Measuring Service-Level Learning Effects in Search Via Query-Randomized Experiments

Paul Musgrave
Amazon Search
Palo Alto, CA, USA
pamusgra@amazon.com

Cuize Han
Amazon Search
Palo Alto, CA, USA
cuize@amazon.com

Parth Gupta
Amazon Search
Palo Alto, CA, USA
guptpart@amazon.com

ABSTRACT

In order to determine the relevance of a given item to a query, most modern search ranking systems make use of features which aggregate prior user behavior for that item and query (e.g. click rate). For practical reasons, when running A/B tests on ranking systems, these features are generally shared between all treatments. For the most common experiment designs, which randomize traffic by user or by session, this creates a pathway by which the behavior of units in one treatment can effect the outcomes for units in other treatments, violating the Stable Unit Treatment Value Assumption (SUTVA) and biasing measured outcomes. Moreover, for experiments targeting improvements to the behavior data available to such features (e.g. online exploration), this pathway is precisely the one we are trying to affect; if such changes occur identically in treatment and control, then they cannot be measured. To address this, we propose the use of experiments which instead randomize traffic based on the search query. To validate our approach, we perform a pair of A/B tests on an explore-exploit framework in the Amazon search page: one under query randomization, and one under user randomization. In line with the theoretical predictions, we find that the query-randomized iteration is able to measure a statistically significant effect (+0.66% Purchases, p=0.001) where the user-randomized iteration does not (-0.02% Purchases, p=0.851).

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Evaluation of retrieval results**.

KEYWORDS

A/B testing; ranking evaluation; design of experiments; behavior feature; spillover effect

ACM Reference Format:
Paul Musgrave, Cuize Han, and Parth Gupta. 2023. Measuring Service-Level Learning Effects in Search Via Query-Randomized Experiments. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3592020

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3592020

1 INTRODUCTION

Online controlled experiments are widely used to evaluate changes to industrial information retrieval systems [5], with the goal of predicting the causal effect of switching to the new version on some set of downstream metrics, such as clicks or purchases. The most common designs for such experiments are A/B tests randomized on users (or sub-user units such as sessions or page-hits) [6, 14]. The causal effect of the change is then estimated by the mean difference in outcomes (e.g. sales) between treatment and control users.

While proper randomization of treatment and control groups ensures most of the conditions necessary for this to be a valid causal estimate, there is one key condition it does not: the "stable unit treatment value assumption" [12, 18]. This asserts that the outcome for each unit is independent of the treatment assignment of any other unit; here, that the outcome for each user is independent of which ranking function was used for any other user.

If we assume that the only effect of which ranking function is used for a user is the direct effect on that user's current and future actions (e.g. purchases), then this assumption holds. For many experiments on ranking functions, this is the effect we expect to be the largest, so such an assumption is a reasonable approximation. However, there can also be indirect effects of the exposure, which are not constrained to the exposed user. If exposing treatment leads the user to purchase a product they would not otherwise have purchased, this can in principle have many downstream effects - on the seller of the product, on people the user interacts with, on the unit economics of the product, etc. Through these indirect effects, the treatment assignment of a user can impact outcomes for many other future users, violating SUTVA - such effects are sometimes known as "spillover effects" [19]. Of interest to us here: a user's treatment assignment can also impact the future behavior of the information retrieval system.

1.1 Feature Spillover

Most modern industrial information retrieval systems incorporate features which aggregate prior user behavior [4]. These features can be very important signals - for instance, Agichtein et al. [1] find that their incorporation improve performance by as much as 31%. Especially important are features which record past behavior for (query, item) pairs (e.g. click rate of an item for a query).

However, the use of these features opens a pathway by which the search results for one user can influence outcomes for other users. The search results for user A determine which items they will engage with, which affects the behavioral features of those items, which affects the search results (and actions) of future users. For an A/B test randomized on users, these are spillover effects.

Unless deliberately designed otherwise, behavioral features will be shared between all arms of A/B tests - ranking in both control and treatment(s) will use a version of the feature which incorporates behavior from all experiment arms. In practice, there are significant obstacles to any design which partitions behavior by treatment - beyond the engineering and infrastructure costs of maintaining multiple versions of the features, this reduces the amount of behavioral data available to each version (by e.g. 50% if experimenting of 50% of traffic), incurring a likely prohibitive opportunity cost. Then, these spillover effects will not just be between units within the same treatment, but across treatments, biasing measurement of the net effect of the intervention.

Beyond avoiding bias, in some cases these indirect effects may be what we are interested in measuring. Consider an experiment which "explores" in the explore-exploit sense, i.e. shows novel items in ranking in order to learn about their relevance. These items may not have the highest immediate expected relevance score, but by exploring them we can potentially improve the system's relevance in the long run. This value is obtained exactly through learning in the search relevance system - the primary means of which is behavioral features, in some form.

In this work, we propose that by changing the unit of randomization of the experiment from user to query, we can largely transform this learning effect from spillover to within-unit. This prevents it from biasing the evaluation of standard ranking interventions, and further allows us to measure the learning effect itself, as is needed to properly capture the value of exploration. To validate this approach, we run two A/B tests on an increase to exploration of new products in a large-scale e-commerce search system: one under query randomization, and the other under user randomization. We observe that the query-randomized test measures statistically significant increases in purchases, both overall (+0.66% p=0.001) and for new products (+2.21%, p=0.029), while the user-randomized text measures no significant change in either (-0.02% p=0.851 overall purchases, +0.48%, p=0.193 new product purchases). This demonstrates that feature spillover effects can be substantial, and that there can be significant benefits from exploration that cannot be measured by traditional user-randomized experiments, but can be measured with query-randomized experiments.

2 RELATED WORK

A similar measurement problem arises for producer-side effects in two-sided marketplaces. In e-commerce systems each purchase involves both a seller and a customer; in web-search or newsfeeds, each engagement involves both a content producer and a consumer. When performing a naive A/B test on users, we again cannot accurately measure indirect effects of the treatment on producers, for the same reasons discussed above. Ha-Thuk et al. [9] propose a counterfactual framework for this setting, based on ensuring that items in treatment are ranked where they would be if all sellers were treated, and likewise for control. This is likewise related to the problem of network bucket testing [3], of measuring effects through A/B tests in the context of a social network, where the treatment of one user may impact outcomes for their neighbors in the social graph. This is often treated by using cluster-based randomization [2, 20]. In this work, we identify a third party whose

User-randomization

Users	Queries					
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
C_1	C	C	C	C	C	C
C_2	T	T	T	T	T	T
C_3	T	T	T	T	T	T
C_4	C	C	C	C	C	C
C_5	T	T	T	T	T	T
C_6	C	C	C	C	C	C

Query-randomization

Users	Queries					
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
C_1	C	T	T	C	T	C
C_2	C	T	T	C	T	C
C_3	C	T	T	C	T	C
C_4	C	T	T	C	T	C
C_5	C	T	T	C	T	C
C_6	C	T	T	C	T	C

Figure 1: Treatment exposure in user- v.s. query-randomized experiments

behavior we may be causally impacting: the search ranking service itself. While the behavior of the service is in principle under our control, it nonetheless depends on the available information, which is changed when users are exposed to the treatment. For this reason, we need to consider the service both as an agent, and as a patient.

We take a use-case of treating cold-start under explore-exploit setting to measure the effect under A/B test frameworks. In related work on explore-exploit tradeoffs in online learning to rank, the setting is generally formulated as a contextual bandit problem, in which (1) a user submits a query, (2) the retrieval system presents a list of results, and (3) the user provides implicit feedback on the relevance of those results [11, 22]. To evaluate these techniques, prior work has focused on offline evaluation using simulated user data [11, 13, 22]. Offline evaluation of exploration policies is an important step to ensure they do not harm user experience by presenting poor rankings; however, when such a policy is deployed, we still wish to evaluate its empirical impact on long-term outcomes, requiring a means of online evaluation. This work discusses important considerations for doing so through A/B testing.

3 QUERY-RANDOMIZED EXPERIMENTS

We use the term "query-randomized experiment" to refer to an experiment where the treatment for a search is determined based on properties of the search query, rather than of the user issuing that query. Primarily, this property will be the search keywords. Figure 1 visualizes how treatment assignment differs between user- and query- randomized experiments.

Using query-randomized instead of user-randomized experiments allows us to measure consequences of exposure which occur within the query, i.e. the impact on immediate and future user engagement in the scope of that query. In particular, the influences on the query-item behavioral features are measured, since they occur within the unit of randomization and analysis rather than across it. We give two related applications where this property is

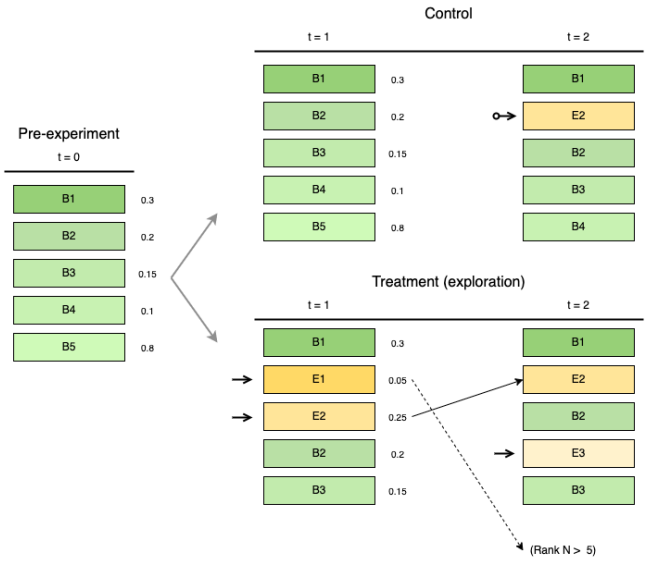

Figure 2: Rankings of items over time for a single query in an exploration experiment. At each time step, new items are explored in Treatment, and user behavior (e.g. click-through rate) is observed for all items. Starting from t=2, the explored items that were successful in previous timesteps have high behavioral features, so are also shown in Control.

useful: combating feature spillover, and measuring the value of exploration.

3.1 Combating Feature Spillover

Measuring within-query outcomes is helpful in preventing feature spillover from biasing the results of general ranking experiments. As discussed in Section 1.1, such spillover violates the stable unit treatment value assumption for user randomization, breaking theoretical guarantees that the A/B test will measure the causal effect of the intervention. In fact, the feature spillover will tend to predictably bias the results of the experiment towards zero, by making both arms more similar to each other.

Suppose that in a user-randomized experiment, a treatment (T) improves the relevance of the results for some query, leading to an increase in engagement (e.g. clicks). Since it is changing the results, it must elevate some items in the search results (and lower others). The newly elevated items in T will accumulate more user behavior, increasing their query-item behavioral features, and elevating their positions in control (C). But notice that the improvements to behavioral features (due to T) will then have improved the results of C more than those of T - by hypothesis, we were already showing those items highly in T. This leads experimental measurements to underestimate the value of any ranking improvement, and in general to be predictably biased towards zero.

Under query-randomized experiments, any impact to query-item behavior features is contained within the query (and hence treatment). While some behavior can still spill over through non-query-keyed behavioral features (e.g. total number of clicks for an item across all queries), these features tend to have much lower influence on ranking, so this substantially mitigates feature spillover.

3.2 Measuring the Value of Exploration

Measuring within-query outcomes is all the more critical when induced improvements in behavioral features are the main goal of the experiment. Consider an intervention which introduces exploration (in the explore-exploit sense) to ranking: presenting search results which do not necessarily have the highest immediate relevance, but whose relevance it is valuable to learn. This exploration can improve long-run payoffs, but most of the benefits will not be for the exposed user - they will rather be for all future searches where we leverage that learning. When the predominant learning mechanism is query-specific behavior features, these benefits will be for future searches for the same query. In a user randomized experiment, they will be distributed approximately equally between Control and Treatment, and hence unmeasurable - relevant items discovered by exploration at time t will be shown in both treatment and control from time $t + 1$ (see Figure 2 for an illustration). By contrast, in a query-randomized experiment, all within-query learning effects are contained within a single unit and so are correctly measured.

3.3 Considerations for Analysis

A key practical difference between query and user randomization is the distribution of traffic across these units. A much larger share of traffic is concentrated in a small fraction of queries (approximately following a power law [17]) than is concentrated in a similar fraction of users. This leads to higher variance in metrics such as purchases per query, compared to purchases per user.

However, much of the variance across units is predictable: most queries with high/low engagement during the experiment also had high/low engagement prior to the experiment. It is then especially important to control for pre-experiment behavior in the analysis. This makes it possible to obtain workable statistical power despite high variance in query frequency. To do so, we use an approach based on CUPED [7], training a model to predict in-experiment outcomes for a query from its pre-experiment metrics; the effect is then estimated by $(T_{\text{actual}} - T_{\text{predicted}}) - (C_{\text{actual}} - C_{\text{predicted}})$.

3.4 Tradeoffs of Query-Randomization

Symmetrically to how user-randomized experiments cannot measure effects outside of the user, the query-randomized experiments cannot measure effects outside of the query. In particular, they will not measure downstream effects on the treated user, such as their behavior in future searches. Depending on the intervention under experiment, these effects may be more or less significant than within-query effects from behavior spillover. They may also be more or less uniform: for many interventions, even if there are downstream user effects, they may be approximable as a fixed multiple of direct effects. Practitioners can choose an appropriate type of experiment based on offline predictions of each class of effect, or if practical, run under both randomization schemes.

4 EXPERIMENTS AND RESULTS

In this section, we present experimental framework to run A/B test under two settings: user and query randomization. For the A/B test, we consider a use-case of cold-start in product search and highlight the difference in results under both results.

4.1 Online Exploration for Cold-start

The cold-start problem in product search refers to the situation that new products often rank lower than they should, given their true relevance. A primary cause is the reliance on historical behavioral features in the search ranking system to signal relevance. New products lack behavioral data, and so rank low. The low position of the products in turn makes it harder for them to receive user feedback. This feedback loop leads to a persistent exploitation bias [21] in favor of old products. In the long term, this hurts user trust, as they may find it hard to discover new products even when specifically searching for them. It also creates a bad experience for sellers, as it takes a long period before their new products rank appropriately.

A natural idea for treating the cold-start problem is to break the feedback loop through exploration. Gupta et al. [8] uses a product level prior model to impute behavioral features for new products. Han et al. [10] further consider a query-product level model and a Bayesian framework where they estimate the behavioral features of new products through an informative prior based on side information, and then update to its posterior mean through new user feedback. They refer their method as Empirical Bayes (EB) online feature exploration. This solution mitigated the cold-start problem, increasing both new product purchases and overall purchases [10]. The experiments were done via usual user-randomized A/B tests. Although in this case, the treatment effect was large and immediate enough to overcome the service-learning spillover, it is still important to more accurately measure the effects of exploration, especially the long-term ones. For this work, we used their system as baseline and run a query-randomized experiment with a variant of the method (BayesUCB) described below as treatment.

4.1.1 BayesUCB. Following the notation in [10], the EB solution replaces the empirical click through rate feature for cold-start item-query pairs $\hat{p} = \frac{m}{n}$ with the Empirical Bayes posterior mean estimate $\hat{p}_{eb} = \frac{\alpha+m}{\alpha+\beta+n}$, where n is the number of times an item is impressed under a query, m is the corresponding number of clicks, and α, β are the parameters of the modelled Beta priors. If we assume the true click rate p of a query-item pair follows a Beta distribution $p \sim \text{Beta}(\alpha, \beta)$ and the click data is drawn from a Binomial distribution $m|n, p \sim \text{Bin}(n, p)$, then the posterior distribution of the click rate also follows a Beta distribution with updated parameters: $p|m, n \sim \text{Beta}(\alpha + m, \beta + n - m)$ which has expectation $\frac{\alpha+m}{\alpha+\beta+n}$. Through use of a prior, cold-start items that lack reliable behavior data can be shown higher, and converge to the empirical estimate as the item accumulates more clicks under the query.

One limitation of the EB method is that the exploration is only achieved through prior imputation and not through explicit "over-estimation"; such approaches (UCB, Thompson sampling) are well-known to be more efficient exploration strategies in multi-arm bandit settings [15]. Thus we propose the BayesUCB algorithm which update the behavioral feature to its posterior quantile instead of the posterior mean. As exact quantile calculation of a Beta distribution is not efficient, we instead use a sub-Gaussian posterior approximation. Since it is known [16] that the beta distribution $\text{Beta}(\alpha, \beta)$ is σ^2-sub-Gaussian for $\sigma^2 = \frac{1}{4(\alpha+\beta+1)}$, our BayesUCB estimation of the behavioral feature can be expressed as $\hat{p}_{ucb} = \frac{\alpha+m}{\alpha+\beta+n} + \sqrt{\frac{\log(1/\delta)}{2(\alpha+\beta+n+1)}}$, where $\delta \in (0, 1)$ controls the degree of exploration. As it increases exploration, this is a good test case for query-randomization better measuring the treatment effect.

4.2 User- vs. Query-Randomized Results

Under user-randomization, the results of this intervention appear insignificant: there is no statistically significant effect on overall purchases or purchases of new products.

Table 1: Results of user-randomized BayesUCB experiment

Metric	% Impact	95% CI	p-value
Purchases	-0.02%	(-0.21%, 0.17%)	0.851
New Product Purchases	0.48%	(-0.24%, 1.21%)	0.193

However, under query-randomization, the results are quite different: there is a statistically significant increase in both overall and new-product purchases.

Table 2: Results of query-randomized BayesUCB experiment

Metric	% Impact	95% CI	p-value
Purchases	0.66%	(0.28%, 1.03%)	0.001
New Product Purchases	2.21%	(0.23%, 4.19%)	0.029

The difference between these results demonstrates that indirect effects through improved query-item behavior features can be substantial. The query-randomized results demonstrate that the additional exploration in treatment improves results for treated queries over time. Yet in a traditional user-randomized experiment, we are empirically unable to measure any effect. This is just what is predicted by our theoretical arguments in sections 1.1 and 3.

5 REMARKS

In this paper, we exhibit a bias in the measurements from traditional A/B test designs as estimates the long-term effects of changes to information retrieval systems. This bias is a consequence of spillover effects that arise in this setting due to shared behavioral ranking features between treatments. In a broad range of ranking experiments, they attenuate the measured treatment effect; for certain experiments focused on improving the information available to the ranking system, i.e. exploration experiments, they prevent measurement of the intended benefit entirely.

We compare theoretically the measurements from user and query randomized experiments for ranking, and propose to use the latter in feature spillover scenarios. In this design, the experimental unit is a query property (such as keywords), rather than the user. Such experiments avoid spillover of changes to query-dependent features, and therefore, measure them correctly rather than being confounded by them. We validate this approach empirically by comparing query-randomized and user-randomized A/B tests results for an exploration experiment. While the user randomized experiment measures no significant changes in either overall or new-product purchases, the query-randomized experiment is able to detect statistically significant increases in both. This approach has been used to measure the value of exploration in Amazon search.

REFERENCES

[1] Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 19–26.

[2] Peter M Aronow and Joel A Middleton. 2013. A class of unbiased estimators of the average treatment effect in randomized experiments. *Journal of Causal Inference* 1, 1 (2013), 135–154.

[3] Lars Backstrom and Jon Kleinberg. 2011. Network bucket testing. In *Proceedings of the 20th international conference on World wide web*. 615–624.

[4] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the learning to rank challenge*. PMLR, 1–24.

[5] Alex Deng, Pavel Dmitriev, Somit Gupta, Ron Kohavi, Paul Raff, and Lukas Vermeer. 2017. A/B testing at scale: Accelerating software innovation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1395–1397.

[6] Alex Deng, Jiannan Lu, and Jonthan Litz. 2017. Trustworthy analysis of online A/B tests: Pitfalls, challenges and solutions. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 641–649.

[7] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *Proceedings of the sixth ACM international conference on Web search and data mining*. 123–132.

[8] Parth Gupta, Tommaso Dreossi, Jan Bakus, Yu-Hsiang Lin, and Vamsi Salaka. 2020. Treating cold start in product search by priors. In *Companion Proceedings of the Web Conference 2020*. 77–78.

[9] Viet Ha-Thuc, Avishek Dutta, Ren Mao, Matthew Wood, and Yunli Liu. 2020. A counterfactual framework for seller-side a/b testing on marketplaces. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2296.

[10] Cuize Han, Pablo Castells, Parth Gupta, Xu Xu, and Vamsi Salaka. 2022. Addressing Cold Start in Product Search via Empirical Bayes. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3141–3151.

[11] Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. 2011. Balancing exploration and exploitation in learning to rank online. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*. Springer, 251–263.

[12] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[13] Rolf Jagerman, Ilya Markov, and Maarten De Rijke. 2020. Safe exploration for optimizing contextual bandits. *ACM Transactions on Information Systems (TOIS)* 38, 3 (2020), 1–23.

[14] Ron Kohavi, Diane Tang, and Ya Xu. 2020. *Trustworthy online controlled experiments: A practical guide to a/b testing*. Cambridge University Press.

[15] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[16] Olivier Marchal and Julyan Arbel. 2017. On the sub-Gaussianity of the Beta and Dirichlet distributions. *Electronic Communications in Probability* 22, 54 (2017).

[17] Casper Petersen, Jakob Grue Simonsen, and Christina Lioma. 2016. Power law distributions in information retrieval. *ACM Transactions on Information Systems (TOIS)* 34, 2 (2016), 1–37.

[18] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.

[19] Betsy Sinclair, Margaret McConnell, and Donald P Green. 2012. Detecting spillover effects: Design and analysis of multilevel experiments. *American Journal of Political Science* 56, 4 (2012), 1055–1069.

[20] Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. 2013. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 329–337.

[21] Tao Yang, Chen Luo, Hanqing Lu, Parth Gupta, Bing Yin, and Qingyao Ai. 2022. Can clicks be both labels and features? Unbiased Behavior Feature Collection and Uncertainty-aware Learning to Rank. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 6–17.

[22] Dragomir Yankov, Pavel Berkhin, and Lihong Li. 2015. Evaluation of explore-exploit policies in multi-result ranking systems. *arXiv preprint arXiv:1504.07662* (2015).