An Experimental Comparison of Click Position-Bias Models

Nick Craswell, Onno Zoeter and Michael Taylor
Microsoft Research, Cambridge UK
{nickcr,onnoz,mitaylor}@microsoft.com

Bill Ramsey
Microsoft Research, Redmond USA
brams@microsoft.com

ABSTRACT

Search engine click logs provide an invaluable source of relevance information, but this information is biased. A key source of bias is presentation order: the probability of click is influenced by a document's position in the results page. This paper focuses on explaining that bias, modelling how probability of click depends on position. We propose four simple hypotheses about how position bias might arise. We carry out a large data-gathering effort, where we perturb the ranking of a major search engine, to see how clicks are affected. We then explore which of the four hypotheses best explains the real-world position effects, and compare these to a simple logistic regression model. The data are not well explained by simple position models, where some users click indiscriminately on rank 1 or there is a simple decay of attention over ranks. A 'cascade' model, where users view results from top to bottom and leave as soon as they see a worthwhile document, is our best explanation for position bias in early ranks.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Experimentation

Keywords

Web search, models, click data, user behavior

1. INTRODUCTION

As people search the Web, certain of their actions can be logged by a search engine. Patterns of behaviour in logs can give an idea of the scope of user activity. They can also be indicative of success or failure of the engine. When deciding which search results (or ads or other elements) to present, click logs are of particular interest. These record which results-page elements were selected (clicked) for which query. Click log information can be fed back into the engine, to tune search parameters or even used as direct evidence to influence ranking [5, 1].

A fundamental problem in click data is position bias. The probability of a document being clicked depends not only on its relevance, but on its position in the results page. In top-10 results lists, the probability of observing a click decays with rank. The bias has several possible explanations. Eye-tracking experiments show that the user is less likely to examine results near the bottom of the list, although click probability decays faster than examination probability so there are probably additional sources of bias [6].

Our approach is to consider several such hypotheses for how position bias arises, formalising each as a simple probabilistic model. We then collect click data from a major Web search engine, while deliberately flipping positions of documents in the ranked list. We finally evaluate the position bias models using the flip data, to see which is the best explanation of real-world position effects.

Although our experiment involves flips, our goal is to model position bias so we can correct for it, without relying on flips. With such a model it should be possible to process a click log and extract estimates of a search result's absolute 'click relevance'. Such estimates could be used in applications where an estimate of probability of click is useful such as ad ranking [9] or evaluation.

2. EXPLAINING POSITION BIAS

We present several hypotheses for how users view a search results list, and each of these leads to a model of position bias. The models are kept separate from their implementation details. Section 4 discusses our implementation choices for each model.

We number our positions $i \in \{1, \ldots, N\}$. In general we make no assumption about the positions, they might be linearly ranked, in a grid, in a staggered layout or arranged in some other way. However, our empirical observations come from a standard top-10 ranking, and one of our models assumes that users tend to view results in order from 1 towards 10.

Most of the models assume there is some underlying 'snippet relevance' variable associated with document d, denoted r_d. This is shorthand: $r_d = p(Click = true|Document = d)$. It is the probability that the user will think of the snippet as relevant enough to warrant a click, defined over the engine's population of users. The models attempt to ex-

WSDM'08, February 11–12, 2008, Palo Alto, California, USA.

plain the observed probability of click c_{di} for document d at rank i, based on r_d and other factors. The shorthand is: $c_{di} = p(Click = true|Document = d, Rank = i)$.

Baseline Hypothesis: The simplest explanation for position bias is that there is none. Users look at all results and consider each on its merits, then decide which results to click. In our baseline hypothesis, the probability of clicking a document at position i is the same as the probability of clicking it at position j.

$$c_{di} = r_d = c_{dj} \tag{1}$$

Note, in our experiments $j = i + 1$, the smallest possible change in position, so our baseline model is potentially a strong predictor. The baseline hypothesis would seem to be at odds with past studies, that have shown that results at or near rank 1 are more likely to be clicked and more likely to be viewed under eye tracking [6]. A decaying click curve can be consistent with the baseline, if the search engine ranks documents with decaying r_d then we will see fewer clicks on lower ranks. However, the baseline hypothesis is not consistent with a decaying attention curve.

Mixture Hypothesis: Another explanation for position bias is that some users will blindly or speculatively click on early ranks. This could happen if the user is uncertain or makes a hasty decision. In that case we have a mixture hypothesis, where some users click due to relevance r_d as in the baseline model and some users click blindly with probability b_i due to the document appearing in rank i. The proportion of users can be explained by a mixture parameter λ:

$$c_{di} = \lambda\, r_d + (1 - \lambda)\, b_i \tag{2}$$

This probabilistic model is new, and we will see that it is very difficult to fit the model against our empirical observations. However, a related approach was used without assuming a probabilistic mixture model, with notably more success. Agichtein *et al.* [1] corrected for position bias by subtracting the background click distribution from this query's click distribution. Then, search results with more than expected clicks were relevant and those with fewer than expected (negative) were irrelevant.

Examination Hypothesis: From eye tracking studies we have direct evidence that users are less likely to look at lower-ranked results. This suggests another hypothesis, which is that each rank has a certain probability of being examined. This could be modelled as a term x_i which is the probability of examining a document given that it is in rank i. To be clicked, a result must be both examined and relevant:

$$c_{di} = r_d\, x_i \tag{3}$$

Richardson *et al.* [9] proposed this model to explain position bias in the ranked list of ads, though did not give details on how x_i is learnt, as this was not the focus of the paper.

The three hypotheses so far can be seen as follows. The baseline hypothesis is that a click is explained by relevance. The mixture hypothesis is that a click is explained by relevance OR blind clicking. The examination hypothesis is that a click is explained by relevance AND examination. The latter two can be seen as two general classes of model, one incorporating a rank bias with OR, and the other with AND. For example, if we wish to fit the AND model to our data, it is not necessary to assume that x_i arises from the probability of examination. Instead it might be trust bias [6], that people trust certain ranks more than others, so a click arises from relevance AND trust (both things must happen before we observe a click). The rank-based term x_i might arise from a number of causes. At its most general, the hypothesis is that clicks arise from a document factor AND a rank factor.

Cascade Model: Now we present a new model for explaining the position effect, which assumes a linear traversal through the ranking, and that documents below a clicked result are not examined. It is inspired by the work of Joachims *et al.* [5, 6] which assumes a linear traversal through the ranking, ending at a clicked result. One classic model is Click > Skip Above. In that model, a clicked document at rank i is thought to be preferred to a skipped document at rank j. The document at rank j is skipped if i is clicked, $j < i$ and j is not clicked. However, our goal is to estimate the probability of click, rather than generate preference pairs.

In the cascade model, we assume that the user views search results from top to bottom, deciding whether to click each result before moving to the next. Each document d, is either clicked with probability r_d or skipped with probability $(1 - r_d)$. In the most basic form of the model, we assume that a user who clicks never comes back, and a user who skips always continues, in which case:

$$c_{di} = r_d \prod_{j=1}^{i-1} (1 - r_{docinrank:j}) \tag{4}$$

To observe a click, the user must have decided both to click (r_d) and skip the ranks above.

3. EXPERIMENTAL SETUP

To test our hypotheses for how clicks arise, we perform a controlled experiment, where we vary the rank at which a document is displayed and observe the change in probability of click. This is done for a small subset of users, as they perform searches in a major search engine. All flips are of adjacent ranks in the top 10, so there are 9 types of flip, that we number $m \in \{1, \ldots, 9\}$.

Each of our empirical observations pertains to a query and two URLs A and B that occurred at ranks m and $m + 1$ in the results list. Therefore an experiment can be identified by a quad: query, A, B, m. Note, the query is not used in our models, it is just used to group our observations (we do not group across queries). We present the results in the order AB and BA. Within an experiment we collect six counts. The number of times the search results were presented in the order AB (N_{A1B2}), the number of clicks on A (N_{A1}) and the number of clicks on B (N_{B2}). For cases where the order was reversed, we also have N_{B1A2}, N_{B1} and N_{A2}. Note, we use 1 and 2 to refer to the upper and lower ranks of the flip, even when $m \neq 1$.

We collect a large number of such experiments via random sampling, then filter in two ways. We ignore any experiments where there are ads or other elements placed above the top-10 list. This means our analysis of top-10 viewing behaviour is focused on cases where the first thing the user sees is our rank 1. Therefore, we do not need to model the effect of ads and other elements. The second filter is to remove experiments where $NA1B2 < 10$ or $NB1A2 < 10$. This leaves us with 108 thousand experiments, with roughly even numbers of experiments at each rank.

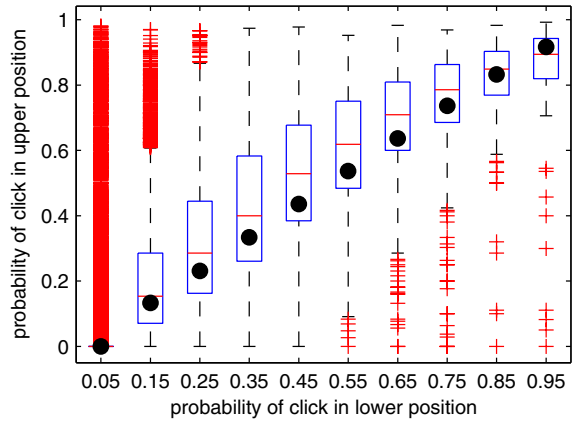

Figure 1: Box plot, all experimental data.

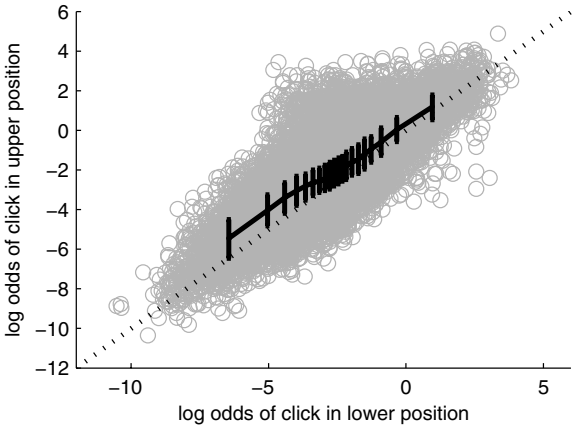

Figure 2: Plot of data from Figure 1 in log odds space, but showing only datapoints with nonzero clicks in both the lower and upper position.

To visualise the data in full we calculated probability of click using ratios. Within each experiment we pair the lower probability N_{B2}/N_{A1B2} with its upper probability N_{B1}/N_{B1A2}, and we also pair the lower probability N_{A2}/N_{B1A2} with its upper probability N_{A1}/N_{A1B2}. Rather than showing a scatter plot of lower and upper probabilities, we create 10 bins according to the lower probability, and show the upper probabilities of each bin, using a box plot (Figure 1). Each box shows the 50th percentile (median) as well as the 25th and 75th. The whiskers show the range of data above and below, up the 1.5 times the interquartile range, with a '+' sign for outliers.

The middle of the box shows the median of upper probabilities, so we show the median of the bin's lower probabilities as a '.', for purposes of comparison. The box plot shows the dataset in full, but has some problems. Most observations lie in the leftmost bin. In the rightmost bin, we actually see a decrease in median click probability in the upper position (this is because any such probability >90% is an outlier, so unlikely to be observed in both the upper and lower positions). For these reasons, we show our subsequent plots in log-odds space.

	AB	BA
9 folds	training data	
1 fold	observe	predict

Table 1: Experimental setup. The training folds can be used to fit parameters. In the test fold, the model is given click observations in position AB, and must predict probability of click in BA.

Figure 2 shows the same data in log odds space. Log odds of probability p is $\log(p/(1-p))$. This has the effect of stretching out the lower probabilities, so we can see the shape of the data more clearly. The error bar plot again shows the lower, median and upper quartiles (as in the box plot). Figure 3 shows the same plot, but separately for each type of flip. Note, our dataset contains a very large number of observations with zero clicks, and we treat these differently in experiments and log odds plots. In experiments we keep zeros and use smoothing as described in Section 4. In log odds plots we remove any datapoint with a zero click observation in the x-axis or y-axis. Zeros can not be shown as-is, because log odds of zero is negative infinity. Further, if they are included via smoothing or adding epsilon they tend to make the plot less readable and less informative about our non-zero datapoints.

3.1 Cross Entropy Evaluation

Our goal is to perform a model-agnostic experiment, such that our experimental design makes minimal assumptions about users. For example, we could have performed our experiments using relevance judgments and NDCG. However, the judgments of an individual relevance assessor when viewing a document have different characteristics from the clicks of a user population when viewing a snippet. In addition, metrics such as NDCG make assumptions about the Gain of a result and Discount function over ranks, and these may be inconsistent with our data.

Our design is to predict the clicks when documents are ranked in order BA, based on observing clicks in order AB. This evaluation does not need manual relevance assessments or NDCG. It bypasses the issue of snippet-content mismatch, since both decisions are made on the basis of snippets. If a model for position bias is accurate, it should be able to predict clicks in ordering BA based on observations in ordering AB.

We perform 10-fold cross validation. For each of the 10, we perform an experiment as indicated in Table 1. Models with parameters can be trained on 9 of the folds, which are otherwise ignored. Then, to test, the model is given click observations in the initial order AB. The model must predict the probability of click in order BA. Note, designation A and B is assigned at random and is not indicative of the engine's original order. We wish to model position bias independently of the engine's original ranking.

We measure the quality of the model's prediction using cross entropy:

$$\textit{Cross Entropy} = -\sum_e p(e) \log p'(e)$$

There are four possible events e: 1) click just A, 2) click just B, 3) click both or 4) click neither. The model has predicted probabilities $p'(e)$ for each e, based on observations of AB.

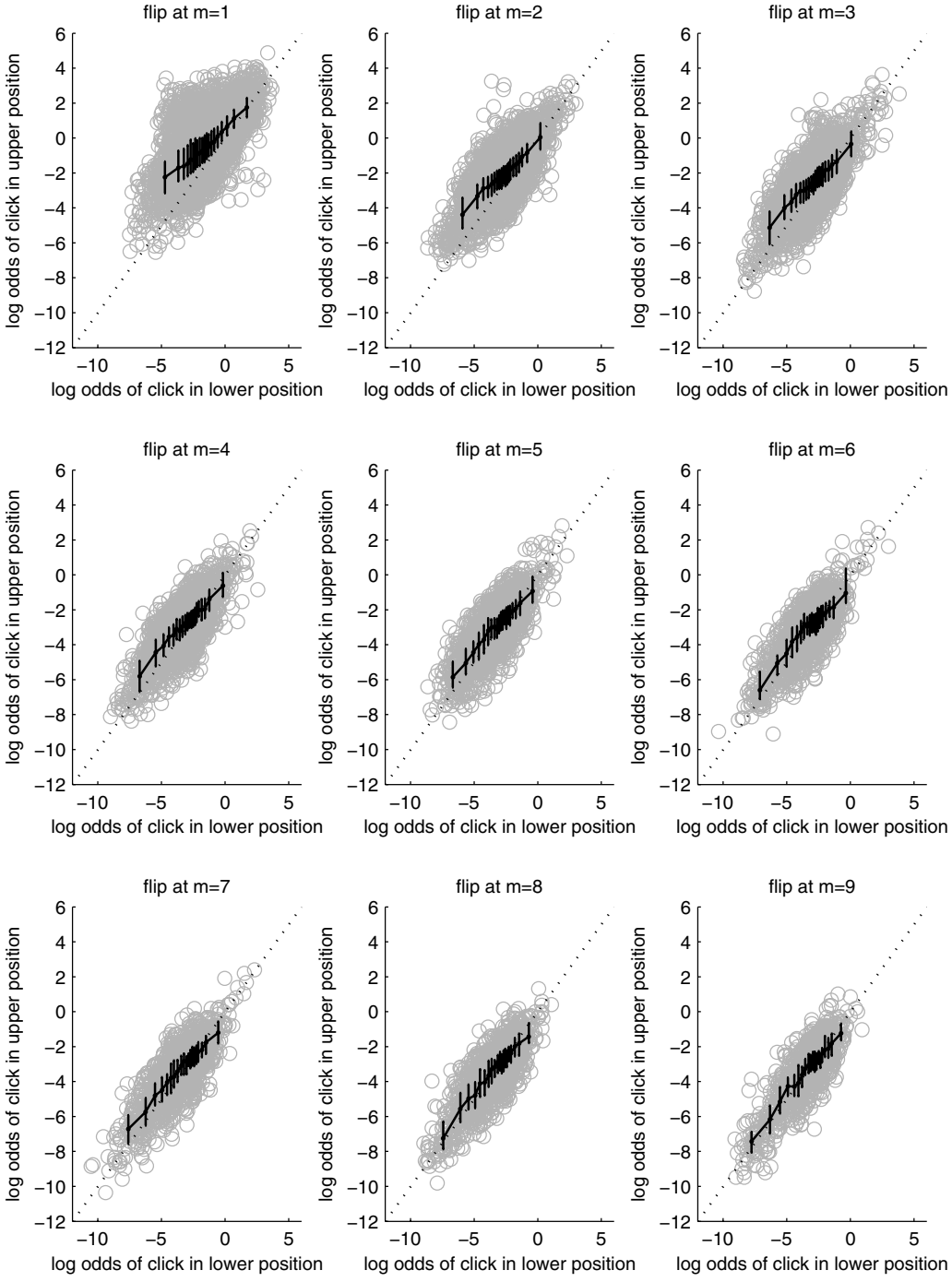

Figure 3: Plot as in Figure 2 broken down by rank (flip between rank m and $m+1$). When a median line lies near the diagonal, there is little systematic position bias. Error bars indicate upper and lower quartile.

Then we have the true probabilities $p(e)$ observed from BA. Cross entropy rewards a model where $p(e) \simeq p'(e)$. We calculate the average cross entropy for each (query,A,B,m) quad. We then average across quads to get overall cross entropy.

One difficulty in evaluation is that many of our models predict probability of click of A and B independently, meaning we assign nonzero probability that *both* results are clicked when presented. Whereas in the cascade model, it is impossible to observe a click on both A and B. Our solution is to evaluate across all 4 events, but never observe click-both. We show that this does not disadvantage the independent A-B models in Appendix A.

4. IMPLEMENTING THE MODELS

All our models begin with calculating the probability of click in the observed AB ordering:

$$c_{A1} = \frac{N_{A1} + \alpha}{N_{A1B2} + \alpha + \beta} \quad (5)$$

$$c_{B2} = \frac{N_{B1} + \alpha}{N_{A1B2} + \alpha + \beta} \quad (6)$$

Again this uses shorthand where 1 is rank m and 2 rank $m+1$. Our smoothing parameters are $\alpha = 1$ and $\beta = 1$. We experimented with different values and these were no worse than other options. Smoothing means we avoid predicting zero probability of click, which would lead to infinite cross entropy error. It also allows us to differentiate between the case with zero clicks and $N_{A1B2} = 10$ and the case with zero clicks and $N_{A1B2} = 50000$. In the latter case, we are more certain that probability of click is very low.

The **baseline model** predicts the BA probability of click based on unadjusted AB probabilities from equations (5) and (6). Thus $c_{A2} = c_{A1}$ and $c_{B1} = c_{B2}$.

The **mixture model**, based on equation (2) has:

$$r_A = \frac{c_{Ai} - (1-\lambda)\, b_i}{\lambda}$$

With some simple manipulation we get:

$$c_{B1} = c_{B2} + (1-\lambda)(b_1 - b_2)$$

Since λ, b_1 and b_2 are all parameters we can learn, we can replace these with a parameter w_1 such that: $c_{B1} = c_{B2} + w_1$ and $c_{A2} = c_{A1} - w_1$. We learn a weight w_i for each of the 9 types of flip that can happen in the top 10. For each flip type, we choose a w_i that minimises cross entropy error on the training folds.

Similarly, the **examination model** can be rearranged to be: $c_{A2} = c_{A1}/w_1$ and $c_{B1} = c_{B2} * w_1$. We can learn a weight for each of the 9 flip types, that minimises cross entropy error on the training folds.

In each of the above models, having estimated the two probabilities of click, we calculate the probabilities of the four possible events: $c_{A2}(1-c_{B1})$, $(1-c_{A2})c_{B1}$, $(1-c_{A2})(1-c_{B1})$, $c_{A2}c_{B1}$. We measure cross entropy error in this event space.

The **cascade model** learns no parameters on the training folds. Instead it uses the same ratios from the AB case (again equations (5) and (6)) as follows:

$$c_{A2} = c_{A1}\,(1 - c_{B1})$$

$$c_{B1} = c_{B2}/(1 - c_{A1})$$

This is precisely consistent with the cascade hypothesis and equation (4), for flips between ranks 1 and 2. However, lower down the ranking it tends to undercorrect. This is because in our current experiment we only know about a pair of documents, not the chain of documents that occurred above. Despite this disadvantage, the cascade model performs well enough to be included here.

The cascade model assigns probability=0 to the click-both event, and assigns probability= $(1 - c_{B1} - c_{A2})$ to click neither.

Finally, we describe a **logistic model** that is not based on a user behaviour hypothesis. Rather, it is based on the observation that in Figure 2 the desired correction should put our data on the diagonal line, and the current median line is somewhat straight and parallel with that line. To shift a line in log odds space we can do:

$$c_{B1} = logodds^{-1}(logodds(c_{B2}) + w)$$

$$c_{A2} = logodds^{-1}(logodds(c_{A1}) - w)$$

Converting the probability of click to log odds space, adding a weight and converting back to a probability. We learn a w for each of the 9 types of flip, to minimise cross entropy error on the training folds. This model is related to logistic regression.

5. RESULTS

First we discuss the constraints applied during parameter tuning, then the overall experimental results.

5.1 Constraints in Tuning

In the logistic model, we first transform our probability onto the real line, then add a weight, then transform results back into probabilities. This means that we can consider any real value for weight w without the possibility of producing an out-of-bounds probability (above 1 or below 0).

Unfortunately in the mixture model and the examination model, there is no such transformation, so we placed constraints on the values of parameters. If a parameter value caused an out-of-bounds error on the training set, we removed it from consideration. This caused a problem for both models, whose optimal parameter values for early ranks lie outside our constraints.

The mixture model, with our current implementation and constraints, is eliminated entirely from our experiments. For every rank and every fold there are some documents in the upper rank with zero clicks. This means any non-zero w risks predicting a negative probability in the lower position. Despite the fact that none of our probabilities are zero, due to smoothing, the w values for the mixture model were severely limited, making it effectively equivalent to the baseline model.

The examination model was able to make some adjustments, with weights for flip-types 1 to 9 being: 1.035, 1.055, 1.090, 1.085, 1.070, 1.030, 1.020, 1.010 and 1.005. These are adjustments of up to 9% in click probability. However, the adjustment for rank 1 is less than the adjustment for rank 2, because there are training examples with high click probability in rank 2. If the weight were higher, then those training examples would be pushed out of bounds when predicting the flip to rank 1. Specifically, in this training fold there was a rank-2 document (an outlier) with $1.035^{-1} = 0.966$ probability of click. Any higher weight would have pushed

Model	Cross Entropy
Best Possible	0.141 ± 0.0055
Cascade	0.225 ± 0.0052
Logistic	0.236 ± 0.0063
Examination	0.247 ± 0.0072
Baseline	0.250 ± 0.0073

Table 2: Overall cross entropy results. Showing mean across 10 folds, plus or minus two standard deviations.

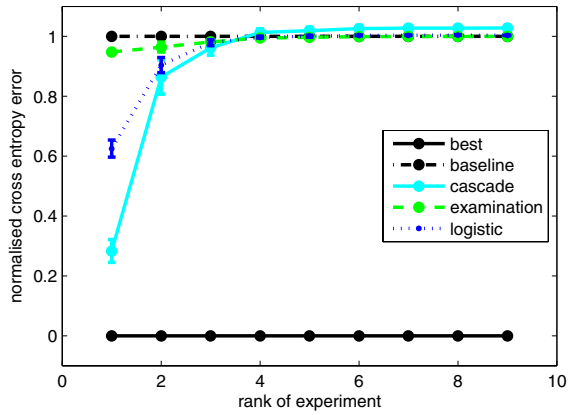

Figure 4: Overall results, by rank. Showing mean error across 10 folds and error bars for two standard deviations.

it out of bounds when estimating its rank-1 click probability. We do not know how the model was implemented in [9] or whether bounds checking is such an issue in ad click prediction, where probabilities of click would tend to be lower than for organic search results.

Although it is possible to allow a full range of weights, and add a hack that puts out-of-bounds probabilities back within bounds, we thought the logistic model is already a nice example of a model that stays within bounds. Also, when we performed some experiments with such a hack, the mixture and examination models performed no better than the logistic model.

Therefore we simply note that our initial simple models had constraint problems: upper documents with no clicks in the case of the mixture model and lower documents with high probability of clicks in the case of the examination model. We did not have constraint problems with the logistic or cascade models.

5.2 Results and Analysis

Overall results are presented in Table 2. Cascade is the best model, superior to the logistic model. The examination model makes barely any adjustment over the baseline. We also calculate a 'best possible' cross entropy, by using the test-set BA counts as our prediction.

This is particularly interesting in our by-rank analysis in Figure 4, where we normalise the error such that the best possible method has error of 0 and the baseline has error of 1. The cascade model performs particularly well at early ranks, closing more than 70% of the gap between baseline

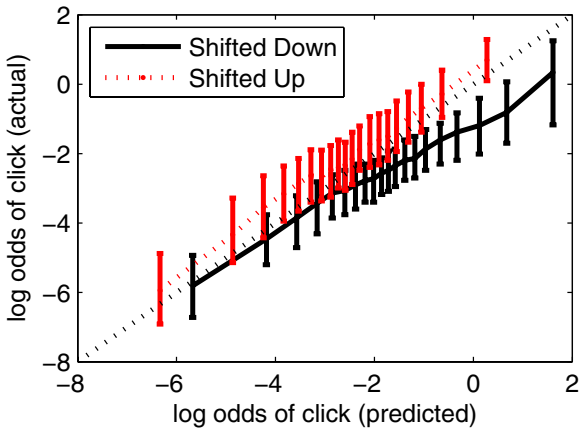

Figure 5: Baseline, predicted vs actual probability of click.

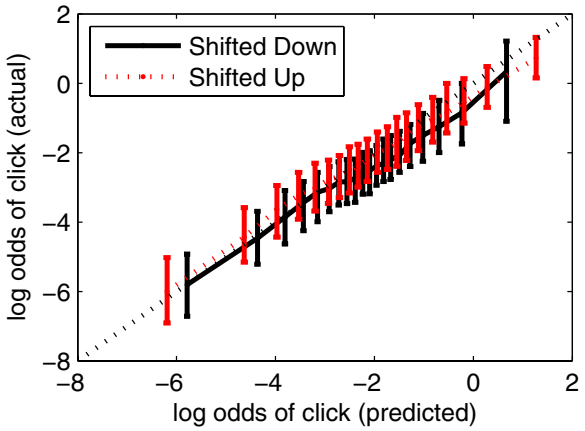

Figure 6: Logit, predicted vs actual probability of click.

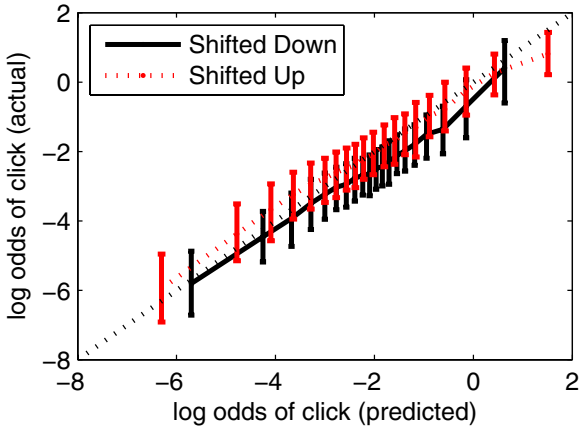

Figure 7: Cascade, predicted vs actual probability of click.

and optimal for rank 1. Although the cascade model performs worse, indeed worse than the baseline, for ranks 4 and beyond, we note that none of the models manage a significant difference from the baseline. We take this to mean that there is a small amount of presentation bias at lower ranks, and a large amount of variability because clicks are rare. This is consistent with Figure 3 where we see some noise but little systematic bias for many of the lower ranks. The median line tends to lie on the diagonal.

Figure 5 shows the sort of mistakes that are made under the baseline no-adjustment regime. This can be compared to Figures 6 and 7 where the median line has moved to the diagonal, albeit without becoming precisely diagonal or significantly reducing the error bars (which again indicate the spread of data with upper and lower quartiles).

Another interpretation of the per-rank results in Figure 4 is that the baseline model is very good in lower ranks, so the learning approaches tend to make little or no adjustment. This means that we can perhaps gather click information with little or no bias from clicks in lower ranks. By contrast, the bias is very strong in early ranks, and our cascade model which takes into account the relevance of the document(s) above performs best. One explanation for this is that users view results lists in two modes. In one mode they view results from top to bottom and click as soon as they see a relevant one. In the other mode they look at all results, so position has a negligible effect. In that case our baseline model of no-bias would be a good predictor of behaviour.

6. OTHER RELATED WORK

Radlinski *et al.* [8] also performed flips and made estimates of relevance. However, that scheme was to perform flips systematically in the system, until it converges on a correct ranking. Our goal is to model position effects so that we can consume a WWW click log with no flips and estimate click probabilities. Also that study was intended to find the correct ordering amongst pairs of documents, which is useful in some situations, for example many machine learning-based rankers train from pairs [5]. Our models estimate probability of click which could be used in other applications such as web search ranking, ad ranking [9] and evaluation [3].

Position bias within a results list can be considered separately from the problem of which results to show. So a randomisation model such as [7] could be used to expose new search results to the public, but it is still necessary to correct for bias within the results list that is shown.

There are click models that we have left unexplored. For example, the models in [3, 4, 2]. However, by considering our baseline, AND model, OR model, logistic model and cascade model, we believe we have covered a variety of potential model types.

7. CONCLUSION

Within our dataset of over 100 thousand observation-types, the cascade is by far the most successful model. This is remarkable in that it makes no use of training data, and is applied parameter-free to the click observations. That said, it performs badly in lower ranks. Although it is not much worse than other models, and no model performs significantly better than the baseline. It is clear that the cascade model is best suited to explaining flips at or near rank 1.

We described some simple models: 'Mixture' under which clicks are relevance OR random, and 'Examination' under which clicks are relevance AND examination. However, these models did not fit well with our data. We can find documents at any rank with no clicks, which is evidence against the random click hypothesis. We can find documents in (for example) rank 3 with a click probability greater than 0.9, which is evidence against the examination hypothesis. Therefore our implementations of both models have issues with constraints. As an alternative we present a simple logistic model, which performs well. Even when we allowed the AND and OR models to have more extreme values of their weights, and corrected any out-of-bound predictions, they did not perform better than the logistic model.

The excellent performance of the cascade model in early ranks, and the unbeaten performance of the baseline model at lower ranks, suggest two modes of results viewing. To compare click levels of adjacent pairs, the recommendation based on our results would be to simply apply the cascade model to correct for presentation bias if the pair is in top ranks, and to use the clicks from other ranks in their uncorrected form.

It would clearly be possible to improve our models and add more parameters. For example, the cascade model contains an assumption about continuation, that if users do not click they continue down the ranking. This is clearly not true, some users will abandon the results list without clicking and without looking at all results. In fact, as users traverse the page, we may find that we lose many users due to clicks on a particularly good result, and we lose many users due to abandonment if there is a particularly bad result. Then, once users click, the current cascade model assumes they are gone, therefore we can never observe multiple clicks on the same list. The case of multiple clicks, which clearly does occur in reality, could be allowed under the cascade model if we gave clicking users some probability of returning to the results list.

Other potentially interesting future work would be to analyse by query type, perhaps navigational queries have different usage patterns from informational queries. We could extend our models to take a whole click log as input, and update estimates of click relevance on the fly. Finally, beyond the scope of our current study, we could consider extending our model beyond simply being based on clicks. If we considered other information in usage logs and other features of users, queries and documents, we may be able to predict the probability of click significantly more accurately. This could eliminate some of the 'noise' we see in our current dataset (Figure 3).

8. REFERENCES

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

[2] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Improving search engines by query clustering. In *JASIST to appear*, 2007.

[3] Georges Dupret, Vanessa Murdock, and Benjamin Piwowarski. Web search engine evaluation using

click-through data and a user model. In *Proceedings of the Workshop on Query Log Analysis (WWW)*, 2007.

[4] Georges Dupret, Benjamin Piwowarski, Carlos A. Hurtado, and Marcelo Mendoza. A statistical model of query log generation. In *String Processing and Information Retrieval, 13th International Conference, SPIRE 2006*, pages 217–228, 2006.

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

[6] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM Press.

[7] Sandeep Pandey, Sourashis Roy, Christopher Olston, Junghoo Cho, and Soumen Chakrabarti. Shuffling a stacked deck: the case for partially randomized ranking of search engine results. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 781–792. VLDB Endowment, 2005.

[8] F. Radlinski and T. Joachims. Minimally invasive randomization for collecting unbiased preferences from clickthrough logs. In *Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1406–1412, 2006.

[9] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 521–530, New York, NY, USA, 2007. ACM Press.

APPENDIX

A. CLICK-BOTH EVENTS IN EVALUATION

In our evaluation we assign click-both events a zero probability. We now show that this does not disadvantage methods that independently assign a probability c_A of clicking A and c_B of clicking B, so assign click-both a probability of $c_A\, c_B$.

Under the independence assumption, we can break the cross entropy of an event such as 'click A but not B' into two parts:

$$-\log(c_A(1-c_B)) = -\log c_A - \log(1-c_B)$$

We abbreviate these two terms as a and $\bar{b}$.

Now consider a test with 100 observations, 20 total clicks on B and 10 total clicks on A. Consistent with those numbers is a case with 5 click-both events such as this:

$$\begin{aligned} \mathit{CrossEnt} &= 0.75(\bar{a}+\bar{b}) + 0.15(\bar{a}+b) + 0.05(a+\bar{b}) + 0.05(a+b) \\ &= 0.9\bar{a} + 0.1a + 0.8\bar{b} + 0.2b \end{aligned}$$

However, equally consistent would be a set of events with no click-both events:

$$\begin{aligned} \mathit{CrossEnt} &= 0.70(\bar{a}+\bar{b}) + 0.20(\bar{a}+b) + 0.10(a+\bar{b}) + 0(a+b) \\ &= 0.9\bar{a} + 0.1a + 0.8\bar{b} + 0.2b \end{aligned}$$

Both scenarios are consistent with 20 B click and 10 A clicks over 100 observations, and both have the same cross-entropy error.

We conclude that we can evaluate with or without click-both events, and models with an independence assumption will have the same cross entropy error. Since the cascade model in its current form is incapable of predicting click-both, we choose to evaluate without click-both events.