks and users, and realism which is particularly important for studying search strategies because artificial tasks and environments may alter how users search. But they cannot explain *why* a user is performing an action, or understand the user’s thought process as they search; the data may be noisy, containing interactions from spammers or bots.

Since data from tabbed browsing interactions are not typically recorded, we instrument a client-side script that records user behavior on the search results page. The script allows us to study branching at scale, a user behavior that has not received attention corresponding to its impact. Our logs span 3 weeks of searches from the Bing search engine (from June 10, 2011 to June 30, 2011), comprising standard query logs supplemented by interaction logs consisting of pageload events and page information. The searches took

place in the Web vertical of the search engine unless otherwise specified. We excluded the zh-CN (China) region in our study, since the branching behavior there was completely different from other regions, as we find in Section 5.1. Collecting data through the search results page makes our findings browser agnostic, differing from a previous study on branching [10]; however, this methodology prevents us from knowing whether the user branched in a new *tab* or new *window*, which slightly differ in function. In total, our data comprised over 3 billion query sessions from an estimated 390 million searchers on the first page of search results.

We develop a scalable approach to studying branching behavior that enables us to analyze real user activity in a natural setting. A small piece of JavaScript code notifies the server whenever the SERP is loaded, and the SERP’s initial load sends a pageload event. When a user backtracks by clicking and using the Back button, a second pageload event is sent to the server. But when a user branches by opening links in new tabs or windows, the page is not reloaded and no pageload event is sent. In other words, when a user branches on the SERP, there is only one pageload event followed by clicks representing branches; when a user backtracks on the SERP, there is one pageload event preceding each click. This allows us to distinguish between branching and backtracking.

Errors in logging may occur, as pageload and click events may be lost due to problems with the user’s computer or network connection or with JavaScript being unavailable. We noticed that 0.84% of queries had events that were obviously out of sync (a click before the first pageload); we discard these because they are temporally illogical and because given their rarity, we believe that removing them does not significantly alter the results. Over 96% of queries had a pageload event sent by the JavaScript, but only 0.1% of query sessions had a click and no pageload.

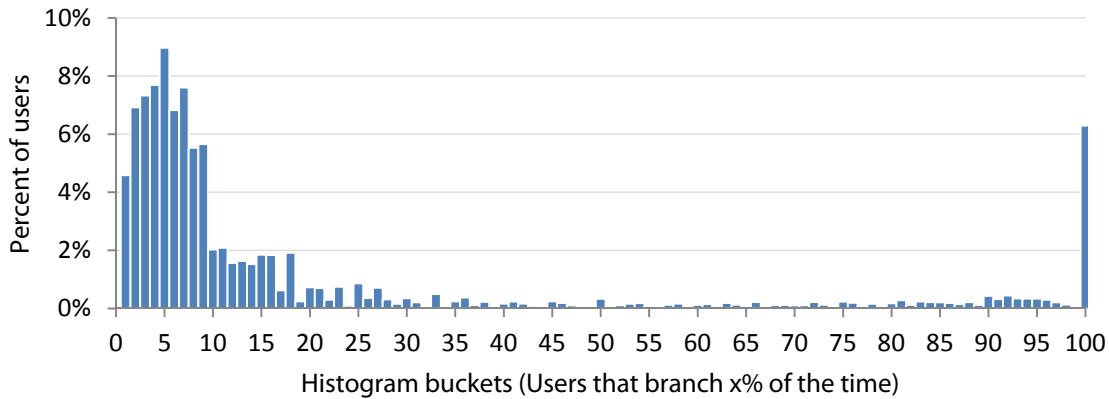


Figure 2: On Web search results, the users who exhibited any branching behavior at all tended to do so between 1–9% of the time that they clicked two results in the same page, with a peak at 5%. Over 6% of these users branched on every query session involving multiple clicks.

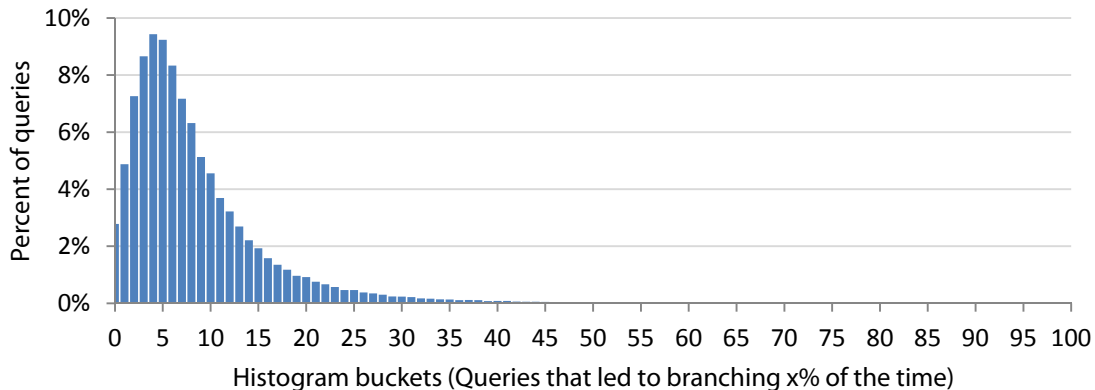


Figure 3: 56.4% of the queries (that ≥ 50 users issued and clicked multiple results) led to branching between 2–9% of the time (with a peak at 4–5%). Queries that had no branching or $\geq 40\%$ branching were rare.

4. CHARACTERISTICS OF BRANCHING

In this section, we examine some basic characteristics of branching behavior in order to understand how frequently users branch and the general characteristics of branching.

Table 1 shows the most common pageload and click sequences that we observed. 57.4% of queries have one click and end, 19.7% have one pageload and end, and 3.9% have missing events. ‘pageload click’ occurred more often than ‘pageload’, indicating that people are more likely to click one link than none. ‘pageload, click, pageload, click’ (2 clicks via backtracking) occurred 5% of the time, while ‘pageload, click, click’ (2 clicks via branching) occurred 0.7% of the time. Backtracking remains more popular than branching on the SERP.

Table 2 presents the event sequences matching several hand-picked regular expressions. Only 0.93% of the data started with ‘click’; a sequence that contains a ‘click’ event but no ‘pageload’ event happens when that JavaScript is disabled or not supported by the browser. This sequence occurred in 0.09% of query sessions, and had no clear association with the geographic region. Overall, 12.9% of queries with multiple clicks could potentially involve branching. However, some of these were actually double clicks, wherein the user clicks the same link twice before the browser navigates to the next Web page. We filter out double clicks

hereafter by discarding consecutive clicks on the same result link. After removing double clicks, we find that branching occurs in 9.1% of each consecutive pair of clicks in the same query sessions.

One surprise was that queries involving both branching and backtracking were quite common. For queries with at least 3 clicks, there were almost as many queries where a user branched and backtracked in the same session, as there were queries where a user only branched. For example, a user may click a link and be unsatisfied, return to the search results page using the Back button, then open 3 other links in new tabs. We mostly observed users branching after backtracking, indicating that perhaps the first clicked link was not what they expected. This represents a shift in the user’s strategy as they are clicking on search results; perhaps after realizing their first click was unsatisfying, they branched multiple results to retrieve more options.

4.1 Branching Users

Is branching caused by particular users being more prone to branch due to personal preference, or by certain classes of queries being more amenable to branching behavior (e.g., for certain classes of queries, people will know beforehand that they want to read multiple pages)? Or perhaps a particular Web browser always causes branching activity. To pin down

the causes of branching behavior, we analyzed the number of users who never branch, always branch, or branch 1%, 2%, ..., 99% of the time. To avoid clusters at values like 1, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, and $\frac{1}{5}$ due to users with few queries in our dataset, we only included users who clicked on multiple search results on the SERP at least 10 times over the 3 weeks. We also only included click sequences beginning with a ‘pageload’ event to remove users without JavaScript.

When visiting two results in the same SERP, we found that 60.9% of the users always used backtracking and 2.5% of the users always used branching. Figure 2 shows the distribution of users who branched by how often they branched. Because this graph excludes all the users who never branched, the proportion that always branched increases from 2.5% to 6.3%. The United States distribution closely resembles the distribution of all non-China regions combined. It is possible that the users who always exhibited branching behavior strongly preferred branching or were using some particular software configuration that always caused branching.

The data indicates that branching is not always performed by frequently branching users, but from those who occasionally use it (e.g., 3–7% of the time). This suggests that certain queries or circumstances besides user habits increase the likelihood of user branching.

4.2 Branching Queries

To further test this hypothesis, we analyzed all the queries in our log to observe the specific queries from which users tended to branch. To avoid artifacts caused by data sparseness, we restricted the set to only queries issued by at least 50 distinct users (who then clicked two search results on the first page of results). The histogram in Figure 3 illustrates that the most common amount of branching for a query was between 2–9% of the time (with a peak at 4–5%). Queries rarely had either no branching or over 40% of the query sessions branched.

Out of the 91,545 distinct queries that matched our filtering criteria, the specific query with the highest proportion of branching was “no strings attached torrent”. In our log, 121 users issued this query and then clicked on two or more results in the results page. For results loaded beyond the first, 258 were loaded via branching and 25 were loaded via backtracking; for a branching ratio of 91%. “No Strings Attached” is a film; users who query for the torrent are looking to illegally download it. However, each of the top results is a page where the user has some small chance of successfully downloading the movie. It seems logical that the large majority of users in this scenario will branch to open many of these links at once rather than checking them one by one to see if they work. Of the 69 queries that contained the word “torrent”, the average branching-to-backtracking ratio was 51%.

Another indicative pattern of branching was the phrase “file extension”. For example, a query for “file extension trp” means the user is wondering what the “trp” extension means and is likely to read through multiple results to learn more. Of the 399 queries that contained “file extension”, the average branching ratio was 22.5%. Other examples of indicative words are “lottery” (66 queries, average branch ratio of 15.6%) and “Facebook” (104 queries, average branch ratio of 14.9%). Queries containing “lottery” generally come from people looking to check their lottery numbers; multiple candidate sites could post the numbers, and the user may

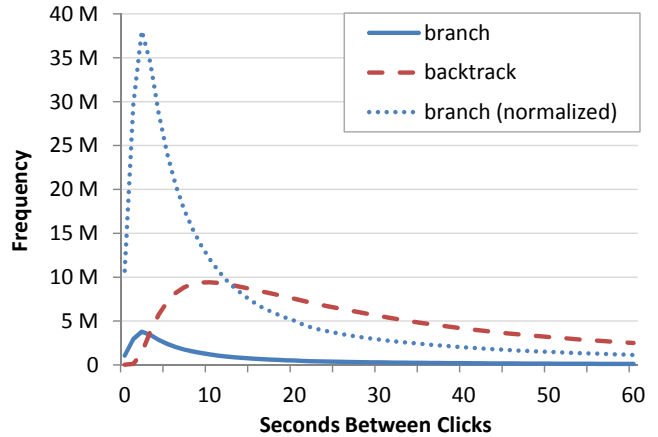


Figure 4: When users branch, the time between successive result clicks is significantly lower than when they backtrack.

not be sure which of those sites has the latest numbers, so they open multiple results. Queries containing “Facebook” are from users looking for their friends but the search engine returns Facebook pages for several people sharing that same name. The user will then open several of these result pages and examine them to find the right person.

The primary word that correlated to low branching was “trailer”. Users searching for movie trailers are unlikely to open multiple results in tabs because each result is likely to play the trailer video, and the audio from multiple videos playing at once would overlap. Of the 54 queries containing “trailer”, the average branching ratio was 3.9%. Most other words did not correlate with low branching by themselves. Low branching tended to be specific to the individual queries where the search engine returned a top result that clearly met the users’ needs so there was no reason for the user to branch multiple results.

5. CLICK INTERVALS

We expect to see different lengths of time between clicks if the underlying user behavior is different in branching and backtracking. Figure 4 examines the Web vertical (for all regions) and displays the frequency of branching (solid line) compared to backtracking (dashed line) in situations in which the user clicked on two links in the same results page. The x-axis indicates the time elapsed between the two clicks. Branching comprises a smaller (but still substantial) proportion of the clicks compared to backtracking, as illustrated by backtracking line covering a much greater area than the branching line. By total frequency, when a user clicked on two links in the same results page, 9.1% of the clicks came from branching activity (over the 3 week period, Web vertical, excluding China). If data from China is included, then 13.4% of all clicks came from branching activity.

The dotted line in Figure 4 normalizes the branching values to have the same total frequency as backtracking, in order to better distinguish the different peak time between clicks. For branching, the peak time between clicks was 2–3 seconds. For backtracking, the peak time between clicks was 10–11 seconds. The time between clicks fell under 30 seconds about 69% of the time with branching but only 41% of the

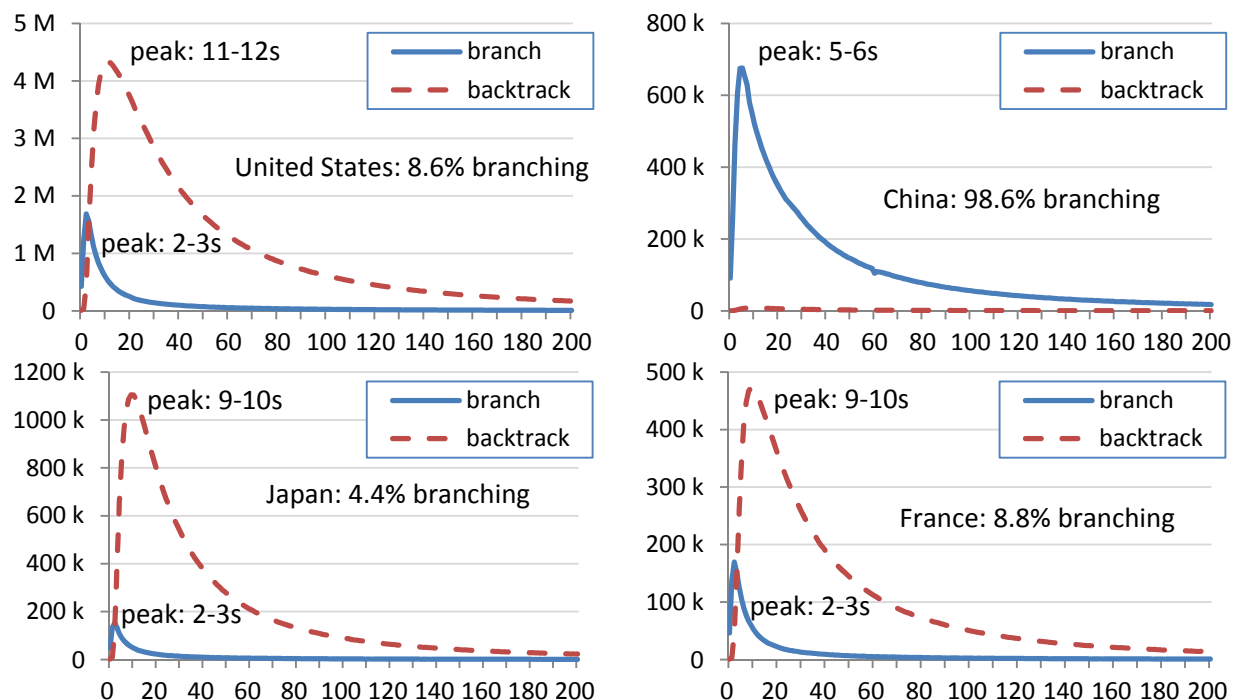


Figure 5: Seconds Between Clicks vs Frequency graph for: United States Web vertical (top left), China Web (top right), Japan Web (bottom left), France Web (bottom right). Most regions had similar branching frequencies, except China, where branching behavior is the default action.

time with backtracking. The time between clicks is sometimes used to judge user satisfaction with the initial clicked results (e.g., [8, 9]). Our findings indicate that branching activity needs to be filtered out or otherwise accounted for with this metric in order to avoid unfair penalization of branched results.

The normalized branching line illustrates that branching had a much sharper and earlier peak, indicating that compared to backtracking, branching has a much larger proportion of clicks that happen quickly. Perhaps when users branch, they often continue looking for more search results to click rather than reading the page that they opened in the new tab before coming back and selecting more links.

5.1 Regions

When we separate the Web vertical branching data by geographic region, we find that United States (8.6% branching) and France (8.8% branching) are similar. However, some regions are outliers: Japan only has 4.4% branching and China has 98.6% branching.

Japan is interesting in that branching occurs less frequently. Also, as shown in Figure 5, the branching distribution is noticeably less sharp around the 2–3 second spike than in other markets. This suggests that people are less likely to branch, and when they do, they are more likely to read the opened page before selecting another.

In China, the default action for popular search engines is to branch when users click on search results. As a result, 98.6% of our data from the Chinese market demonstrated branching behavior. The remaining 1.4% may have been due to users who found a way to disable this behavior, or people using browsers that do not support opening multiple

windows. The branching distribution for China has a sharp curve, but not as sharp as most other branching curves. Most markets have branching distributions that peak in the 2–3 second range, but China’s peak is at 4–5 seconds. Without backtracking, users may prefer taking the time to look at the open result in the new tab before returning to the results page to open another result.

The ubiquity of branching in China suggests that branching may become more significant as the Internet user base in China increases, and for users accustomed to this habit in China that migrate to other countries. However, to avoid confounding the data with other factors that may differ in Chinese Web search, most of our comparisons and analyses in this paper exclude data from China.

5.2 Verticals

We found that the News and Web verticals had similar branching rates. Intuitively, there are news topics for which people will want to seek out as much information as possible, which may mean reading multiple articles. However, news may not possess an equivalent to queries like “torrent” and “Facebook” where the user knows ahead of time that they need to screen multiple candidate Web results to find what they want. In terms of time between clicks, we found that when users backtrack with news, they spend more time on the page (17–18 second peak) before clicking Back. When users branch with news, the time still peaks at 2–3 seconds. These times support our hypothesis that users read the clicked page before clicking Back, but not before branching. Users are likely to spend more time reading a news article (compared to a general Web result) before deciding that they want another.

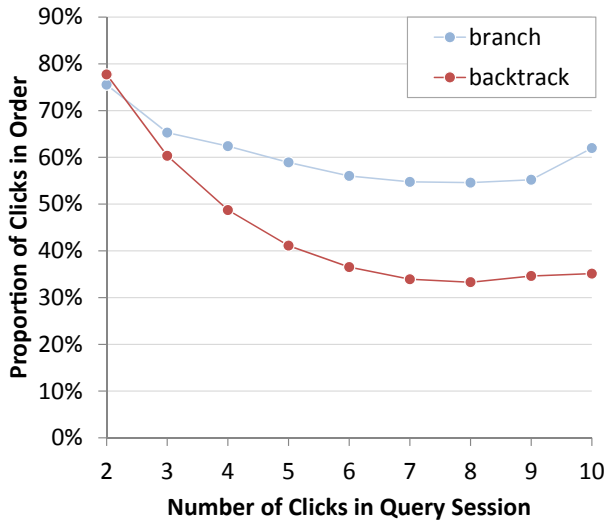


Figure 6: The effect of branching vs. backtracking on click order for multi-click queries, controlling for user. Users are more likely to click search results from top to bottom when they are branching.

We also looked at branching on the Images and Videos verticals, but our method for detecting branching was invalid in these verticals due to interface differences compared to the Web and News verticals.

6. LINEAR TRAVERSAL HYPOTHESIS

Queries with multiple clicks are interesting because they represent cases in which the user is not finished after selecting one link. In these situations, the user may have not been sufficiently satisfied, or was interested in viewing more diverse sources of information. In our data, 11.7% of queries with JavaScript enabled have at least 2 clicks.

Users can open multiple links using branching, as we have seen, they usually use the Back button after viewing each link. Here we investigate the examination pattern for the two behaviors: how often do users satisfy the *linear traversal hypothesis* [12] in each case? We want to revisit this common assumption through the lens of multiple clicks. This hypothesis suggests that users review search results from top to bottom and do not return to any previously viewed result after they have skipped it. It has been a basic assumption in several search interaction models [2, 4, 7] and is intuitive since search results are sorted in order of relevance. However, whether this hypothesis is true is a particularly tricky question to answer because it can be confounded by the effects of user style and query type and because examination behavior is difficult to observe at scale. We analyze a related question: are users *clicking* search results from top to bottom?

From this point we will refer to click sequences satisfying the linear traversal hypothesis as LTH and denote click sequences that contradict the linear traversal hypothesis as \neg LTH. While the linear traversal hypothesis does not directly tell us how users are *examining* search results, a LTH click sequence says that either users are examining search results from top to bottom, or that they are returning to a previously examined search result to click.

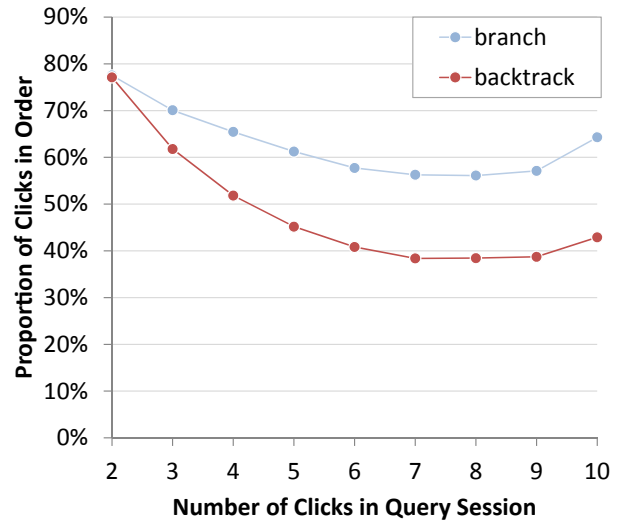


Figure 7: The effect of branching vs. backtracking on click order for multi-click queries, controlling for query. Users are more likely to click search results from top to bottom when they are branching.

The search models cited above cannot account for \neg LTH behavior; Chapelle and Zhang remark, “*Sessions for which the clicks are not in the same order as the ranking have zero probability under our model. On average, roughly 3% of the sessions contain one or more out-of-the-order clicks. We could have decided to swap the click order for these sessions, but we simply discarded them.*” However, most queries have only one click—one click cannot be out of order. When we look at queries featuring two or more clicks, 30% are \neg LTH. Moreover, for queries featuring three or more clicks, 44% are \neg LTH. Thus, while Chapelle and Zhang have found that only 3% of queries have \neg LTH clicks, this number is significantly higher for queries with multiple clicks. We believe these query sessions with \neg LTH clicks may encapsulate information that should not be discarded.

To deliver meaningful results, we need to control for the *user* effect, as differences in click order in branching and backtracking behavior may result from those users having a particular style of clicking. Additionally, the uneven distribution of the number of clicks by each user may skew the results. To control for these factors, we only select query sessions from users who have engaged in both branching and backtracking; additionally, for each user we only sample one query session that involves branching and one query session that involves backtracking. This method ensures that we know whether the LTH clicks are related to whether they are branched, not because of user differences.

We control for the potential effect of a query on multi-click behavior in the same way as for a user, by replacing ‘user’ with ‘query’ in the above steps. Similarly, we know that query differences are not why the clicks are LTH. Thus, our analysis distinguishes between the effect of branching or backtracking behavior on click order in queries with multiple clicks. Note that controlling for both user and query at the same time (the interaction effect) in the same manner interferes with user behavior: a user that issues the same query twice may have different click behavior the second time simply because they already issued the same query.

Figures 6 and 7 both show that \neg LTH is more common when there are more clicks. This is intuitive considering there are more permutations for the click order when there are more clicks. More interestingly, when branching, users are more likely to click as LTH when there are more than two clicks, even when controlling for user or query. LTH clicks remain at around 54–61% of occurrences when branching with 5 or more clicks. However, for queries with exactly two clicks, there is little difference in click order between branching and backtracking; in fact, when controlling for user, users who are backtracking are slightly more likely to do LTH clicks.

7. BRANCHING IN CLICK MODELS

The data that describes the user’s branching behavior can also be used to better model their search result examination behavior. Several click models (e.g., [2, 6, 7, 20]) use hidden states to represent whether a user continued examining search results after clicking. Having accurate information about a user’s focus can improve these models. Additionally, knowing whether a user opened a search result in new tabs or windows (i.e., branching) can provide a different signal than if they simply clicked and returned (i.e., backtracking). For example, branching may cause users to click results in a sequential top-to-bottom order, opening any attractive result, but backtracking may entail a different strategy of finding the most potentially satisfying search result.

Here we will describe a click model that we will modify using our understanding of user branching behavior. We will describe the Perplexity metric used to evaluate the model. Finally, we will report the experiment and results.

7.1 Click Model

We will replicate the Dynamic Bayesian Network (DBN) model [2], the most cited click model since the Cascade Model (which compared favorably to all models before it). The DBN model compares favorably to the Cascade Model [2, 24], and fares well compared to other models (e.g., [24, 25]), making it a solid baseline for our purposes.

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

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

In this model, a user examines search results from top to bottom, assessing each result for whether or not it is attractive enough to click, a decision that depends only on the attractiveness of the link a_u . 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.

We implement the DBN model with $\lambda = 1$, labeled *Algorithm 1* in [2], to simplify the inference of latent variables.

7.2 Click Perplexity

Though we cannot evaluate unobserved events in a click model such as which results the user examined, we can test how well it predicts clicks, the observable events. Several click model studies [6, 7, 24, 25] have evaluated click perplexity as a measure of predicting click-through rates. In those studies, query sessions were divided evenly into training and test sets, each comprising at least 5 query sessions with clicks on search results. The click perplexity quantifies how much the test data surprises the trained model; it is computed for each combination of query and position as,

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

where p_i is the perplexity in the i th position, N the number of links, and q_i^n the predicted click probability for the n th query session. The exponent represents the cross-entropy estimated from a probability distribution. If the perplexity is 1, the trained model perfectly predicted the test data. A lower perplexity means the model was more accurate in predicting the test data. Because the lower bound of the perplexity depends on the click-through rate of the query, the perplexity varies substantially depending on the position of the search result. Therefore, we computed a separate perplexity value for every search result position and report perplexity values for each of the top ten rank positions.

7.3 Experiment

Zhang and Zhao remarked that a page loading from a clicked link does not necessarily mean the user has examined that page [23], since it may have opened in a background tab. Thus, a user branching on a link does not determine at that time whether that link has satisfied their search need. Our experiment exploits the following insight: when users branch, they are likely to click multiple search results even if one of the clicked search results is satisfying, i.e.,

$$P(E_{i+1}|E_i = 1, C_i, S_i) = P(E_{i+1}|P(E_i)) = \gamma$$

We use γ as the probability that the user stops examining search results at each point. Once this happens, the user will go through the opened links. Our tests showed that $\gamma = 0.9$ performed best, indicating that at each search result snippet or a 10% chance of going to review the opened tabs. For the original model, we used new s_u priors for these branching queries (which helps the original model) since the probability of continuing to examine search results is higher than across all queries.

We conduct an experiment by running this ‘branching model’ alongside the original model for comparison. As in prior work, query sessions were divided evenly into training and test sets, each comprising at least 5 query sessions with clicks on search results. We computed the perplexity for each position and averaged the scores across a total of 1,178,176 unique queries. Figure 8 shows the perplexity for the two models across different positions. The original model poorly predicts click-through rates for query sessions

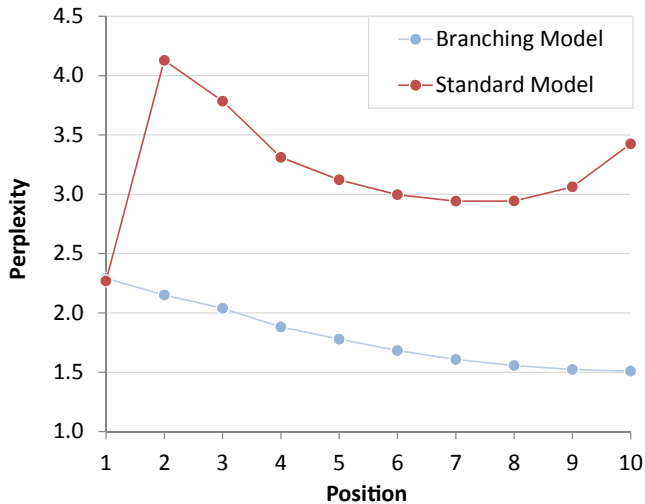


Figure 8: Perplexity for the standard DBN model with $\lambda = 1$ computed for queries where the user branched multiple clicks (lower Perplexity is better). The branching model (where users are assumed to branch all the links before examining them) compared favorably to the standard model.

involving branching, while the branching model performs better.

For positions 2–10, the original performs worse than for position 1; this is because the original model assumes the user will stop if the first result is satisfying. We believe this assumption is not true when the user is branching because they do not review the opened link until after all the clicks. Therefore, it is hard to predict multiple clicks in branching since there are more unexpected clicks on lower-ranked search results. The Perplexity will generally be high because clicks on lower-ranked results are unexpected, but this occurs frequently in queries featuring branching by definition. In practice, the branching model can be used to improve the overall click model, since the search engine can apply the branching model whenever it determines using pageload events that a user has branched.

8. DISCUSSION

Tabbed browsing is popular and has changed peoples’ search behaviors by allowing them to simultaneously load multiple search results to view later. We know that users who open links in tabs behave differently based on evidence that they are more likely to click search results from top to bottom. This has an impact on click models, which should be careful about how they process query logs. It would be prudent for click models to consider that users may not be going from top to bottom, as many models assume, especially when users are backtracking.

Recent work in information retrieval has used the time between clicks as a feature to model user behavior [8, 9]. We find that when users branch, this duration is often 2–3 seconds, even across different regions and verticals. This is shorter than the approximately 10 seconds between clicks when users are backtracking. The peak also has a noticeably smaller variance when users branch. Since we would imagine that the reading time for different Web pages varies, a con-

sistent click interval suggests that users are reading the page when they backtrack but not when they branch. This evidence suggests that the ‘time between clicks’ feature is not useful for determining the relevance of the clicked link when branching; it instead represents the time spent examining each search result to determine whether to branch. The anomaly in the Chinese region in which links are branched by default presents additional evidence. There, the most common dwell time is 5–6 seconds instead of 2–3 seconds, confirming our belief that users do not review the opened link when they choose to branch, but will sometimes review the opened link when they are forced to branch. Our findings show that the time between clicks is only a useful feature when the user is backtracking.

Our study differs from a previous one [10] because it infers the branch or backtrack action by checking for pageload events between clicks, rather than through a Web browser plug-in. While we are able to capture behavior across all Web browsers rather than just one, we are thus limited to a single search engine. It is possible that user interaction behavior may differ across search engines, or users of different search engines possess different branching patterns. Furthermore, we do not collect data about tab switching, which can track the user’s focus in the browser. Data on anomalies due to different search engines or tab switching can be measured with a different data collection method.

Although we attempt to detect instances of branching and backtracking when possible, this is sometimes impossible given the available data. When the last recorded event is a click in a query session, we cannot tell whether this is a branch or regular click. One fewer click is required to detect backtracking since it is defined as a click followed by a pageload, whereas two consecutive clicks are required to detect branching. Finally, though we believe manual Reload actions are uncommon in the middle of a query session, they can send ‘pageload’ events, leading us to incorrectly classify a branch as a backtrack. Overall, our technique for distinguishing branching from backtracking may have overcounted backtracking, resulting in estimates of branching more conservative than the true proportion. However, the fraction of clicks that were branched did align closely with those reported in previous work on a browser-based data set [10].

Users in the Chinese market nearly always branch, a behavior forced by their search engines’ default settings. Some people speculate² that this default started because Internet access is slower in China and so branching enables Web pages to load in the background. Others believe default branching is economically motivated as it allows a search engine to keep users on its site longer. However, a qualitative study of how Chinese users perceive branching versus backtracking would be interesting since their default action is opposite of that for the American and European users, who have been previously studied [1, 5, 18, 22].

Because branching queries have shorter time intervals between clicks (2–3 seconds compared to 5–30 seconds when backtracking), and *because* their clicks more often move from top to bottom, branching compels a different examination strategy. The improved branching model provides further evidence of our belief that users are not examining the search result landing pages following every click when they branch.

²<http://www.quora.com/China/Why-do-most-hyperlinks-on-Chinese-websites-open-a-new-window-tab>

9. CONCLUSIONS

Branching is a core feature of tabbed browsing. We develop a technique to differentiate between branching and backtracking, i.e., when users open links in new tabs versus when users click and use the Back button to return to the search results page. Users branch for 9.1% of clicks, although some users and queries tend to have more branches than others.

Due to a number of usability advantages, in some cases branching is replacing backtracking. Search examination behavior differs for queries requiring more than two clicks, which are likely to represent informational or difficult queries: users click on search results from top to bottom more often when they branch than when they backtrack. The time between clicks is substantially shorter for branching than backtracking. A large portion of these clicks are 2–3 seconds apart whereas backtracking queries possess a wider range of click intervals.

This set of differing branching behaviors led us to our click model experiment. We find that a modified click model performs better for branching by knowing that users do not examine the linked search result when they click, a previously held assumption in existing click models. This behavior means that the user continues examining even when they have clicked a potentially satisfying result. Better accounting for user search strategies can improve click models for query sessions in which branching occurs.

This work can be fruitful beyond search result pages. Search engines can distinguish between branching and backtracking by adding pageload events to the SERP, an event that can be implemented in other websites with hyperlinks like social news or photos sharing sites. Many sites can also use branching information to build user models and help rank their items.

10. ACKNOWLEDGMENTS

The authors thank Susan Dumais and Yu Chen for insightful discussions about the work, and Katherine Ye for editing earlier drafts of the paper.

11. REFERENCES

- [1] A. Aula, N. Jhaveri, and M. Käki. Information search and re-access strategies of experienced web users. In *Proceedings of WWW*, pp. 583–592, 2005.
- [2] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of WWW*, pp. 1–10, 2009.
- [3] F. Chierichetti, R. Kumar, and A. Tomkins. Stochastic models for tabbed browsing. In *Proceedings of WWW*, pp. 241–250, 2010.
- [4] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of WSDM*, pp. 87–94, 2008.
- [5] P. Dubroy and R. Balakrishnan. A study of tabbed browsing among mozilla firefox users. In *Proceedings of CHI*, pp. 673–682, 2010.
- [6] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of SIGIR*, pp. 331–338, 2008.
- [7] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *Proceedings of WWW*, pp. 11–20, 2009.
- [8] A. Hassan, R. Jones, and K. L. Klinkner. Beyond dcg: user behavior as a predictor of a successful search. In *Proceedings of WSDM*, pp. 221–230, 2010.
- [9] Y. He and K. Wang. Inferring search behaviors using partially observable markov model with duration (pomd). In *Proceedings of WSDM*, pp. 415–424, 2011.
- [10] J. Huang and R. W. White. Parallel browsing behavior on the web. In *Proceedings of Hypertext*, pp. 13–18, 2010.
- [11] B. J. Jansen and A. 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.
- [12] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of SIGIR*, pp. 154–161, 2005.
- [13] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of CIKM*, pp. 699–708, 2008.
- [14] B. MacKay and C. Watters. Exploring multi-session web tasks. In *Proceedings of CHI*, pp. 1187–1196, 2008.
- [15] M. R. Meiss, B. Gonçalves, J. J. Ramasco, A. Flammini, and F. Menczer. Agents, bookmarks and clicks: a topical model of web navigation. In *Proceedings of Hypertext*, pp. 229–234, 2010.
- [16] Y. Miyata and D. Norman. *Psychological issues in support of multiple activities*. 1986.
- [17] A. Spink, M. Park, B. J. Jansen, and J. Pedersen. Multitasking during web search sessions. *Information Processing & Management*, 42:264–275, January 2006.
- [18] A. Thatcher. Web search strategies: The influence of web experience and task type. *Information Processing & Management*, 44(3):1308–1329, May 2008.
- [19] M. Viermetz, C. Stolz, V. Gedov, and M. Skubacz. Relevance and impact of tabbed browsing behavior on web usage mining. In *Proceedings of Web Intelligence*, pp. 262–269, 2006.
- [20] K. Wang, N. Gloy, and X. Li. Inferring search behaviors using partially observable markov (pom) model. In *Proceedings of WSDM*, pp. 211–220, 2010.
- [21] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Off the beaten tracks: exploring three aspects of web navigation. In *Proceedings of WWW*, pp. 133–142, 2006.
- [22] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Not quite the average: An empirical study of web use. *ACM Transactions on the Web*, 2:1–31, March 2008.
- [23] H. Zhang and S. Zhao. Measuring web page revisitition in tabbed browsing. In *Proceedings of CHI*, pp. 1831–1834, 2011.
- [24] Y. Zhang, D. Wang, G. Wang, W. Chen, Z. Zhang, B. Hu, and L. Zhang. Learning click models via probit bayesian inference. In *Proceedings of CIKM*, pp. 439–448, 2010.
- [25] F. Zhong, D. Wang, G. Wang, W. Chen, Y. Zhang, Z. Chen, and H. Wang. Incorporating post-click behaviors into a click model. In *Proceedings of SIGIR*, pp. 355–362, 2010.