Vertical Search Blending – A Real-world Counterfactual Dataset

Pavel Procházka, Matěj Kocián, Jakub Drdák, Jan Vršovský, Vladimír Kadlec, Jaroslav Kuchař
Seznam.cz, Czech Republic
{pavel.prochazka,matej.kocian,jakub.drdak,jan.vrsovsky,vladimir.kadlec,jaroslav.kuchar}@firma.seznam.cz

ABSTRACT

Blending of search results from several vertical sources became standard among web search engines. Similar scenarios appear in computational advertising, news recommendation, and other interactive systems. As such environments give only partial feedback, the evaluation of new policies conventionally requires expensive online A/B tests. Counterfactual approach is a promising alternative, nevertheless, it requires specific conditions for a valid off-policy evaluation. We release a large-scale, real-world vertical-blending dataset gathered by *Seznam.cz* web search engine. The dataset contains logged partial feedback with the corresponding propensity and is thus suited for counterfactual evaluation. We provide basic checks for validity and evaluate several learning methods.

KEYWORDS

Multi-armed Contextual bandit, Counterfactual dataset, Search engine Off-policy learning

ACM Reference Format:
Pavel Procházka, Matěj Kocián, Jakub Drdák, Jan Vršovský, Vladimír Kadlec, Jaroslav Kuchař. 2019. Vertical Search Blending – A Real-world Counterfactual Dataset. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331345

1 INTRODUCTION

One of the primary goals of a search engine is to serve users as relevant responses as possible. For some queries, utilizing specific sources of information – verticals – can improve this relevance, e.g. for location query a map could be shown, for song query a videoclip link could be included. The vertical does usually not cover the full query meaning, so the search engine should proceed carefully. A well adopted technique is to forward the query to each vertical, collect the responses, and then blend (fuse, aggregate) some of them into an enriched search engine result page (SERP), see [9].

While various approaches have been suggested [1], the multi-armed bandit (MAB) algorithms provide a simple and powerful model of this setting [2]. Depending on the context such as the query, the search engine decides which verticals to display (which arm to play), and then collects user responses such as clicks. This feedback is only partial, user behavior depends on what was displayed. The mission of the MAB algorithms is thus to balance between exploitation and exploration: on one hand to optimize immediate business metrics such as the click-through rate (CTR), on the other hand to collect sufficiently diverse feedback data to learn efficiently. One possible approach seen among the MAB algorithms is to model the reward for each arm and then for example select the arm with the best expected reward with some randomization (epsilon-greedy) or use Thompson (posterior) sampling [16]. Another approach is to view the arm selection as a classification problem, where the training relies on counterfactual risk minimization instead of traditional maximum likelihood approach [8].

The standard MAB model conventionally assumes that the user behavior (reward distribution) is stationary and is not affected by the MAB model. These assumptions are not strictly true for our vertical blending problem, and we must be aware of them.

Measuring the performance of a system under partial feedback presents a significant challenge. Conventional online A/B testing is expensive and difficult to scale, so there is a high demand for a *reliable* offline evaluation.

1.1 Goals and Contribution

This paper provides the following contribution:

(1) We release[1] a vertical search blending dataset collected by *Seznam.cz* search engine and provide its basic description.
(2) We formalize the *Seznam.cz* vertical search blending procedure. Following this formalism, we describe two levels of the MAB problem, that is, the item (vertical source) selection at a given fixed position (a per-position setup) and a complete SERP composition up to a given position.
(3) Using the dataset and the described logging procedure, propensities required in both counterfactual learning and evaluation are derived in the considered MAB-scenarios.
(4) We evaluate an offline estimate of some user feedback SERP business metrics for several baseline counterfactual learning methods trained using the per-position setup.
(5) We conduct sanity checks to address the estimate validity.

1.2 Related Work

MAB approaches have been well established in information retrieval and their usage occurs in numerous works related to vertical search blending. Usage in ranking was considered in [3, 6, 17], specifically for federated search in [2]. Counterfactual approach in the search engine (speller) is adopted in [12]. A partial feedback restriction appears also in recommender or advertising systems [5, 13, 14].

[1]Dataset and supporting code can be found at https://github.com/seznam/vertical-search-blending-dataset.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331345

In contrast to standard supervised learning setup, when dealing with partial feedback, it is difficult to get an unbiased estimate of online performance because one can observe feedback only for the actions taken by a logging policy. The correspondence between offline and online metrics is studied e.g. in [7]. One of the general methods of offline policy evaluation is the Inverse Propensity Scores estimator (IPS) [15, 19]. The IPS estimator often suffers from high variance. To deal with this issue, [4] suggested using additive control variates, which gave rise to the doubly robust estimator and [18] proposes to use multiplicative control variates, which they called self-normalized inverse propensity score estimator (SNIPS).

A dataset for counterfactual evaluation was published in [11]. The main difference is that our dataset comes from a search engine, compared to online advertising, with different structure and goals. Also, the dataset contains logged propensities for each individual position, and the described *Seznam.cz* logging procedure enables to calculate the logging propensity for the whole SERP up to an arbitrary position, so that several counterfactual setups are available.

2 DATASET

The dataset was produced by *Seznam.cz* search engine during five weeks in August/September 2018. There are over 8×10^7 SERPs blending up to 20 vertical sources and sorted organic search results. The proportions of the verticals are unbalanced. Due to privacy reasons and *Seznam.cz* business policies, the dataset sub-samples the total traffic served by *Seznam.cz* with an unspecified ratio. All personalized features were removed, queries, organic search and vertical sources are identified only by their numerical IDs.

Each (blended) SERP is represented as a single line in the dataset. One line is a tuple of 63 fields, describing the SERP composition and other related features. The first seven fields contain features common to the whole SERP (such as query id, time-stamp, responded verticals, etc.). Each of the 14 SERP positions is described by 4 fields (feedback, action propensity, action and domain). The SERP is formed by ten organic results and up to four vertical sources, that is, the SERP length varies from 10 to 14. If the SERP length is less than 14, the unused fields are filled with empty strings. The dataset was collected according to Algorithm 1.

The SERP features together with the sorted domains of the organic search results are known prior to the blending and thus form a context χ_i for the i-th example (SERP). The *action* is the result chosen by the logging policy to be inserted at given position k. This action is denoted[2] a_k (Algorithm 1 – line 6) and it depends on the SERP features χ and the decisions made up to the $(k-1)$-th position

$$\boldsymbol{b}_{k-1} \coloneqq (a_1, \ldots, a_{k-1}) .$$

The probability that the logging policy picks the action a_k at the k-th position can be written as

$$q_k^0 \coloneqq p_0 (a_k | a_1, \ldots, a_{k-1}, \chi) = p_0 (a_k | \boldsymbol{b}_{k-1}, \chi) \tag{1}$$

and this probability is recorded as the *action propensity* (Algorithm 1 – line 7). Finally, the *feedback* represents the partial user feedback serving for various rewards/metrics definitions, with values corresponding to *skip*, *click*, or *last click of the SERP*.

[2]Within this paper, the example (one SERP) is exclusively indexed by i and the position within SERP by k. If possible, we drop index i for better readability.

Algorithm 1: Per-position Vertical Blending

Data: Context χ, organic search results $\mathcal{G} = (g_1, \ldots, g_{10})$, the set of available verticals $\mathcal{A}$
Result: K, SERP $\boldsymbol{b}_K = (a_1, \ldots, a_K)$ and corresponding propensities $(q_1, \ldots, q_K)$

1 $k \leftarrow 0$ // number of filled positions
2 $j \leftarrow 0$ // number of used organic search results
3 $A \leftarrow \mathcal{A}$ // remaining verticals
4 **while** $j < 10$ **do**
5 $k \leftarrow k + 1$ // new position is being filled
6 select a_k from $A \cup \{g_{j+1}\}$ // action
7 $q_k \leftarrow p_0(a_k | a_1, \ldots, a_{k-1}, \chi)$ // propensity
8 **if** $a_k \in A$ **then**
9 $A \leftarrow A \setminus \{a_k\}$ // a_k cannot be blended again
// avoid verticals for the next 3 positions
10 **for** $m = 1$ **to** $\min(3, 10 - j)$ **do**
11 $k \leftarrow k + 1; \quad a_k \leftarrow g_{j+1}; \quad j \leftarrow j + 1; \quad q_k \leftarrow 1;$
12 **end**
13 **else**
14 $j \leftarrow j + 1$ // organic search result was used
15 **end**
16 **end**
17 $K \leftarrow k$ // SERP length

The logging policy implements conventional logistic regression for each action, where the position is considered as a feature (part of the context). Starting with the first position, CTR is predicted for all available actions, i.e. the first unused organic search result and all available vertical sources. Posterior distribution over CTR values is formed for each available action by Gaussian distribution with mean equal to the predicted CTR and variance given by an unspecified heuristic. The logging policy then samples from the posterior distributions corresponding to available actions and the action with the highest sampled CTR is chosen (i.e. Thompson sampling is performed in Algorithm 1 – line 6). The probability of this event q_k^0 is computed numerically (by doing 10 000 trials) and logged (Line 7 in Algorithm 1). If a non-organic vertical source is selected, it is removed from the actions available for later positions (Algorithm 1 – line 9) and the three following positions are automatically filled with organic search results (Algorithm 1 – lines 10-12). When organic search result is selected, the next organic search result is taken as possible action for the next position and the available verticals remain unchanged (Algorithm 1 – line 14). The described procedure is applied until all (ten) organic search results are placed in the SERP (Algorithm 1 – line 4).

3 COUNTERFACTUAL MODELS

In this section, we describe two setups showing how to use the dataset for counterfactual analysis. We first describe a simplified per-position model having limited prediction capabilities of the overall system performance (excepting the first position [13]). The second setup is used to evaluate metrics offline on the whole SERP, where the SERP is composed by the per-position trained models according to Algorithm 1.

3.1 Vertical Selection for a Given Fixed Position

This setup assumes that the logging policy is applied up to the $(k-1)$-th position according to Algorithm 1. The new model is requested to return the propensity of the action undertaken by the logging policy, according to this new model:

$$q^A_{i,k} = p_A\left(a_{i,k}|\boldsymbol{b}_{i,k-1}, \chi_i\right). \tag{2}$$

We can evaluate the SNIPS estimate [18] of the mean reward/metric function for the alternative model at the k-th position as

$$\hat{R}_k(A) = \frac{\frac{1}{N}\sum_{i=1}^N c_{i,k}\delta_{i,k}}{\frac{1}{N}\sum_{i=1}^N c_{i,k}}, \tag{3}$$

where $\delta_{i,k}$ stands for the corresponding metric/reward of the i-th SERP at k-th position, N is the number of examples and

$$c_k = \frac{p_A\left(a_k, \boldsymbol{b}_{k-1}|\chi\right)}{p_0\left(a_k, \boldsymbol{b}_{k-1}|\chi\right)} = \frac{p_A\left(a_k|\boldsymbol{b}_{k-1}, \chi\right)p_0\left(\boldsymbol{b}_{k-1}|\chi\right)}{p_0\left(a_k|\boldsymbol{b}_{k-1}, \chi\right)p_0\left(\boldsymbol{b}_{k-1}|\chi\right)} = \frac{q^A_k}{q^0_k}, \tag{4}$$

where we dropped the i-index and used definitions (1), (2).

3.2 Complete SERP

In this setup, an action chosen by a policy determines the whole SERP. However, policies are assumed to employ a per-position vertical selection policy and construct the SERP incrementally, as described in Algorithm 1, where the alternative per-position model is applied on lines 6 and 7. The propensities of the logging and alternative SERP blending policies are

$$p_0\left(\boldsymbol{b}_K|\chi\right) = \prod_{k=1}^K q^0_k, \quad p_A\left(\boldsymbol{b}_K|\chi\right) = \prod_{k=1}^K q^A_k \tag{5}$$

and the mean of a SERP feedback metric δ_i is SNIPS-estimated as

$$\hat{R}(A) = \frac{\frac{1}{N}\sum_{i=1}^N c_i\delta_i}{\frac{1}{N}\sum_{i=1}^N c_i}, \quad c_i = \frac{p_A\left(\boldsymbol{b}_{i,K}|\chi_i\right)}{p_0\left(\boldsymbol{b}_{i,K}|\chi_i\right)}. \tag{6}$$

Other counterfactual methods for training policies constructing the whole SERP have been proposed, e.g. the slate estimator [7]. Nevertheless, because of *Seznam.cz* blending infrastructure constraints, we restrict our attention to per-position blending based training and use this setup for evaluation only.

4 EXPERIMENTS

We evaluate SERP policies according to Section 3.2, where we consider the SERP length K as a parameter, that is, we truncate the SERP at the K-th position. This setup inherently includes the per-position model from Section 3.1 for the first position when $K = 1$. Evaluating the SNIPS estimate of SERP metrics in this way, the number of available actions (i.e. possible SERP compositions up to the K-th position) can be controlled, which can be useful to balance the offline SNIPS estimate validity and ability to express as complete SERP-metrics as possible.

Given the logged actions propensities for an alternative method, we can estimate mean of any metric that can be computed from the context and logged user feedback. Within this paper, we are interested in the SNIPS estimate (6) of the following SERP metrics:

- $\hat{R}^{\text{NDCG}}$: $\delta_i :=$ DCG/IDCG, where DCG $= \sum_{k=1}^K \frac{h_k}{\log(k+1)}$ with $h_k = 1$ if the *last click* was on position k and $h_k = 0$ otherwise. IDCG denotes ideal DCG with click on the first position ($h_1 = 1$), that is, IDCG $= \frac{1}{\log 2}$.
- $\hat{R}^{\text{CTR}}$: $\delta_i = 1$ if there was at least one click in SERP on arbitrary position and $\delta_i = 0$ otherwise. A significant property of this metric is that it is non-decreasing with K for fixed N, because there can be either more or equal clicks if some additional positions are considered. We will later use this property within our sanity check.
- $\hat{R}^{\text{VCTR}}$: $\delta_i = 1$ if at least one non-organic vertical was clicked and $\delta_i = 0$ otherwise.

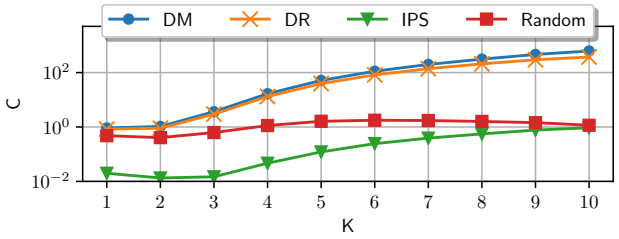

Figure 1: SNIPS denominator as a function of K for evaluated policies.

Policy	K=1			K=2		
	$\hat{R}^{\text{CTR}}$	$\hat{R}^{\text{NDCG}}$	$\hat{R}^{\text{VCTR}}$	$\hat{R}^{\text{CTR}}$	$\hat{R}^{\text{NDCG}}$	$\hat{R}^{\text{VCTR}}$
DM	0.435	0.382	0.012	0.523	0.435	0.014
DR	**0.445**	**0.393**	0.011	0.535	0.449	0.011
IPS	0.408	0.343	**0.177**	**0.639**	**0.509**	**0.264**
Random	0.404	0.351	0.045	0.474	0.377	0.087
Logging	0.433	0.379	0.020	0.526	0.434	0.054

Table 1: SNIPS estimates of the considered metrics for SERP length $K = 1$ and $K = 2$.

We used Vowpal Wabbit [10] for counterfactual learning of baseline methods [4], particularly direct method (DM), doubly robust (DR) and inverse propensity score (IPS). These methods are trained on the task of choosing a vertical for a given SERP position according to Section 3.1, where roughly one half of the dataset (data from August) was used for training.

The remaining part of the dataset, that is, data from September, was used for off-policy evaluation according to Section 3.2. Since all trained policies are deterministic, the alternative action propensity $p_A\left(\boldsymbol{b}_K|\chi\right)$ in (5) is 1 if the alternative action (SERP) is identical with the logged one and zero otherwise.

In addition to the trained methods, we also consider two reference policies for evaluation: first, a policy that chooses the action uniformly at random from the available actions, and second, the logging policy described in Section 2, where data and features beyond the provided dataset were used for training. The evaluated metrics are shown in Tables 1 and 2 for SERPS up to length $K = 4$.

A critical question of each offline evaluation in partial feedback systems is the estimate validity. We evaluate SNIPS denominator (6) that should be equal to 1 as far as the estimate assumptions

Policy	K=3			K=4		
	$\hat{R}^{\text{CTR}}$	$\hat{R}^{\text{NDCG}}$	$\hat{R}^{\text{VCTR}}$	$\hat{R}^{\text{CTR}}$	$\hat{R}^{\text{NDCG}}$	$\hat{R}^{\text{VCTR}}$
DM	0.490	0.342	0.001	0.518	0.355	0.000
DR	0.488	0.342	0.001	0.513	0.351	0.000
IPS	0.529	0.360	**0.076**	0.605	0.406	0.019
Random	0.511	0.359	0.042	0.529	0.363	0.031
Logging	**0.597**	**0.471**	0.074	**0.614**	**0.480**	**0.076**

Table 2: SNIPS estimates of the considered metrics for SERP length $K = 3$ and $K = 4$.

hold [11]. Distance of the denominator from 1 is reported as sanity check in [11] and it is shown as a function of position K in Figure 1 for the evaluated methods.

When the SNIPS denominator does not equal 1, we know that the assumptions are not fulfilled exactly, but the exact correspondence between the SNIPS estimate reliability and the denominator distance from 1 is not clear. In our setting, however, we can additionally use the fact that $\hat{R}^{\text{CTR}}$ should be non-decreasing in K. Comparing $\hat{R}^{\text{CTR}}$ for $K = 2$ and $K = 3$ in Tables 1 and 2, we see that $\hat{R}^{\text{CTR}}$ decreases for DM and DR, which is not consistent with the non-decreasing property of true CTR metric. We thus deduce that the SNIPS estimate is unreliable from the third position for DM and DR policies. On the other hand, the estimates on SERPs of lengths $K = 1, K = 2$ seem reasonable, because the performance is similar to the logging policy for DM and DR policies. Nevertheless, we cannot be sure that it is valid without an online test.

The normalization constant is far from 1 for the IPS estimator and we thus consider its predictions unreliable. This is probably caused by propensity over-fitting, since vertical sources are sparsely placed on the first position by the logging policy and if a vertical is blended on the first position, the logging propensity is very small.

The performance comparison on SERPs of lengths $K = 1, K = 2$ (see Table 1) reveals that DM-policy performs similarly to the logging policy, which is an expected result, because it is not randomized; on the other hand, the DM-policy uses less information for training. Finally, the logged propensity is utilized by DR the policy, since it achieves the best $\hat{R}^{\text{NDCG}}$ and $\hat{R}^{\text{CTR}}$.

5 CONCLUSIONS

We release a large-scale real-world dataset concerning blending web search verticals that is suitable for counterfactual analysis. We consider two counterfactual approaches over the dataset. The per-position model is considered to train models recognizing quality of individual vertical sources for a given context at fixed position. We are interested in determining which of the models is the best rather than in estimating the concrete value of reward for each model. In contrast with that, the second setup, considering partial/whole SERP, is not used for training the models. We only evaluate per-position learned models.

To address the off-policy estimate validity, we conduct sanity checks induced by properties of the evaluated metrics with respect to the dataset. Following these sanity checks, we determined setups where we consider the estimates to be reliable and in such cases evaluated the estimates of NDCG, CTR per whole SERP, and CTR per non-organic verticals.

The results point out that a direct implementation of basic counterfactual learning method (IPS) does not work well on the dataset probably due to low randomization of the logging policy. On the other hand, the logged propensities are utilized by DR learning method, according to the performance comparison between DR and DM logging policies. The dataset can be used for counterfactual learning, but issues like propensity over-fitting must be taken into account.

REFERENCES

[1] Jaime Arguello. 2017. Aggregated Search. *Found. Trends Inf. Retr.* 10, 5 (March 2017), 365–502.

[2] Jie at al. 2013. A Unified Search Federation System Based on Online User Feedback. In *Proceedings of the 19th ACM SIGKDD (KDD '13)*. ACM, New York, NY, USA, 1195–1203. https://doi.org/10.1145/2487575.2488198

[3] Vorobev at al. 2015. Gathering Additional Feedback on Search Results by Multi-Armed Bandits with Respect to Production Ranking. In *Proceedings of the 24th International Conference on WWW (WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1177–1187. https://doi.org/10.1145/2736277.2741104

[4] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th ICML (ICML'11)*. Omnipress, USA, 1097–1104.

[5] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 198–206.

[6] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2011. Contextual Bandits for Information Retrieval. In *NIPS 2011: Workshop on Bayesian Optimization, Experimental Design and Bandits: Theory and Applications*, Vol. 12.

[7] Thorsten Joachims and Adith Swaminathan. 2016. SIGIR Tutorial on Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement. In *Proceedings of the 39th International ACM SIGIR*. ACM, 1199–1201.

[8] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. 2018. Deep Learning with Logged Bandit Feedback. In *International Conference on Learning Representations* (iclr 2018 ed.).

[9] Oren Kurland and J. Shane Culpepper. 2018. Fusion in Information Retrieval: SIGIR 2018 Half-Day Tutorial. In *The 41st International ACM SIGIR (SIGIR '18)*. ACM, New York, NY, USA, 1383–1386.

[10] John Langford, Lihong Li, and Alexander Strehl. 2007. Vowpal wabbit open source project. (2007). https://github.com/VowpalWabbit/vowpal$_{\text{w}}$abbit/wiki

[11] Damien Lefortier, Adith Swaminathan, Xiaotao Gu, Thorsten Joachims, and Maarten de Rijke. 2016. Large-scale Validation of Counterfactual Learning Methods: A Test-Bed. *CoRR* abs/1612.00367 (2016). arXiv:1612.00367

[12] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on WWW*. ACM, 929–934.

[13] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on WWW*. ACM, 661–670.

[14] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 31–39.

[15] Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (04 1983), 41–55.

[16] Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. A Tutorial on Thompson Sampling. *Foundations and TrendsÂő in Machine Learning* 11, 1 (2018), 1–96. https://doi.org/10.1561/2200000070

[17] Marc Sloan and Jun Wang. 2012. Dynamical Information Retrieval Modelling: A Portfolio-armed Bandit Machine Approach. In *Proceedings of the 21st International Conference on WWW (WWW '12 Companion)*. ACM, New York, NY, USA, 603–604. https://doi.org/10.1145/2187980.2188148

[18] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems*. 3231–3239.

[19] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudík, John Langford, Damien Jose, and Imed Zitouni. 2016. Off-policy evaluation for slate recommendation. *CoRR* abs/1605.04812 (2016). arXiv:1605.04812 http://arxiv.org/abs/1605.04812