Underestimation Refinement: A General Enhancement Strategy for Exploration in Recommendation Systems

Yuhai Song*†, Lu Wang*, Haoming Dang, Weiwei Zhou, Jing Guan,
Xiwei Zhao, Changping Peng, Yongjun Bao, Jingping Shao
Business Growth BU, JD.com
songyuhai.syh@gmail.com,{wanglu241,danghaoming,zhouweiwei14}@jd.com
{guanjing,zhaoxiwei,pengchangping,baoyongjun,shaojingping}@jd.com

ABSTRACT

Click-through rate (CTR) prediction based on deep neural networks has made significant progress in recommendation systems. However, these methods often suffer from CTR underestimation due to insufficient impressions for long-tail items. When formalizing CTR prediction as a contextual bandit problem, exploration methods provide a natural solution addressing this issue. In this paper, we first benchmark state-of-the-art exploration methods in the recommendation system setting. We find that the combination of gradient-based uncertainty modeling and Thompson Sampling achieves a significant advantage. On the basis of the benchmark, we further propose a general enhancement strategy, Underestimation Refinement (UR), which explicitly incorporates the prior knowledge that insufficient impressions likely leads to CTR underestimation. This strategy is applicable to almost all the existing exploration methods. Experimental results validate UR's effectiveness, achieving consistent improvement across all baseline exploration methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommendation System, Contextual Bandit

ACM Reference Format:

Yuhai Song, Lu Wang, Haoming Dang, Weiwei Zhou, Jing Guan,, Xiwei Zhao, Changping Peng, Yongjun Bao, Jingping Shao. 2021. Underestimation Refinement: A General Enhancement Strategy for Exploration in Recommendation Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3462983

1 INTRODUCTION

Personalized recommendation systems have made great progress in recent years, benefiting from deep neural networks due to the flexibility to learn rich representations of user preference and item characteristics from historical interactions [7, 12, 17, 18, 25, 29]. Click-through rate (CTR) prediction is a classic task in recommendation systems, predicting the likelihood that the item will be clicked by a user in some context.

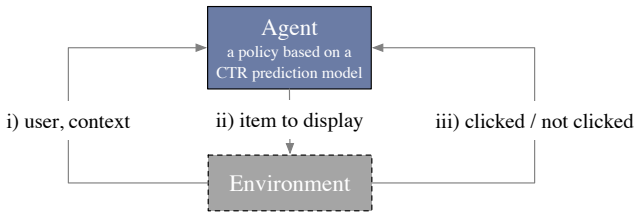

Figure 1: CTR prediction in a closed feedback loop

CTR prediction models are mainly trained in the manner of supervised learning, while only utilizing user feedback (clicked or not clicked) of the displayed items selected by previously deployed models. In the closed feedback training and serving loop, the continuously updated CTR model tends to focus on some small fraction of items with high estimated CTRs, whereas long-tail items hardly gain sufficient impressions. This leads to a CTR model performing sub-optimally on long-tail items. As a result, *underestimated items* (of which the CTRs are underestimated by chance) have few opportunities to be displayed, and are therefore treated unfairly — some impressions and clicks they deserve are lost, and furthermore it hurts the *social welfare* of the recommendation system.

In addition to supervised learning, the CTR prediction model in the closed feedback loop could also be formulated as a contextual multi-armed bandit problem (see Figure 1) — at every iteration, i) the agent (with a CTR model) sees a context (the user and some other context), ii) selects an action (displays an item), and iii) receives a reward (clicked or not clicked); the objective is to maximize the sum of the collected rewards (total number of clicks) [14, 15]. The agent has to balance between *exploiting* the current CTR model by selecting the item with the highest estimated CTR, and *exploring* potentially sub-optimal items to derive more information to update the CTR model effectively. In this framework, the underestimation problem is usually implicitly addressed when dealing with the exploration-exploitation trade-off dilemma (E&E).

Upper Confidence Bound (UCB) [3] and Thompson Sampling (TS) [20] are two commonly used algorithms to address E&E, both of which rely on maintaining a posterior distribution of the reward, that is a distribution of CTR in this work. In other words, we have to

*Equal contribution
†Work done while interning at JD.com

SIGIR '21, July 11–15, 2021, Virtual Event, Canada.

ACM ISBN 978-1-4503-8037-9/21/07.
https://doi.org/10.1145/3404835.3462983

model the uncertainty of CTR prediction. It is not trivial in practice since CTR prediction models are usually neural networks. In this paper, we compare a variety of well-established uncertainty modeling methods based on neural networks, including Bayesian Neural Networks [4], Gaussian Processes [26], Monte Carlo Dropout [10], and our proposed simplified versions of gradient-based uncertainty modeling methods [27, 28]. We benchmark these methods combined with TS and UCB as the downstream algorithms in the scenario of CTR prediction. Our proposed method Gradient TS is one of the best benchmark methods, and in the meanwhile has an outstanding advantage that it does not need to modify/retrain the neural network deploying in production.

In real applications of contextual bandits, especially working with deep neural networks, some theoretical results about regret bounds are no longer applicable [27, 28], indicating not to rely solely on theoretical models in practice. Motivated by this, we further propose an enhancement strategy, Underestimation Refinement (UR), which explicitly incorporates the prior knowledge that insufficient impressions likely lead to CTR underestimation. As such, UR gives the items which are possibly underestimated more impression opportunities. This strategy could be used to reinforce almost all existing exploration methods, including those in our benchmarks. Experimental results validate UR's effectiveness — by paying more attention to long-tail items, it achieves consistent improvement for all baseline exploration methods.

2 PROBLEM SETTING

A contextual multi-armed bandit problem can be formalized as below. At time t, the agent receives a context $x^{(t)} \in \mathbb{X}$, and needs to select an action $a^{(t)} \in \mathbb{A}$ from K available actions based on a policy $\mathcal{P} : \mathbb{X} \to \mathbb{A}$. Some reward $r^{(t)}$ will be generated and could be used to update the policy $\mathcal{P}$. Our objective is to maximize the cumulative reward $R = \sum_{t=1}^{T} r^{(t)}$.

In a typical case of a recommendation system, the action is to display an item from a candidate set, and the context can include some side information, such as the user features and the display position. After the item being displayed, we will get a reward $r^{(t)} = 1$ if the user clicks the item, and $r^{(t)} = 0$ otherwise. For CTR-based ranking policies, the new instance $(x^{(t)}, a^{(t)}, r^{(t)})$ will be used to update the CTR prediction model $f : \mathbb{X} \times \mathbb{A} \to [0, 1]$, which is usually a deep neural network in practice.

For contextual multi-armed bandit problems, there are three popular algorithms to tackle the exploration-exploitation dilemma: ϵ-greedy, Thompson Sampling (TS), and Upper Confidence Bound (UCB). Both TS and UCB are based on a posterior distribution over the score used for ranking, such as the predicted CTR in the recommendation system, which can be modeled by a Bayesian Neural Network for example. We will introduce how these uncertainty modeling methods can be applied in recommendation systems in the next section.

The ϵ-greedy algorithm greedily selects the "best" item (the highest predicted CTR) for a proportion of $1 - \epsilon$ of the impressions, and selects an item randomly with uniform probability for a proportion of ϵ. It will be reduced to pure exploration when $\epsilon = 1$ and to pure exploitation when $\epsilon = 0$.

Thompson Sampling maintains a (parameterized) posterior distribution of CTR for each item. At each iteration, it first scores each candidate item by drawing a sample from the posterior distribution, which could be formulated as:

$$s_{\mathrm{TS}}(x, a) \stackrel{\mathrm{def}}{=} \sigma(\hat{y}), \quad \hat{y} \sim \mathcal{N}(\mu(x, a), \lambda \cdot \Sigma(x, a)), \tag{1}$$

where $s_{\mathrm{TS}}(x, a) \in (0, 1)$ is the score of the item a with the context x, $\sigma(\cdot)$ is the sigmoid function, $\mu(x, a) \in \mathbb{R}$ is the predicted mean, $\Sigma(x, a) \in \mathbb{R}^+$ is the predicted variance and λ is a tunable parameter to balance the exploration-exploitation trade-off. We then display the item with the highest score, and the posterior distributions could be updated with the feedback.

UCB also maintains a (parameterized) posterior distribution of CTR for each item. Instead of sampling a random score, it scores each item deterministically as

$$s_{\mathrm{UCB}}(x, a) \stackrel{\mathrm{def}}{=} \sigma\left(\mu(x, a) + \lambda \cdot \sqrt{\Sigma(x, a)}\right), \tag{2}$$

where $s_{\mathrm{UCB}}(x, a)$ is the upper confidence bound of the item a with the context x.

3 UNCERTAINTY MODELING

As mentioned above, both TS and UCB need to maintain a (parameterized) posterior distribution over predictions, for which approximation is usually necessary especially when combined with neural networks. For the empirical issue of CTR prediction in recommendation systems, we mainly consider the neural network of transforming the ⟨item, context⟩ pair to a low-dimensional space via the bottom layers and then modeling the posterior distribution with an uncertainty module via the top layer. Four types of uncertainty modeling methods are discussed below including our proposed methods Gradient TS and Gradient UCB.

Bayesian Neural Networks. Bayesian Neural Networks are commonly used to model prediction uncertainty by introducing weight uncertainty [4]. We could simply add a Bayesian Neural Network layer as the top layer. The uncertainty in the weights of the top layer is encoded in a normal variational posterior distribution.

Gaussian Processes. Gaussian Processes are a generic supervised learning method and provide well-calibrated uncertainty for prediction [19]. We could add a Gaussian Process module (with a Bernoulli likelihood in the CTR prediction case for example) as the top layer, which is equivalent to a Gaussian Process with a kernel parameterized by a neural network. Then we could utilize an efficient form of stochastic variational inference, leveraging local kernel interpolation and inducing points [9, 26].

Monte Carlo Dropout. Dropout is one of the most popular regularization techniques for deep neural networks, where the output of each neuron is independently zeroed out with a certain probability at each forward pass [24]. There is a profound connection between dropout networks and approximate Bayesian inference [10]. Monte Carlo Dropout (MC Dropout) just averages the predictions of the neural network while keeping dropout active, and the result could be interpreted as a sample from a posterior distribution of CTR. Therefore, MC Dropout could be naturally combined with Thompson sampling. In practice, we only need to add a dropout module as the top layer.

Algorithm 1: Underestimation Refinement

Input: context x, candidate items $a_1, \ldots, a_m$, impression count of each item $c_1, \ldots, c_m$, (random) score function $s : \mathbb{X} \times \mathbb{A} \to \mathbb{R}$, score threshold $\tilde{s}$, count threshold $\tilde{c} > 0$, hyperparameters $\alpha > 0, \beta > 0$

for $i = 1, \cdots, m$ **do**
  Compute the score $s_i = s(x, a_i)$
  if $s_i \leq \tilde{s}$ *and* $c_i \leq \tilde{c}$ **then**
    $s_i \leftarrow s_i \cdot \left(1 + \frac{\alpha}{\sqrt{\beta + c_i}}\right)$
  end
end

Output: the item with the highest score

Gradient-based uncertainty modeling. Apart from these Bayesian approximation approaches mentioned above, gradients with respect to weights can also be utilized to characterize prediction uncertainty [2], which therefore could also be combined with TS or UCB, such as Neural UCB [28] and Neural TS [27]. To make them practical for recommendation systems, we propose a simplified version of Neural UCB termed Gradient UCB and a simplified version of Neural TS termed Gradient TS. Specifically, Gradient UCB scores each item as:

$$s_{\text{GUCB}} \stackrel{\text{def}}{=} \sigma\left(h(x,a) + \lambda \cdot \sqrt{\mathbf{g}(x,a)^\top \mathbf{g}(x,a)}\right), \tag{3}$$

where $h(x,a) := \sigma^{-1}(f(x,a))$ is the logit of CTR predicted by the deep neural network, and $\mathbf{g}(x,a)$ is the gradient vector of the last layer weights. Gradient TS scores each item by drawing a sample from the Gaussian distribution:

$$s_{\text{GTS}} \stackrel{\text{def}}{=} \sigma(\hat{y}), \quad \hat{y} \sim \mathcal{N}(h(x,a), \lambda \cdot \mathbf{g}(x,a)^\top \mathbf{g}(x,a)). \tag{4}$$

One of the advantages of Gradient TS and Gradient UCB is that they do not need to modify/retrain the neural network in production (such as adding an uncertainty modeling layer).

4 UNDERESTIMATION REFINEMENT

The contextual multi-armed bandit provides a theoretical framework addressing the exploration-exploitation dilemma. With some conditions satisfied, such as linear realizability (the expectation of the reward of each item is linear with respect to the item and context features), several algorithms prove to achieve $O(\sqrt{T})$ regret bounds, e.g., LinUCB/SupLinUCB [8, 15], and Thompson Sampling [1].

However, for real applications like CTR prediction in recommendation systems, when combined with neural networks and binary rewards, it is much more difficult to derive similar regret bounds [27]. In these complex situations, it is not advisable to rely entirely on theoretical models. Instead, we explicitly consider the prior knowledge that insufficient impressions likely leads to CTR underestimation, and propose a strategy to explore more deeply these candidate items which could be underestimated.

Our proposed Underestimation Refinement (UR) strategy is given in Algorithm 1. The idea is quite simple: if the item has few impressions and has a small estimated CTR, we increase (refine) its ranking score according to the current number of impressions. These original ranking scores could be derived from any exploration methods.

Algorithm 2: Online Recommendation System Simulator

Input: randomized impression dataset $\mathcal{D}$, initialization dataset $\mathcal{D}_{\text{init}}$, (random) score function $s : \mathbb{X} \times \mathbb{A} \to \mathbb{R}$ and corresponding learning method, size of partition $N > 0$

$\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{init}}$
Train s with $\mathcal{D}_{\text{train}}$
$r_{\text{all}} \leftarrow 0$
for *each N-size log entries partition* $\mathcal{D}_{part} \subseteq \mathcal{D}$ **do**
  Initialize $\mathcal{D}_{\text{simulator}} \leftarrow \emptyset$
  for $(x, a, r, A) \in \mathcal{D}_{part}$ **do**
    if $\arg\max_{a' \in A} s(x, a') == a$ **then**
      $\mathcal{D}_{\text{simulator}} \leftarrow \mathcal{D}_{\text{simulator}} \cup \{(x, a, r)\}$
      $r_{\text{all}} \leftarrow r_{\text{all}} + r$
    end
  end
  Remove the oldest $|\mathcal{D}_{\text{simulator}}|$ samples from $\mathcal{D}_{\text{train}}$
  $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{simulator}}$
  Train s with $\mathcal{D}_{\text{train}}$
end

Output: r_{all}, i.e., total reward (number of clicks)

In other words, UR is a universal enhancement strategy that could work with any existing exploration method.

5 EXPERIMENTS

In this section, we compare several practical exploration methods via an online recommendation system simulator based on a public dataset. Experimental results show the effectiveness of our exploration method Gradient TS and our enhancement strategy UR.

5.1 Experimental Setup

We use the Yahoo! front page news dataset (Yahoo! R6B) [16]. There are around 28 million lines of log data in chronological order, within 15 days of October 2011, of which each line contains a user feature $x \in \mathbb{X}$, a displayed item ID $a \in \mathbb{A}$, a binary label for the user's feedback $r \in \{0, 1\}$, and a candidate set of items $A \subseteq \mathbb{A}$. The displayed items were randomly sampled from the corresponding candidates, which allows us to simulate the online recommendation systems.

In our simulation, we initialize a deep model with the first 80,000 log entries and simulate a closed feedback loop with the remaining log entries, which is summarized in Algorithm 2. We iterate over log entries one by one. For each log entry, if the item in the candidate set with the largest score happens to be the one recorded as displayed in the log entry, we will collect the ⟨user, item, click⟩ tuple, i.e., (x, a, r) as a new sample, otherwise skip the log entry. We update the CTR prediction model every 80,000 log entries, and the training set consists of the newest 80,000 samples. We use log entries of the first day to tune hyperparameters for each exploration method and the strategy UR by grid search, and evaluate each method with log entries of the last 14 day.

We use the same neural network architecture across all exploration methods. We first embed the user features and the item ID

Table 1: Results (mean ± std) on the Yahoo! R6B dataset. The symbol •/○ indicates that the method with UR is significantly better/worse than the corresponding one without UR on the criteria based on independent t-tests at 90% significance level. The largest mean value of every column is in bold.

	w/o UR (baseline)		w/ UR (enhancement)	
	number of clicks	long-tail imp (%)	number of clicks	long-tail imp (%)
Random	25433.5 ± 82.0	**7.200 ± 0.036**	N/A	N/A
Greedy	39423.3 ± 1397.9	0.086 ± 0.016	41987.6 ± 1536.3•	0.456 ± 0.035•
ϵ-Greedy	43645.0 ± 1407.6	0.530 ± 0.026	45098.8 ± 1198.3•	0.738 ± 0.035•
MC Dropout	45907.2 ± 705.0	1.760 ± 0.106	46486.7 ± 530.0•	2.133 ± 0.125•
Gradient TS	**47033.7 ± 849.0**	0.748 ± 0.072	**47302.0 ± 823.5**	1.206 ± 0.108•
Gradient UCB	39046.8 ± 1241.0	0.087 ± 0.017	41977.6 ± 868.0•	0.507 ± 0.051•
BNN TS	42505.0 ± 866.4	0.724 ± 0.078	43383.8 ± 420.6•	1.372 ± 0.127•
BNN UCB	36940.0 ± 1216.9	0.191 ± 0.018	38070.8 ± 1263.7•	0.255 ± 0.031•
GP TS	42448.7 ± 1401.7	2.968 ± 0.251	42406.6 ± 927.8	**3.074 ± 0.289**
GP UCB	30024.8 ± 1134.4	0.081 ± 0.012	32302.8 ± 1203.2•	1.386 ± 0.202•

to 6-dimensional vectors respectively, and concatenate them to get a 12-dimensional vector, which is then fed to a fully-connected network (16-8-1) to predict CTRs. For exploration methods, the last layer is replaced by an uncertainty modeling layer. We compare Bayesian Neural Networks (BNN), Gaussian Processes (GP), Monte Carlo Dropout (MC Dropout), and our proposed gradient based methods (Gradient TS and Gradient UCB). Apart from the uncertainty-based exploration methods mentioned above, we also compare the Random strategy (pure exploration), the Greedy strategy (pure exploitation) and ϵ-Greedy, where $\epsilon = 0.05$.

We take two metrics to evaluate exploration methods — the total number of clicks and the impression proportion of *long-tail items*. Long-tail items are defined as the items whose total number of impressions belongs to the smallest 20%. The total number of clicks is the total reward in simulation, i.e., r_{all} in Algorithm 2, indicating the *social welfare*. On the contrast, the impression proportion of long-tail items reflects the degree of exploration and also the *fairness* of recommendation systems to some extent.

5.2 Experimental Results

Experimental results (mean ± std) are reported in Table 1; each experiment is repeated 8 times. Experimental results are twofold. On the one hand, from the results of the baseline methods (see w/o UR columns in Table 1), we have three main observations. i) Exploration methods, especially these based on TS, perform better than both pure exploration (Random) and pure exploitation (Greedy), indicating that addressing E&E carefully is necessary for this online recommendation system. ii) Uncertainty modeling combined with TS is usually better than the one with UCB. This finding is consistent with previous work [9]. iii) Gradient TS achieves great success compared with other baseline methods. Considering the fact that Gradient TS does not require to modify/retrain the neural networks deploying in production, this result makes Gradient TS attractive and promising in practice.

On the other hand, the effectiveness of UR is investigated. First of all, three observations from the baseline results also hold for the enhanced case with UR. More importantly, we would like to show the enhancement effect of UR. Since all the baseline methods are tuned carefully, a natural question is whether these methods still have room for further exploration. The results with UR give a positive answer (in order to test the effect of UR, we conduct independent t-test at 90% significance level) — in terms of impression proportion of long-tail items, the enhanced methods with UR have a significant and consistent improvement on the degree of exploration across all baseline methods; moreover, in terms of the total number of clicks, the social welfare is further improved accordingly.

6 RELATED WORK

CTR prediction with deep neural networks. Deep neural networks have achieved great success for CTR prediction, in which most of existing works focus on designing network architectures for modeling users and items effectively [7, 18, 25, 29]. Recently, a surge of research efforts have been made to deal with the bias in CTR prediction, where one of the causes of bias is that the training data is observational rather than experimental [5]. [22] proposed an approach to handling selection bias in evaluation and training of recommendation systems based on propensity scoring; [6] applied the transfer learning methods (domain adaptation) to improve the model performance for long-tail items; [21] proposed a model-agnostic meta-learning (MAML) based method to train an unbiased model. Different from the above works, we utilize exploration methods to selectively display more long-tail and possibly underestimated items, which then could improve real CTR prediction performance rather than in an ideal unbiased setting.

Exploration for deep models in recommendation systems. [13] observed that evaluation of recommendation models based on closed feedback loop is not compatible with the situation of the random open loop, and pointed that exploration is able to alleviate closed loop effects. [23] proposed an exploration strategy based on ϵ-Greedy to address the cold start problem, which could be seen as a special case of our proposed UR when working with ϵ-Greedy. [9] proposed to combine Gaussian Processes parameterized by neural networks with exploration methods in online advertising systems; [11] proposed to model uncertainty with dropout in recommendation systems. As we show in the experiment section, both types of the above methods could be enhanced by UR.

7 CONCLUSION

In this work, we study the underestimation problem for CTR prediction in online recommendation systems. We benchmark a wide range of practical deep uncertainty modeling methods. Our proposed method Gradient TS is a simplified and practical version of the recently proposed theoretical exploration method [27], and shows great advantage not only in performance but also in its promising application prospects due to its simplicity. Moreover, we propose a universal enhancement strategy, Underestimation Refinement (UR), applicable to nearly all the existing exploration methods. The positive results of UR suggest that there still exists a lot of room for carefully-designed exploration strategies when addressing the exploration-exploitation trade-off in the case of deep uncertainty modeling.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2013. Further Optimal Regret Bounds for Thompson Sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*. 99–107.

[2] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *International Conference on Learning Representations*.

[3] Peter Auer. 2003. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, 3 (2003), 397–422.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *Proceedings of The 32nd International Conference on Machine Learning*. 1613–1622.

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *arXiv preprint arXiv:2010.03240* (2020).

[6] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 579–588.

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[8] Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. 2011. Contextual bandits with linear Payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Vol. 15. 208–214.

[9] Chao Du, Yifan Zeng, Zhifeng Gao, Shuo Yuan, Lining Gao, Ziyan Li, Xiaoqiang Zhu, Jian Xu, and Kun Gai. 2020. Exploration in Online Advertising Systems with Deep Uncertainty-Aware Learning. *arXiv preprint arXiv:2012.02298* (2020).

[10] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*. 1050–1059.

[11] Dalin Guo, Sofia Ira Ktena, Pranay Kumar Myana, Ferenc Huszar, Wenzhe Shi, Alykhan Tejani, Michael Kneier, and Sourav Das. 2020. Deep Bayesian Bandits: Exploring in Online Personalized Recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 456–461.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Amir H Jadidinejad, Craig Macdonald, and Iadh Ounis. 2020. Using Exploration to Alleviate Closed Loop Effects in Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2025–2028.

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

[15] 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 World Wide Web*. 661–670.

[16] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*. 297–306.

[17] Hu Liu, Jing Lu, Hao Yang, Xiwei Zhao, Sulong Xu, Hao Peng, Zehua Zhang, Wenjie Niu, Xiaokun Zhu, Yongjun Bao, et al. 2020. Category-Specific CNN for Visual-aware CTR Prediction at JD. com. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2686–2696.

[18] Hu Liu, Jing Lu, Xiwei Zhao, Sulong Xu, Hao Peng, Yutong Liu, Zehua Zhang, Jian Li, Junsheng Jin, Yongjun Bao, and Weipeng Yan. 2020. Kalman Filtering Attention for User Behavior Modeling in CTR Prediction. In *Advances in Neural Information Processing Systems*, Vol. 33.

[19] Carl Edward Rasmussen and Christopher K I Williams. 2005. *Gaussian Processes for Machine Learning*.

[20] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. *A Tutorial on Thompson Sampling*. Vol. 11. 1–96 pages.

[21] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 309–318.

[22] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33nd International Conference on Machine Learning*. 1670–1679.

[23] Parikshit Shah, Ming Yang, Sachidanand Alle, Adwait Ratnaparkhi, Ben Shahshahani, and Rohit Chandra. 2017. A practical exploration system for search advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1625–1631.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[25] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17*. 12:1–12:7.

[26] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. 2016. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, Vol. 29. 2586–2594.

[27] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. 2021. Neural Thompson Sampling. In *International Conference on Learning Representations*.

[28] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural Contextual Bandits with UCB-based Exploration. In *Proceedings of the 37th International Conference on Machine Learning*, Vol. 1. 11492–11502.

[29] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1059–1068.