Evaluating Web Search Using Task Completion Time*

Ya Xu
Stanford University
yax@stanford.edu

David Mease
Google Inc.
dmease@google.com

ABSTRACT

We consider experiments to measure the quality of a web search algorithm based on how much total time users take to complete assigned search tasks using that algorithm. We first analyze our data to verify that there is in fact a negative relationship between a user's total search time and a user's satisfaction for the types of tasks under consideration. Secondly, we fit models with the user's total search time as the response to compare two different search algorithms. The two search algorithms we chose for comparison are close in quality, but still differ enough that other evaluation methods have had some degree of success in separating them. We confirm that our methodology is in fact sensitive enough to detect that one of these two algorithms has a statistically significant speed advantage over the other. Finally, we propose an alternative experiential design which we demonstrate to be a substantial improvement over our current design in terms of variance reduction and efficiency. The alternative design we use is a type of cross-over design, which proves to be advantageous since it mitigates the large variation in task completion times that we observe among different users carrying out the same task.

Categories and Subject Descriptors

H.1 [**Information Systems**]: Models and principles; H.3 [**Information Systems**]: Information storage and retrieval; G.3 [**Mathematics of Computing**]: Probability and Statistics

General Terms

Design, Experimentation, Performance, Measurement

Keywords

Experiment design, Evaluation metrics, Interactive IR and visualization, Question answering

1. INTRODUCTION

Traditional evaluation techniques for Information Retrieval (IR) systems in general focus on combining relevance judgements from individual documents. Common metrics in this case are precision, recall, Mean Average Precision (MAP) [6] and binary preference (bpref) [7]. However, these types of evaluation miss a lot of aspects of the user experience. Ingwersen and Järvelin [8] discuss some of these problems in their book, such as the assumption of document independence as well as the lack of user interaction. Additionally, the user interaction aspect will continue to become a larger issue as modern search engines add more interactive functionality to assist users. For instance, most popular search engines currently offer tools to assist users with query formulations such as query suggestions, query refinements and query completions.

For these reasons we believe that there is strong justification to continue to develop evaluation methodology using user-oriented techniques such as in Kagolovsky and Moehr [10]. Specifically, in this paper we seek to directly address the question of whether the search algorithm helps the user complete the task more efficiently. We will use a type of interactive evaluation in which users are assigned a task and the metric of interest is the time until the user completes the task.

Using the time until task completion is attractive as a metric for evaluating search algorithms since it is a holistic measurement of the user's interaction with the search task. However, one may be concerned that there could exist certain types of tasks in which a better IR system could lead to users actually spending more time on the task rather than less time. For that reason, we devote Section 3 of this paper to verifying that this is generally not the case for the types of task under consideration. That is, we show that the user's self-reported satisfaction has a negative relationship with the total time spent for the tasks we consider.

We note that other research has also established a negative relationship between quality and the time spent on the task. Allan et al. [2] showed that time on task had a statistically significant negative relationship with system retrieval accuracy as measured by bpref. Al-Maskari et al. [1] observed a statistically significant decrease in the geometric mean time to find the first relevant document when comparing a system with high average precision to a system with low average precision. Su [12] identified time as the most frequently mentioned reason given by users as contributing to their rating of the overall success of the IR system. Turpin and Scholer [13] found a negative relationship between precision at rank 1 and the time users took to find their first relevant document.

The data we use in this paper come from an experiment in which we compared the task completion times for paid participants assigned to use two different search algorithms.

*A two page version of this paper was published in *Proceedings of ACM SIGIR,* 2009

The experiment and the data collection are described in detail in Section 2. In order to use this data to compare the two search algorithms, it is helpful to control for variation due to user effects as well as task variation. In Section 4 of the paper we do this by fitting statistical models. Finally, in the last part of the paper we address the question of the experimental design. In Section 5 we show that by using a better design we can improve the overall efficiency of the experiment dramatically.

2. DATA

2.1 Search Algorithms A and B

The data we collect will compare task completion times for two search algorithms which we call search algorithm A and search algorithm B. These two algorithms were chosen because they show a measurable quality difference using an nDCG-type metric [9]. Specifically, algorithm A has a moderate advantage over algorithm B. If our time until task completion measurement is sensitive enough, we believe we will be able to confirm this advantage.

2.2 Tasks

In order to create the collection of tasks, we had 150 paid participants describe a difficult task which they had recently attempted. Specifically, part of the instructions read

> We are looking for a story about some information which you have recently tried to find on the internet but had a difficult time in doing so.

We provided a number of other guidelines including

> . . . the information needed to answer the question must be publicly accessible on the internet.

Six of the resulting 150 tasks had to be removed since they were not clearly formulated or did not follow the instructions, and some of the remaining tasks were edited slightly for readability. From these 144 tasks we randomly sampled 100 final tasks to be used in our experiment. Two examples which are representative are given below.

> Example Task #1:
> I once heard a song in the ending credits from a movie about a group of young lawyers or college students from back in the 80's. Jami Gertz and Kirk Cameron were stars in this movie. I think the song is called "Forever Young" but I want to know what the movie is called and who sings the song.
>
> Example Task #2:
> I'm trying to find out what Washington State governor served the shortest term in the past hundred years.

2.3 Users

Once the 100 tasks were obtained, another group of 200 paid participants was selected to act as users to attempt these tasks. The 200 paid participants were randomly split into 2 groups of 100 each, with one group assigned to use search algorithm A and the other assigned to use search algorithm B. These participants (which we will call "users" going forward) acquired tasks until all tasks were completed by 30 users each. Users were permitted to keep acquiring tasks until they desired to stop or until there were no more remaining. Some number of user's tasks had to be discarded due to data entry errors, giving slightly less than the 30 users per task desired. For search algorithm A there was an average of 28.5 users per task and for search algorithm B there was an average of 29.1 users per task. Some users in each group of 100 did not actually have a chance to participate in the experiment since all tasks had been completed before they began. Of the 100 users assigned to search algorithm A, only 93 actually participated giving an average of 30.6 tasks per user. For the 100 users assigned to search algorithm B, only 83 actually participated giving an average of 35.5.

2.4 Time

In the instructions for the project, the users were told to read the task and then to click a "start searching" button which would begin the search session by opening the appropriate search algorithm. The users were instructed to

> . . . keep searching until you believe you have found the answer or until you think a typical user would give up.

When finished the users were asked to click a "finish searching" button so that we could record the total task time.

The resulting task times for both search algorithm A and search algorithm B user groups revealed heavily right skewed distributions. This would cause trouble for our descriptive analysis and violate some assumptions for our quantitative analysis later. The log transformed time, however, generally follows a normal distribution for each group. Figure 1 shows the distribution for search algorithm A. The distribution for algorithm B is similar. Therefore, we use the (natural) log of the time (in seconds) throughout the entire paper. The average log time for algorithm A is 5.21 with a standard deviation of 1.0, compared to 5.37 with a standard deviation of 0.90 for B. This suggests an advantage for search algorithm A over B, but in order to correctly quantify the error in the estimate and determine statistical significance we will need to control for variation introduced by both users and tasks. This is done in Section 4. First, however, we examine the data to visually confirm that there is a negative relationship between time to completion and satisfaction.

3. RELATIONSHIP BETWEEN TIME AND SATISFACTION

In order to confirm that our experimental set up yields a negative relationship between task time and satisfaction, we asked the users to self report on their satisfaction with the task immediately upon clicking the "finish searching" button. Specifically, we asked the participants to indicate how satisfied they were with their search experience using the five choices below.

1) Very Dissatisfied
2) Dissatisfied
3) Neutral
4) Satisfied
5) Very Satisfied

We will refer to this measurement as "satisfaction" throughout this paper. The numeric computations will be based on 1 through 5 coding with 1 being "Very Dissatisfied".

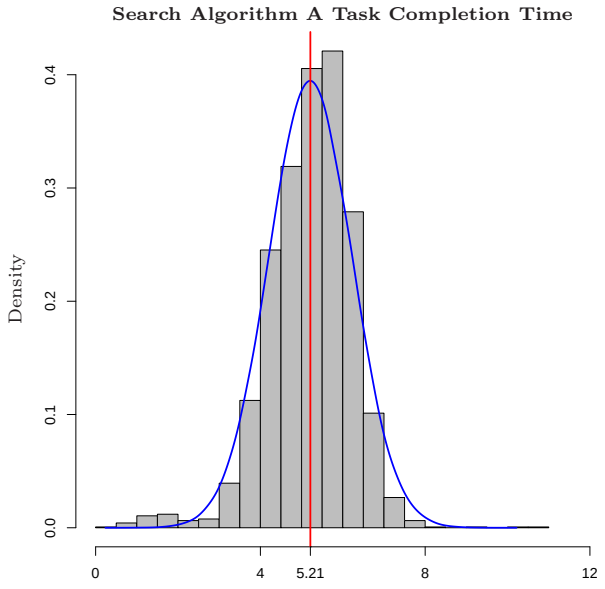

Figure 1: The empirical distribution of the log transformed time generally follows a normal distribution.

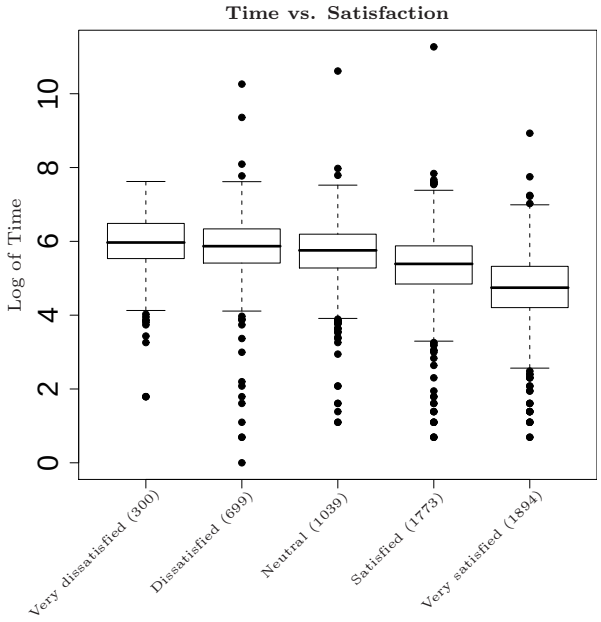

Figure 2: The boxplots show the relationship between log time and satisfaction across all user/task pairs. The x-axis labels also include the number of observations for each level.

3.1 Across All User/Task Pairs

We begin by looking at the relationship between the total task time and satisfaction across all 5756 user/task pairs (2846 from search algorithm A and 2910 from search algorithm B). Figure 2 provides boxplots for the log time for each of the 5 satisfaction levels. The median log time (as indicated by the middle bars of the boxplots) clearly decreases as the satisfaction level increases. The relationship is fairly strong, with a correlation of -0.42.

3.2 Controlling for User Variation

One of the reasons that the correlation across all user/task pairs is not even stronger than -0.42 is that there is substantial variation from user to user with regard to the time variable. In other words, some users simply consistently take longer than other users to complete tasks. We can eliminate some of this variability from our analysis here in a couple of different ways. One way is to aggregate all of the users for each task by taking the mean log time and the mean satisfaction score (again using the 1 though 5 numerical scale). This is shown in Figure 3 which has a point for each of the 100 tasks on search algorithm A and a point for each of the 100 tasks on search algorithm B. Here we see a very strong relationship, with a correlation of -0.84 for the search algorithm A group and -0.86 for the search algorithm B group. Clearly the tasks which take the longest time are those which lead to the lowest satisfaction scores.

Because the 100 tasks are each done using both search algorithms A and B, we have drawn line segments in Figure 3 to connect all algorithm A points with their corresponding algorithm B points. Also note that Figure 3 has a somewhat non-linear shape at the top end of the scatter cloud, which can be partially explained by the fact that satisfaction scores are capped at 5.0.

A second way to analyze this same relationship is to consider the data for each user separately. If we compute each

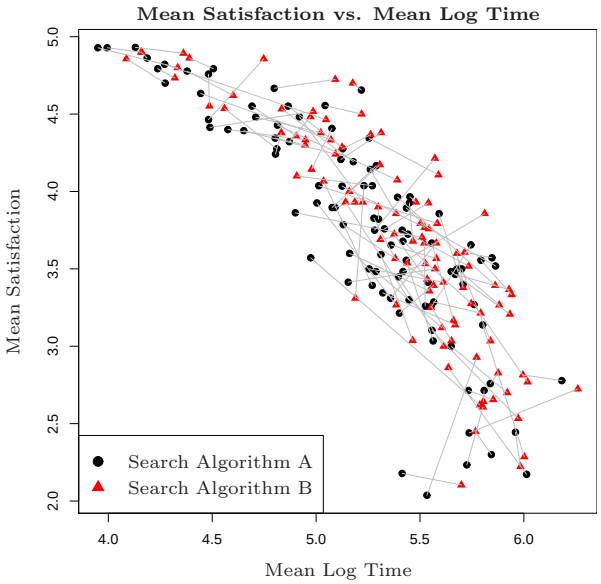

Figure 3: Log time and satisfaction are highly correlated after averaging within the tasks. Every pair of points corresponding to the same task is connected with a line segment.

user's correlation between his or her log times and satisfaction scores for the different tasks he or she attempted, we can see that the majority of these correlations are negative. The histogram of the correlations for users who have done at least 10 tasks is shown in Figure 4. Out of these 142 users, 141 had a negative correlation. Also, more than 50% of these correlations are less than -0.6. Thus we can con-

clude that for most users, the tasks which take them the longest time are generally those tasks which make them the most dissatisfied.

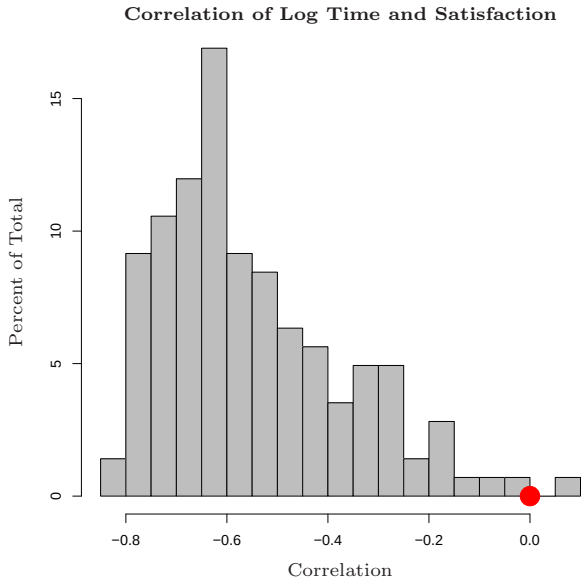

Figure 4: The histogram of correlations between log time and satisfaction for each user shows that almost all correlations are negative.

3.3 Delta Time Versus Delta Satisfaction

While it is clear from the analysis so far that tasks which take users the longest time are the tasks which make them the most dissatisfied, this does not necessarily mean that if we change a search algorithm in a way that leads users to take less time that we should necessarily conclude users are more satisfied. For this type of inference, we should not look across different tasks but rather we need to look at the same task under differing quality conditions. Fortunately, the data we have allow us to do just that since the same 100 tasks are completed using both search algorithms A and B. Thus, if we study the relationship between the delta in the log time and the delta in the satisfaction scores for each task using the two search algorithms, we can see whether an increase in time for a task is predictive of a decrease in satisfaction for the same task.

Looking back to Figure 3 we can believe that this will likely be the case, since in that figure the line segments joining the pairs of tasks generally seem to have negative slopes which are similar in magnitude. We can also examine this relationship directly in Figure 5 which plots the difference in the mean satisfaction score against the difference in the mean log time between search algorithms A and B for all of the 100 tasks. Indeed, the proposed negative relationship does exist with a correlation of -0.44. Thus we do in fact have support for the belief that decreases in task time are predictive of increases in satisfaction.

4. SEARCH ALGORITHM COMPARISON

We now turn to the question of whether or not our experiment yields sufficient evidence to conclude that one search

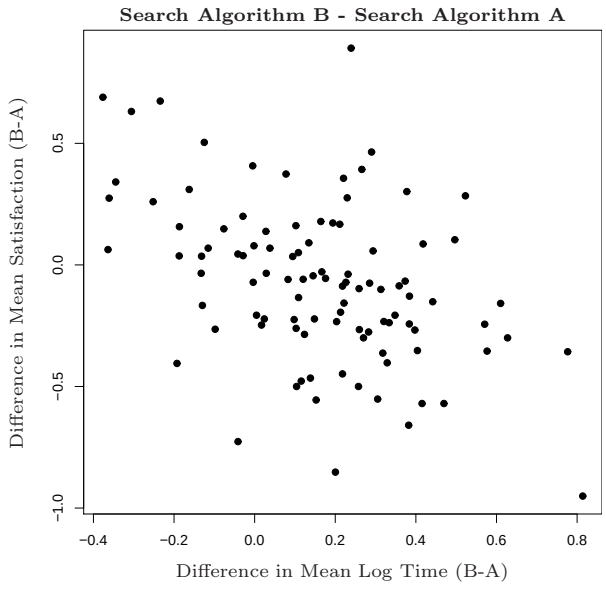

Figure 5: The difference in average log time and the difference in average satisfaction between search algorithms A and B are negatively correlated across the 100 tasks.

algorithm actually leads users to find their results faster than the other. Recall that in Section 2 we mentioned that the average log time for search algorithm A was 5.21 compared to 5.37 for search algorithm B. This would suggest search algorithm B is $e^{(5.37-5.21)} - 1 = 17\%$ slower. Similarly, plotting the average task completion log time using search algorithm B against the average log time using search algorithm A for each of the 100 tasks (Figure 6) shows that users take less time on search algorithm A for 75 out of the 100 tasks. This seems to confirm results from other metrics which indicate that A is the preferable algorithm. However, regarding statistical significance, it would be incorrect to apply a simple binomial test or even a two-sample t-test on the log time since the observations are not independent. Any two tasks completed on the same search algorithm are done by an overlapping set of users. Likewise, when comparing the means of the log time we need to consider the variation in the data due to users as well as tasks to correctly determine the statistical significance and the error in the estimation. In fact, the user variation is of particular concern given the present experimental design in which the search algorithm A users are distinct from the search algorithm B users. With this design, if by chance a few more fast users were assigned to search algorithm A than B, that could potentially account for a difference as large as we are seeing.

To conclude whether there is indeed a significant difference in task time between the two search algorithms, we need an approach that will account for the correlation and variance introduced by users as discussed above. We propose to use an ANOVA model with mixed effects as described in the following subsection.

4.1 ANOVA Model

Analysis of variance (ANOVA) is a classical method in statistics to separate out variations due to different explana-

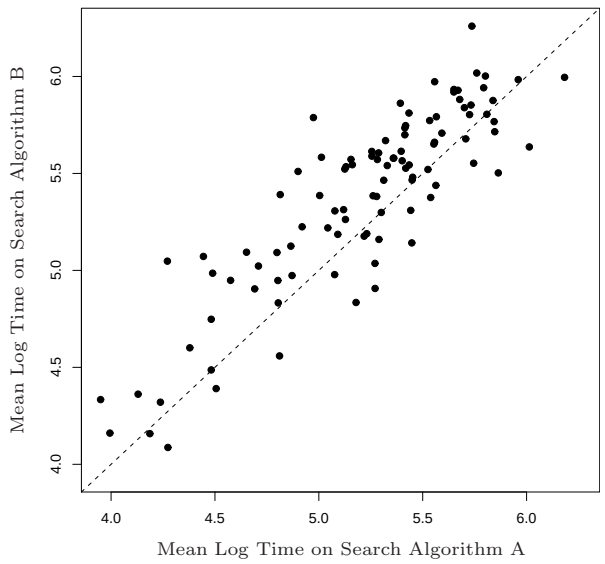

Figure 6: Users spend a longer average time on search algorithm B than on search algorithm A for 75 of the 100 tasks. The line plotted is the diagonal $y = x$ line, and 75 out of the 100 points lie above it.

tory variables. For background on ANOVA, see the book by Searle et al. [11]. Under our current experimental design, there are three variables contributing to the variation in time: the search algorithm effect, the user effect and the task effect. Because we are not interested in any particular user nor any particular task in the experiment, we treat both user and task effects as random in the model. By associating common random effects to observations sharing the same users or tasks, the model flexibly represents the covariance structure in the data. Conversely, the search algorithm effect is treated as a fixed effect, and it measures the difference between the two search algorithms after accounting for the variations in users and tasks. The model has the following mathematical form:

$$\log(Time)_{ij} = \mu + E_{B-A} + U_i + T_j + \epsilon_{ij} \ . \qquad (1)$$

Here, μ is the grand mean, E_{B-A} is the fixed effect due to the difference between the search algorithms A and B, U_i and T_j are random effects due to the variations across users and tasks respectively, and ϵ_{ij} is the error term. We follow the classical random effect models and suppose that all of the random terms are independent and are normally distributed with mean 0 and variances σ_U^2, σ_T^2 and σ_ϵ^2 for users, tasks and errors respectively. The normality assumptions seem reasonable since the log time is approximately normally distributed as shown in Figure 1.

4.2 Results

We fit the model in (1) using the `lme4` package [4, 5] available in `R`, an open source language for statistical computing. The resulting parameter estimates are given in Table 1. For reasons beyond the scope of this paper, p-values estimated directly from t-statistics in a mixed-effects model may not be accurate [3]. Therefore, we consider the 95% confidence interval estimated using Markov Chain Monte Carlo sampling (function `mcmcsamp` in `R`) instead of the naive p-value to check whether the search algorithm effect is significant.

The difference between the two search algorithms E_{B-A} is estimated to be 0.16 with a standard deviation of 0.079 and a 95% confidence interval of $(0.037, 0.290)$. In other words, the same user would take about $e^{0.16} - 1 = 17\%$ longer to complete the same task on search algorithm B than on A with a 95% confidence interval of (4%, 33%). This result confirms an advantage for search algorithm A over B for our task completion time metric, just as we have observed using an nDCG-type metric.

The increase in time for algorithm B is statistically significant after allowing for random user and task effects. However, the confidence interval (4%, 33%) is considerably wide. This is largely due to the current experimental design in which any given user does tasks on either A or B exclusively. We will examine a better experimental design in the following section.

Interestingly, from Table 1 we note that users seem to be slightly more variable than tasks, as σ_U^2 is estimated to be somewhat larger than σ_T^2. The practical interpretation of this is that two randomly selected users assigned to the same task will generally differ slightly more than two randomly selected tasks assigned to the same user. This large user variation is the main reason for the improvements we will see using the design proposed in the following section. Finally, we also note that the likelihood ratio tests show that both user and task random effects are statistically significant with p-values less than 10^{-15}.

Parameter	Estimation
μ	5.22
E_{B-A}	0.16
σ_U^2	0.23
σ_T^2	0.20
σ_ϵ^2	0.53

Table 1: ANOVA model parameter estimates

5. CROSS-OVER EXPERIMENTAL DESIGN

The results from the current experimental design are encouraging. After accounting for the substantial variation in both users and tasks, we are able to find a significant difference between the two search algorithms. However, the current experimental design may fail to distinguish two search algorithms which are closer in quality due to the large uncertainty in the estimation noted in the previous section.

The current experimental design protects against variation due to tasks since the same tasks are used for both search algorithms. However, user variation remains a problem due to the algorithm A users being distinct from the algorithm B users. Thus, any variation among users directly leads to increased uncertainty in our estimated search algorithm difference. As we saw in the previous section, this is a concern since the user variation is estimated to be slightly larger than the task variation. To overcome this problem, we propose a cross-over experimental design, where every participant uses both search algorithms, as described in detail in Section 5.2.

In this section, we will take a model-based approach to

compare the current experimental design with this proposed cross-over design. We first examine analytically the variance of the estimated search algorithm difference E_{B-A} under both designs in Subsections 5.1 and 5.2. We then proceed to quantify the improvement from the cross-over design in terms of variance, power and sample size. Throughout this section we will assume that our data is generated from model (1).

5.1 Variance Using Current Design

We will begin with the current design and analyze theoretically the variance of the estimated search algorithm effect under model (1). Suppose there are $2m$ users and $2n$ tasks in the experiment. Without loss of generality we assign users 1 through m to use search algorithm B and users $m+1$ through $2m$ to use search algorithm A. For our purpose here, we simplify the design a little bit by assuming every user does all of the $2n$ tasks. This was not the case for the data we collected, but the results we show in this section can be generalized to cases in which users do not do all tasks as discussed in Section 5.3.

With this design, the search algorithm effect can be computed as

$$\widehat{E}_{B-A} = \frac{1}{2mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{2n}\log(Time)_{ij} - \sum_{i=m+1}^{2m}\sum_{j=1}^{2n}\log(Time)_{ij}\Big).$$

Under model (1), this can be expanded to

$$\begin{aligned}\widehat{E}_{B-A} &= \frac{1}{2mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{2n}(\mu + E_B + U_i + T_j + \epsilon_{ij}) \\ &\quad - \sum_{i=m+1}^{2m}\sum_{j=1}^{2n}(\mu + E_A + U_i + T_j + \epsilon_{ij})\Big) \\ &= (E_B - E_A) + \frac{1}{m}\Big(\sum_{i=1}^{m}U_i - \sum_{i=m+1}^{2m}U_i\Big) \\ &\quad + \frac{1}{2mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{2n}\epsilon_{ij} - \sum_{i=m+1}^{2m}\sum_{j=1}^{2n}\epsilon_{ij}\Big).\end{aligned}$$

We see here that the random effects due to tasks are canceled since the same tasks are used for both search algorithms, but $\widehat{E}_{B-A}$ still depends on user effects, which will contribute to the variance as we will see next. Using the normality and independence assumptions for the random effects U_i and ϵ_{ij}, we can compute the variance of $\widehat{E}_{B-A}$ to be

$$\begin{aligned}\mathrm{Var}(\widehat{E}_{B-A}) &= \frac{1}{m^2}(2m\sigma_U^2) + \frac{1}{4m^2n^2}(4mn\sigma_\epsilon^2) \\ &= \frac{2}{m}\sigma_U^2 + \frac{1}{mn}\sigma_\epsilon^2. \qquad (2)\end{aligned}$$

Thus the uncertainty in our estimated search algorithm effect does not depend on the task variance component σ_T^2 but it does depend on the user variance component σ_U^2, and the effect of this component can only be reduced by increasing the number of users $2m$. This is problematic since recruiting more participants for the study can be time consuming and expensive.

5.2 Variance Using Cross-over Design

To eliminate the variation due to users as well as tasks, we desire a design in which we not only use the same tasks for both search algorithms, but we also use the same users for both search algorithms. It is not immediately obvious how to do this since we do not want to permit the same user to do the same task on both search algorithms, as that would introduce a learning effect. Not to mention, this would be a very artificial experience for the users. However, using what is known as a cross-over design will solve this problem.

In our cross-over design, the same user will be assigned to both search algorithms A and B, but for different tasks. Again, we suppose there are $2m$ users and $2n$ tasks. Without loss of generality, we assign the first m users to do the first n tasks on B and the second n tasks on A, while the second m users do the second n tasks on B and the first n tasks on A. Again, every user does all of the $2n$ tasks, only half on search algorithm A and the other half on B.

Under this design, the search algorithm effect can then be computed as

$$\begin{aligned}\widehat{E}_{B-A} = \frac{1}{2mn}\Big(&\sum_{i=1}^{m}\sum_{j=1}^{n}\log(Time)_{ij} + \sum_{i=m+1}^{2m}\sum_{j=n+1}^{2n}\log(Time)_{ij} \\ &- \sum_{i=1}^{m}\sum_{j=n+1}^{2n}\log(Time)_{ij} - \sum_{i=m+1}^{2m}\sum_{j=1}^{n}\log(Time)_{ij}\Big).\end{aligned}$$

As we did for the current design, we can expand this expression under model (1) and group the same effects together to get

$$\begin{aligned}\widehat{E}_{B-A} &= (E_B - E_A) \\ &\quad + \frac{1}{2mn}\Big(\sum_{i=1}^{m}nU_i + \sum_{i=m+1}^{2m}nU_i - \sum_{i=1}^{m}nU_i - \sum_{i=m+1}^{2m}nU_i\Big) \\ &\quad + \frac{1}{2mn}\Big(\sum_{j=1}^{n}mT_j + \sum_{j=n+1}^{2n}mT_j - \sum_{j=n+1}^{2n}mT_j - \sum_{j=1}^{n}mT_j\Big) \\ &\quad + \frac{1}{2mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\epsilon_{ij} + \sum_{i=m+1}^{2m}\sum_{j=n+1}^{2n}\epsilon_{ij} \\ &\qquad - \sum_{i=1}^{m}\sum_{j=n+1}^{2n}\epsilon_{ij} - \sum_{i=m+1}^{2m}\sum_{j=n+1}^{2n}\epsilon_{ij}\Big) \\ &= (E_B - E_A) + \frac{1}{2mn}\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}\epsilon_{ij} + \sum_{i=m+1}^{2m}\sum_{j=n+1}^{2n}\epsilon_{ij} \\ &\qquad - \sum_{i=1}^{m}\sum_{j=n+1}^{2n}\epsilon_{ij} - \sum_{i=m+1}^{2m}\sum_{j=n+1}^{2n}\epsilon_{ij}\Big).\end{aligned}$$

Now the effects due to both users and tasks are canceled. Again, using the normality and independence assumptions for the errors, we can compute the variance of $\widehat{E}_{B-A}$ to be

$$\mathrm{Var}(\widehat{E}_{B-A}) = \frac{1}{4m^2n^2}(4mn\sigma_\epsilon^2) = \frac{1}{mn}\sigma_\epsilon^2. \qquad (3)$$

Compared with (2) from the current design, (3) from the cross over design completely removes the user variation, and hence largely reduces the uncertainty in the estimated search algorithm effect.

5.3 Variance Comparisons

We would like to quantify the improvement in terms of variance if we had used the cross-over design in our experiment. To do so, we first match the theoretical variance in (2)

with the empirical variance obtained in Section 4. We then compute the variance reduction using the cross-over design.

In the analysis for these two experimental designs above, we assumed that every user does all $2n$ tasks. We need to first relax this assumption a bit to more closely resemble the data we collected, in which users do not need to finish all tasks. We now suppose that each user does $2k$ tasks with $2k \leq 2n$ and the same task is done by the same number of users on both algorithms. Using similar computations to those in the previous sections, we can compute that (2) still holds, only with n replaced by k. The assumptions for the cross-over design can be relaxed similarly, with the difference being that each user does k tasks on both algorithms. Again, we can get (3) with n replaced by k.

In our experiment, every user did not do the same number of tasks with each search algorithm. However, on average we have roughly $m = 90$ and $k = 15$. Using the estimated values in Table 1, $m = 90$ and $k = 15$ give a theoretical standard deviation of 0.074 as computed by (2) (with n replaced by k). We note that this is close to the actual estimate of 0.079 in Section 4. From (3) we can compute that the cross-over design reduces this standard deviation to 0.020, which is a 73% reduction. We can further compute the (theoretical) 95% confidence interval for the search algorithm effect using the cross-over design as (12.8%, 22.0%), which is substantially narrower than we reported in Section 4 for the current design.

Design	$\text{Var}(\widehat{E}_{B-A})$	$\text{Sd}(\widehat{E}_{B-A})$
current design	0.00551	0.074
cross-over design	0.00039	0.020

Table 2: Theoretical variance comparison between the two designs

5.4 Power Comparisons

In addition to considering the reduction in variance resulting from the cross-over design, we can also quantify the improvement in terms of the *power*. The term power has to do with the probability of detecting a statistically significant difference assuming one exists. For our purposes, we assume ranking algorithm B is truly slower than ranking algorithm A. We hence define our power to be the probability of a design correctly detecting B's loss at a (two-sided) 95% confidence level, which can be written as

$$1 - \Phi\left(\frac{\Phi^{-1}(.975)\,\text{Sd}(\widehat{E}_{B-A}) - E_{B-A}}{\text{Sd}(\widehat{E}_{B-A})}\right), \qquad (4)$$

where Φ is the standard normal cumulative distribution function and Φ^{-1} is its inverse.

From (4), we can observe that the power will increase as $\text{Sd}(\widehat{E}_{B-A})$ decreases, which is intuitive since less uncertainty in our estimate will increase the likelihood of detecting a true difference. Specifically, when the true value of E_{B-A} is in fact 0.16, the standard deviation of 0.074 from our current design (as in Table 2) leads to a power of 0.58. For the cross-over design the smaller standard deviation of 0.020 increases the power to 1.00 to 2 decimals of accuracy.

Equation (4) also shows that the power monotonically increases with E_{B-A}, which is intuitive since larger differences are easier to detect. Figure 7 plots the power as a function of E_{B-A} for both the cross-over design and the current design. From this graph we can see that the cross-over design is powerful enough to detect differences on the order of $E_{B-A} = 0.06$ with probability close to .85. For perspective, a difference of $E_{B-A} = 0.06$ implies a $e^{0.06} - 1 = 6\%$ difference in time until task completion. The current design has a very small (roughly 13 percent) chance of detecting such a small difference. The vertical line on the plot shows the power for $E_{B-A} = 0.16$ under both designs as discussed in the previous paragraph.

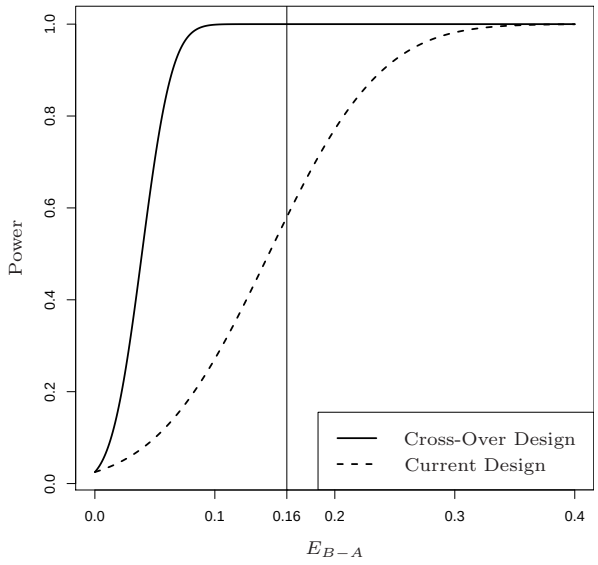

Figure 7: Power comparison for current design and cross-over design

5.5 Sample Size Comparisons

One final way to compare the efficiency of the cross-over design to that of the current design is in terms of sample size. Specifically, we can ask the following question. If we wanted to achieve the same variance (and thus power) as the cross-over design using the current design, how many more users would we need? Section 5.3 shows that with $m = 90$ and $n = 15$, the cross-over design gives a variance of 0.00039. With the same $n = 15$, the current design needs $m = 1,270$ users to achieve the same variance, computed by solving (2) for m. Thus we can say that the cross-over design reduces the amount of users (paid participants) needed by more than a factor of 10 in this case.

6. CONCLUSIONS

There are two main results in this paper. First, we confirmed that time until task completion has a negative correlation with user satisfaction on all levels. This general relationship has been observed in other studies, and our contribution has been to add more evidence in support of this.

Secondly, we have demonstrated that time until task completion can be used as a metric to differentiate ranking algorithms of moderately different quality in a reasonably sized experiment. However, because there is substantial variation

in different user's task completion times for the same tasks, we have shown that using a cross-over design provides considerable gains in efficiency. We note that this user variation is a natural challenge with a metric such as time until task completion since certain users simply spend more time than others for the same task.

7. ACKNOWLEDGMENTS

The authors are grateful to many of their Google colleagues for their assistance with this paper, especially Rehan Khan, Scott Huffman, Shan Wang, Eiji Hirai, Anna Ma, Udi Manber and Rajan Patel.

8. REFERENCES

[1] A. Al-Maskari, M. Sanderson, P. Clough, and E. Airio. The good and the bad system: does the test collection predict users' effectiveness? In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, New York, NY, USA, 2008. ACM.

[2] J. Allan, B. Carterette, and J. Lewis. When will information retrieval be "good enough"? In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 433–440, New York, NY, USA, 2005. ACM.

[3] R. H. Baayen, D. J. Davidson, and D. M. Bates. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, In Press, Corrected Proof.

[4] D. Bates. Computational methods for mixed models. http://cran.us.r-project.org/web/ packages/lme4/ vignettes/Theory.pdf, December 2008.

[5] D. Bates. Linear mixed model implementation in lme4. http://cran.r-project.org/web/packages/lme4/ vignettes/Implementation.pdf, December 2008.

[6] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM.

[7] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM.

[8] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[9] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[10] Y. Kagolovsky and J. R. Moehr. Current status of the evaluation of information retrieval. *J. Med. Syst.*, 27(5):409–424, 2003.

[11] S. R. Searle, G. Casella, and C. E. McCulloch. *Variance Components*. Wiley, New York, 1992.

[12] L. T. Su. Evaluation measures for interactive information retrieval. *Inf. Process. Manage.*, 28(4):503–516, 1992.

[13] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18, New York, NY, USA, 2006. ACM.