Predictive Relevance Uncertainty for Recommendation Systems

Charul Paliwal
Amazon
Bengaluru, India
charup@amazon.com

Anirban Majumder
Amazon
Bengaluru, India
majumda@amazon.com

Sivaramakrishnan Kaveri
Amazon
Bengaluru, India
kavers@amazon.com

ABSTRACT

Click-through Rate (CTR) module is the foundation block of recommendation system and used for search, content selection, advertising, video streaming etc. CTR is modelled as a classification problem and extensive research is done to improve the CTR models. However, uncertainty method for these models are still an unexplored area. In this work we analyse popular uncertainty methods in the context of recommendation system. We found that popular uncertainty models fails to capture the predictive uncertainty of the CTR model that exist unique to the recommendation models and is not prevalent in the traditional classification models. We empirical show why a different uncertainty measure is required for the recommendation system CTR prediction models. We propose PRU (Predictive Relevance Uncertainty), a single forward pass uncertainty approach for a sample as a distance from the predictive relevance samples of the training data. We show the efficacy of the proposed predictive relevance uncertainty (PRU) on selective prediction. Further, we demonstrate the utility of the proposed framework on the downstream task of OOD detection and active learning while maintaining the latency of a single pass deterministic model.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

Recommendation Systems; Uncertainty Quantification; CTR Prediction

ACM Reference Format:
Charul Paliwal, Anirban Majumder, and Sivaramakrishnan Kaveri. 2024. Predictive Relevance Uncertainty for Recommendation Systems. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3589334.3645689

1 INTRODUCTION

Click-through Rate (CTR) prediction problem is ubiquitous in today's e-commerce, advertising, search and video streaming services. CTR models predict the likelihood of a user clicking on an item, be it a product, web article or an ad. The modeling involves two steps: in the first step (known as inference), one mines short-term and long-term history of the user and item metadata to rank all eligible items and surfaces the one (or few) with the highest estimated CTR. In the second step (referred to as training), the system collects appropriate feedback based on customer's interaction and retrains the model with the latest available information.

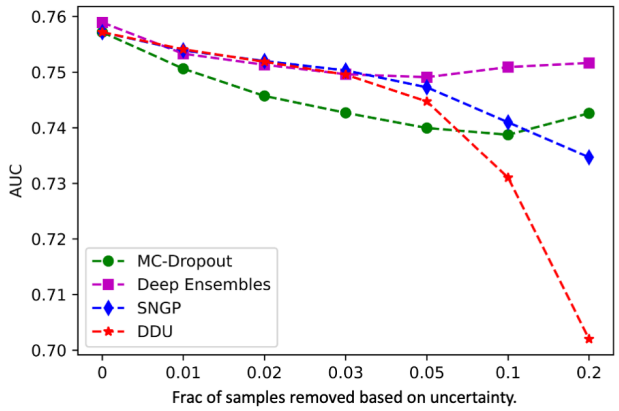

Figure 1: Uncertainty experiment on the Avazu dataset using MC-Dropout, DDU, SNGP and Deep Ensembles. We report AUC after discarding the least confident predictions from the test dataset.

CTR prediction is challenging for multiple reasons. First, due to relatively rare occurrence of positive samples, the training data is often insufficient to fit large parameter space of the model which leads to variability in its predictions. Moreover, dynamic user behavior, new customers and items, external events may require the model to predict on distribution of samples that was not observed during training time. Finally, CTR prediction involves out-of-distribution (OOD) samples by virtue of positional and presentation bias, content selection bias and user targeting. This results in inaccurate and over-confident estimation of CTR leading to a drop in its performance. One approach is to identify OOD samples via uncertainty quantification and filter them out at prediction time. Uncertainty captures the notion of model's confidence in accurately predicting the target label and therefore datapoints with high uncertainty are typically associated with erroneous predictions. Uncertainty estimates can benefit the model several ways e.g. it allows to make informed decisions and allocate resources wisely. Uncertainty is the deciding factor to trade exploration (recommending new items) with exploitation (recommending popular items) in multi-arm and contextual bandit algorithms. Finally, uncertainty estimates can guide active learning strategies, enabling the system to focus on uncertain predictions and acquire new data to improve the model.

WWW '24, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0171-9/24/05.
https://doi.org/10.1145/3589334.3645689

However unlike NLP, computer vision where much progress has been made on uncertainty-aware learning, reliable and efficient estimation of uncertainty is an open problem for recommender systems.

Click through rate estimation is often modeled as a classification problem where the goal is to classify an input into two classes: clicked and non-clicked. We expect the model to be inaccurate on highly uncertain points and hence if we were to remove these low-confidence predictions, the model performance would improve on the rest. In Figure 1, we plot the result of this experiment on the Avazu [5] dataset using state of the art uncertainty quantification techniques: DDU [20], SNGP [19], MC-Dropout [9] and Deep Ensembles [16]. We see a reverse trend where instead of improvement, the performance either stays flat or degrades significantly as we filter out least confident predictions. This suggests that uncertainty for CTR models is poorly explained by the current SOTA literature and needs deeper investigation. Recommendation system setting is different from the traditional classification setting as in the recommendation setup, neighbourhood of a datapoint has heterogeneous labels suggesting high degree of overlap between class-conditionals. Fig 2 is the t-SNE visualization of features from the last layer before the final classification layer and shows the class conditionals for two recommendation data-set. We tried to replicate the similar setup as used in the classification settings where the last layer feature embeddings are projected into lower dimensions while preserving both the local and global neighbourhood structure using t-SNE plots.

In this work, we investigate uncertainty estimation for CTR models and recommendation systems in general. We first provide insights on why recommendation problems need special treatment for uncertainty quantification. Guided by our insights, we propose PRU (Predictive relevance Uncertainty), a novel single pass deterministic uncertainty model that can be utilised over any existing CTR model for uncertainty estimation with no changes to the model architecture. Essentially, PRU is a meta-learning algorithm where we can plug in any model for CTR estimation and get accurate and efficient estimation of uncertainty. Experiments on benchmark datasets show superiority of PRU over state of the art baselines for uncertainty estimation in recommendation domains across variety of downstream tasks. We make the following contribution in this paper:

- We empirically study the SOTA uncertainty quantification for recommender systems and show that they fail to capture the true notion of predictive uncertainty.
- We present Predictive Relevance Uncertainty (PRU), a novel approach to quantify uncertainty for deep CTR prediction models, which can provide efficient uncertainty estimations along with the predictions and is compatible with any deep CTR models.
- We evaluate the effectiveness of PRU on selective prediction, out of distribution (OOD) detection and active learning. We perform a thorough and comprehensive set of experiments on three public benchmark datasets for CTR modeling comparing against several SOTA techniques for uncertainty estimation.

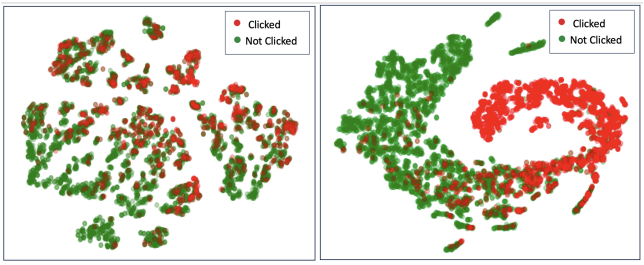

Figure 2: Feature distribution of clicked (in red) and non-clicked (in green) samples for Avazu (left) and MovieLens (right) dataset (best viewed in color).

- Our experimental result suggests that PRU achieves statistically significant +16% lift in selective prediction as compared to the strong uncertainty baselines. To highlight the accuracy of uncertainty quantification, we evaluate PRU on the downstream task of out of distribution (OOD) detection and active learning. Compared to strong baselines, PRU achieves +5% lift in OOD detection and +0.9% , +7% lift in active learning for different datasets.

2 RELATED WORKS

2.1 CTR prediction Problem

The purpose of CTR prediction is to estimate the probability that a user will click on an item. Although loosely used in the context of click, the definition is broad enough to capture any interaction such as purchase, video stream etc. One challenge in recommender systems is to find balance between *memorization* and *generalization*. Memorization refers to learning frequent co-occurrence of items from historical data whereas generalization refers the ability of the model to predict on unseen patterns. Cheng et al [4] proposed Wide&Deep which combines a DNN with a linear model and are trained jointly. The linear model encodes sparse features such as item-id, cross features between user and item that helps in memorization. On the other hand, the DNN component helps in generalizing to unseen patterns.

DeepFM [10] is another popular technique for CTR estimation that augments a traditional Factorization Machine (FM [22]) with a DNN component. Unlike Wide&Deep, DeepFM can be trained end to end without any feature engineering. DeepFM has further been extended to incorporate explicit feature interactions [18], adding a diversity loss in training objective to avoid overfititng [3] etc.

Zhou et al [32] presents Deep Interest Network (DIN) which adaptively learns user representation based on historical behavior with respect to certain ad. Despite learning contextual representation of users, DIN offers limited support for feature interaction. This was subsequently addressed in Deep&Cross networks (DCN [28, 29]).

2.2 Uncertainty

Popular methods of quantifying uncertainty includes Bayesian Neural Network [2], MC Dropout [9] and Deep Ensembles [17]. MC Dropout is a scalable alternative to the BNN models [9]. Deep ensemble aggregates collection of trained neural network to quantify

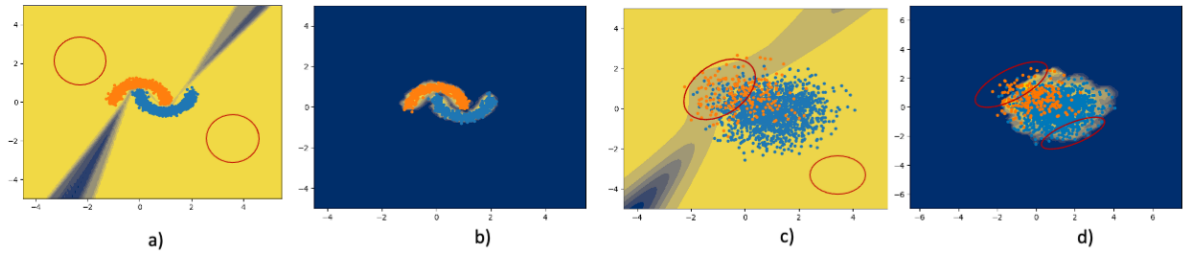

Figure 3: Failure modes for Model uncertainty (Deep ensembles) and Density/Distance aware uncertainty (DDU) in two moons dataset and Synthetic data-set with class overlap and imbalance. Blue denotes the high uncertainty region and yellow denotes the low uncertainty region. Red circle denotes the failure points. a) Model uncertainty for two moons dataset b) Density/Distance aware uncertainty (DDU) for two moons dataset. c) Model uncertainty for synthetic data d) Density/Distance aware uncertainty (DDU) for synthetic data.

the model uncertainty while MC dropout uses dropout enabled forward pass to quantify the model uncertainty. These framework can captures the model uncertainty where the model uncertainty is quantified by a sample distance from the decision boundary. Also, these are computationally expensive as it requires multiple forward passes to obtain the uncertainty measure. Therefore, single pass deterministic models are proposed where the uncertainty of a sample is quantified based on the distance/density from the training data[19, 20, 27].

Predictive uncertainty can be classified into two kinds [8, 12]: *aleatoric* or data uncertainty and *epistemic* or model uncertainty. While epistemic uncertainty can be reduced by collecting more data, aleatoric uncertainty, on the other hand requires instrumenting new features. Recent research has focused on disentangling uncertainty into these two components. Disentanglement is helpful for classification scenario where decision can be refused or delayed like as in a classifier with reject option [1] or reducing uncertainty in active learning scenario [24]. There has been recent work on disentangling uncertainty. For example, Kendall et al [13] define a model to estimate both aleatoric and epistemic uncertainty for regression and classification models. Matias et al [25] extends this techinque to richer class of models.

There has been extensive study of uncertainty models in the classification [15, 23], computer vision [14] and NLP [7, 21, 30] domain but remain unexplored in the recommendation literature. One recent work on recommendation system uncertainty is evaluated only for Movielens and Netflix dataset where user-item pairs are used to quantify different form of uncertainty [6]. Further, it doesn't quantify uncertainty based on the distance from the training data distribution. In this work, we propose PRU where the uncertainty can be quantified on any CTR dataset and model architecture.

3 UNCERTAINTY FOR RECOMMENDATION PROBLEMS

We argue that uncertainty should be higher for 1) OOD (Out of Distribution) samples i.e. for the samples that are away from the training data distribution. As the model has not seen this type of data samples at the training time, prediction on them cannot be trusted, 2) points near the decision boundary: as the model is confused about which class the samples belongs to, it leads to high variance in the model score. We check the notion of uncertainty for the traditional classification setting of two-moons dataset [1] and class overlap with imbalance synthetic dataset in Fig 3.

We plot the uncertainty surface for the two moons dataset in Fig 3a for model uncertainty using deep ensemble [17] and Fig 3b for density/distance aware uncertainty (DDU [20]). Yellow denotes the low uncertainty region and blue denotes the high uncertainty region. We show the failure modes for the uncertainty algorithm using red circles in Fig 3. For example in Fig 3a, we observe lower uncertainty in the red circle, whereas it is expected to be higher as this region (OOD sample region) is away from the training data distribution, therefore it is a failure mode for the uncertainty algorithm. Distance aware uncertainty captures the OOD detection problem better as compared to the model uncertainty as shown in Fig 3b. We observe high uncertainty for both OOD samples and decision boundary for density/distance aware uncertainty (DDU) in the traditional classification setting of two-moons dataset as shown in Fig 3b.

We then simulate the behavior of class overlap and imbalance observed in the recommendation system by using noise of 0.7 and class imbalance ratio of 0.2 on the two moons dataset. We observe this setting resembles the recommendation system setting as there is class overlap (neighbourhood of an instance is not homogeneously populated by instances of the same true class) and class imbalance (observe negative class (non clicks) dominating the positive class (clicks)) as shown in Fig 3c and 3d. In this setting model and distance aware uncertainty fails to capture the correct notion of uncertainty.

We observe high uncertainty for the minority class as the decision boundary for the model gets shifted towards the minority class in case of model uncertainty. Further, OOD samples (points away from the training data distribution) will be considered as low uncertainty samples as shown in Fig 3c. This notion of uncertainty can be harmful to the recommendation system, as the in distribution minority class samples are assigned higher uncertainty then the OOD samples. Using density/distance-aware uncertainty in this setting also fails to provide a true notion of uncertainty. This is because the uncertainty will be lowest in the class overlapping region,

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html

given that the training data distribution is concentrated mostly for both classes in that region. Furthermore, uncertainty will be lowest for the predictive relevance samples (samples where the model is confident in clicks/No clicks). Although density/distance aware uncertainty assign high uncertainty to the points away from the training data distribution. But it also assign high uncertainty to the predictive relevance samples where the model precision is high. This can again hurt the performance when utilizing uncertainty for the downstream task in recommendation system.

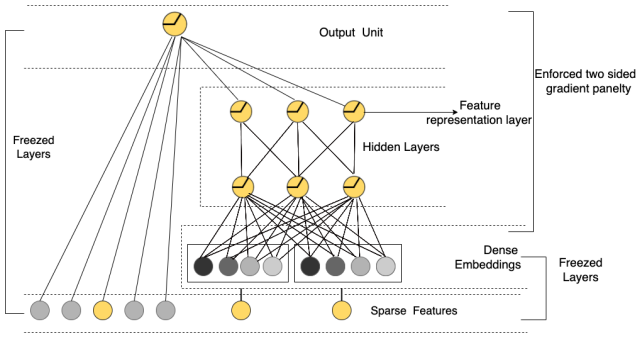

Figure 4: High-level architecture of the model and training steps.

4 PREDICTIVE RELEVANCE UNCERTAINTY

In this work we propose, Predictive Relevance Uncertainty (PRU) to estimate the uncertainty of deep CTR prediction models, which can provide efficient uncertainty quantification and is compatible with any deep CTR model. We define the notion of predictive uncertainty as a distance from the training samples that are highly predictive relevant. Highly predictive relevant are the samples where the model precision is high. In this way, the proposed framework PRU will capture the uncertainty of the OOD samples that are away from the training data distribution as well as points that have high variance due to the class overlap issue. At high level, PRU implements the following steps: 1) identify training samples that have high predictive relevance, 2) fit a density estimator on the regularized feature space of predictive relevant samples, 3) estimate uncertainty of a test sample by computing likelihood under the density estimator. Next we describe each of these steps in detail.

4.1 Predictive Relevant Instance Selection

We define predictive relevant samples as the ones where model precision is high, thereby, predicted score and the ground truth labels are close. Model will have lower loss on the predictive relevance samples. We fit a Gaussian Mixture Model (GMM) to cluster datapoints on their observed training loss i.e. points with similar loss are put in the same cluster. Samples belonging to the GMM component with the smallest mean can be selected as high predictive relevant samples. This modeling procedure however is class-agnostic, and reflect the same degree of loss for both the classes (click/no click). Given that in recommender systems, majority of the data is biased towards the negative class (i.e. no click), this approach will end up picking the predicted relevant samples primarily from the majority class. To avoid this behavior, we use class specific GMM to determine the predictive relevant samples for each class individually.

4.2 Density Estimation

A deep learning CTR model is typically composed of a feature transformation layer $h(x)$ that maps input instances to a hidden representation space and an output function $g(h(x))$ that maps the hidden representations to an output space. To utilize the feature space for distance awareness and density estimation, the hidden representation space is required to follow the bi-Lipschitz constraint so that distance in the latent space $d_h(h(x), h(x'))$ is bounded by the distance $d_x(x, x')$ in the input data manifold, for any inputs x, x'. More formally, we require $h(\cdot)$ to satisfy the bi-Lipschitz condition [19],

$$L * d_x(x, x') \leq d_h(h(x), h(x')) \leq U * d_x(x, x') \tag{1}$$

for positive and bounded constants $0 < L < 1 < U$. Here d_x and d_h denote any meaningful distance metric in the input and hidden representation space. The upper Lipschitz bound $d_h(h(x), h(x')) \leq U * d_x(x, x')$ is an important requirement to ensure robustness of the feature transformation layer which prevents the hidden representation $h(x)$ to be overly sensitive to perturbations in the input space. On the other hand, the lower Lipschitz bound $L * d_x(x, x') \leq d_h(h(x), h(x'))$ prevents the hidden representations to be unnecessarily invariant to large changes in the input space. Together, the bi-Lipschitz condition ensures that distances in the representation space are truthful representation of distances in the input space.

Algorithm 1 PRU Density Estimation

procedure TRAIN
  Train Baseline DNN model $p(\mathbf{y}|\mathbf{f}_\theta(\mathbf{x})$ with $(\mathbf{X}, \mathbf{Y})$
  Get predictive relevance samples $\mathbf{X_{pr}} \subset \mathbf{X}$
  for each class c with samples $\mathbf{X_c} \subset \mathbf{X_{pr}}$ **do**
    compute feature representation $z(x) = h_\theta(x)$
    Compute mean μ_c:
    $\mu_c \leftarrow \frac{1}{|\mathbf{X_c}|} \sum_{\mathbf{x} \in \mathbf{X_c}} h_\theta(x)$
    Compute covariance Σ_c:
    $\Sigma_c \leftarrow \frac{1}{|\mathbf{X_c}|-1} \sum_{\mathbf{x} \in \mathbf{X_c}} (h_\theta(x) - \mu_c) \cdot (h_\theta(x) - \mu_c)^T$
    Compute Cluster weights π_c:
    $\pi_c \leftarrow \frac{|\mathbf{X_c}|}{|\mathbf{X_{pr}}|}$
    $q(z|y = c) \sim \mathcal{N}(\mu_c; \Sigma_c), q(y = c) = \pi_c$

Gradient penalty and spectral normalization are two techniques of enforce the bi-Lipschitz condition [26]. Spectral normalization is a simpler technique used in distance aware uncertainty frameworks. It is applied to hidden weights in order to enforce bi-Lipschitz smoothness in representations [19, 20] Techniques proposed in [19, 20] restrict the framework to ResNet [11] to ensure sensitivity to the change in input. To utilize the existing architectures, we use two side gradient penalty, regularising the Jacobian with respect to the input embedding of the model. Therefore, the proposed PRU framework requires no changes in the model architecture and thereby compatible to any CTR framework. Overall we use spectral normalization to ensure stabilized training and two-sided Jacobian regularisation to encourage sensitivity to the inputs as shown in

Fig 4. The reference figure is sourced from the Wide&Deep [4], as depicted in Fig. 4.

Post-training, we fit a Gaussian mixture model on the loss residuals i.e. $\log f(x)$ or $\log(1 - f(x))$ depending on the true label. The GMM returns m mixing components $\{(\mu_i, \Sigma_i)\}_{i=1\cdots m}$ and a vector of mixing coefficient π_k for each sample x_k. Datapoints are mapped to the cluster based on max probability in π_i and we pick the cluster with lowest mean as the relevant sample set $\mathbf{X_{pr}}$.

As a final step, we identify the positive and negative samples in $\mathbf{X_{pr}}$. For each class c, let $\mathbf{X_c} \subset \mathbf{X_{pr}}$ be the subset of relevant samples. We fit a multi-variate normal distribution $q(z \mid y = c) = \mathcal{N}(z \mid \mu_c, \Sigma_c)$ via maximum likelihood estimation. Algorithm 1 highlights the pseudocode of our algorithm.

4.3 Uncertainty Quantification using PRU

We quantify uncertainty by calculating the marginal likelihood of the hidden feature representation under density estimator on the regularized predictive relevance samples as $q(z) = \sum_{y=c} q(z|y = c) \cdot q(y = c)$

4.3.1 Epistemic and Aleatoric Uncertainty: PRU captures both epistemic and aleotoric uncertainty similar to other distance aware uncertainty frameworks [19, 27]. When a point is far from the training data distribution (epistemic uncertainty), PRU uncertainty will be high as the likelihood of belonging to any of the class will be low. When a point lies in class overlapping data distribution (aleotoric uncertainty), PRU uncertainty will be high as these samples will be far from the predicted relevance samples of the classes.

4.3.2 Computational Complexity. : Let N_0 denote the number of PRU samples for class 0, N_1 represent the number of PRU samples for class 1, and D indicate the dimensionality of the feature space. When training a GDA, the total computational cost is given by $N_0D^2 + N_1D^2 + D^3$. The computation of the covariance matrix per class incurs a complexity of $N_0D^2 + N_1D^2$, while the computation of the inverse and determinant of the covariance matrices through Cholesky decomposition requires D^3 per class. During inference, evaluating the density of a single point involves calculating the distance from class means (D) and matrix-vector multiplications (D^2). Consequently, the overall inference complexity for evaluating the density of a single sample is D^2.

5 EXPERIMENTS

In this section, we will evaluate the efficacy of the proposed framework. First, we will assess the uncertainty estimates of the proposed framework in the traditional CTR setting to determine the level of trust we can place in the model predictions. Then, we will evaluate how the proposed framework would perform if utilized for the downstream tasks of OOD detection and active learning. We evaluate the performance of the proposed Predictive Relevance Uncertainty (PRU) with the following SOTA uncertainty quantification methods.

- MC-Dropout[9] : MC-Dropout provides a distribution over predictions via a sequence of forward passes by dropout-enabled pre-trained network. We use 10 forward passes and compute the variance in predictions to get the uncertainty estimates.
- Deep Ensembles[17]: Deep Ensembles aggregates a collection of trained neural networks with different initialization. We use 5-ensemble models to compute the uncertainty estimates.
- Spectral-Normalized Neural Gaussian Process (SNGP) [19]: SNGP is a single-pass, deterministic model that encode predictive uncertainty via distance-awareness.
- Deep Deterministic Uncertainty (DDU) [20] : DDU first fit each class component by computing the empirical mean and covariance, of each class feature vectors. Then compute the likelihood of a test sample belonging to each class component to quantify confidence.

5.1 Relevance to Prediction Scores

We use the following Evaluation Metrics to evaluate the uncertainty estimates.

- Selective prediction : High uncertainty is linked with low prediction performance. Therefore, the obtained confidence scores are thresholded and model is only evaluated on low uncertainty samples. Since we are filtering high uncertainty samples, it is expected that on the remaining samples, the model performance metric will improve. We report the increase/ decrease in AUC and PRAUC after filtering the high uncertainty samples.
- Latency: We report the time in (ms) to compute the uncertainty for 1k samples (ms/1k examples).

5.1.1 Datasets. We experiment with two open benchmark CTR datasets, 1) **MovieLens:** [2] Movielens data contains tagging record (user ID, movie ID, tag). Tag denotes the target and is assigned class 1 if user tags the movie. 2) **Avazu:**[3] This dataset contains 22 feature fields including user features and advertisement features for mobile advertisements and uses click records by users as the labels. We reuse the preprocessed data by [5] and follow the same settings on data splitting and preprocessing. Further details on creating the data split and preprocessing is provided by BARS [33].

5.1.2 Implementation: We use the two popular backbone model (DeepFM [10] and Wide&Deep [4]) based on FuxiCTR [33, 34]. We set the embedding dimension to 16, default MLP size to [200, 200], learning rate 0.01 to train the backbone model. We use the batch size of 8192 and 20480 for MovieLens and Avazu respectively. We use 10 forward passes for the MC-Dropout, 5 ensemble model for Deep Ensembles. We evaluate PRU for ($m = 3, 4, 5$) gaussian mixture components. Note that we fit the m GMM modes on the training loss to determine the predictive relevant samples.

5.1.3 Experimentation. We report AUC metric at different coverage in Table 1. AUC-95 denotes increase in AUC in bps if 5% of the most uncertain data points are removed from the evaluation set (1 bps = 0.01%) i.e. (AUC after filtering top uncertain 5% samples - AUC without filtering)× 10000. Positive value implies an improvement. We report the AUC at different coverage of 95, 90, 80 by filtering 5%, 10% and 20% of the most uncertain data based on the uncertainty algorithm. AUC metric can be biased to the majority data, therefore we also report the PRAUC-90. PRAUC-90 denotes

[2] https://grouplens.org/datasets/movielens/
[3] https://www.kaggle.com/c/avazu-ctr-prediction

Table 1: Selective prediction at different coverage. (*) indicates the statistically significant of PRU improvement over the best baseline (two-sided t-test with $p < 0.05$).

		DeepFM					Wide&Deep				
		AUC-95	AUC-90	AUC-80	PRAUC-90	latency	AUC-95	AUC-90	AUC-80	PRAUC-90	latency
Movielens	MC Dropout	25.63	38.47	49.21	53.90	70.23	23.97	34.65	41.34	46.05	67.87
	5-ensemble	7.69	-1.51	-37.19	-33.42	35.39	10.32	8.40	-12.38	-12.74	36.74
	DDU	-38.19	-41.81	-25.86	-83.09	7.22	-40.20	-49.33	-35.52	-103.34	6.99
	SNGP	-21.91	-37.20	-58.71	-65.71	18.20	-20.72	-35.06	-55.09	-56.55	15.54
	PRU (m=3)	28.84	47.86	66.97	77.01	7.36	29.55	48.59	66.53	77.36	7.02
	PRU (m=4)	**29.81***	**49.32***	**68.03***	**79.89***	7.43	**30.66***	**50.64***	**68.20***	**81.16***	6.98
	PRU (m=5)	29.28	48.45	67.49	78.23	7.23	30.01	49.41	67.32	78.90	7.12
Avazu	MC Dropout	-144.19	-134.92	-67.62	-774.35	107.69	-150.24	-146.55	-91.82	-805.23	103.35
	5-ensemble	-58.49	-39.50	-8.82	-359.42	64.68	-98.34	-80.25	-12.79	-544.92	60.45
	DDU	-115.72	-246.89	-534.00	-196.55	16.47	-122.66	-258.80	-546.25	-275.46	14.91
	SNGP	-62.63	-108.43	-172.99	-50.63	28.78	-64.55	-106.17	-155.34	-82.22	28.35
	PRU (m=3)	9.24	41.14	101.28	7.03	15.91	10.28	55.49	113.67	12.05	14.84
	PRU (m=4)	15.90	59.75	124.59	21.54	16.21	23.28	69.40	**132.30***	**15.06***	14.96
	PRU (m=5)	**23.28***	**73.34***	**146.16***	**30.19***	16.25	**27.95***	**72.54***	132.02	14.65	15.01

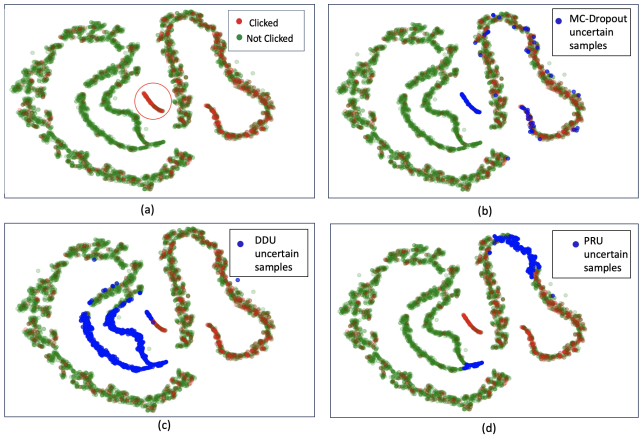

Figure 5: TSNE plots of the feature extractor layer from the baseline model for Avazu. a) denotes the distribution of features mapped to class: Not clicked (green) and class: Clicked (Red). b) High Uncertainty samples from MC dropout in blue. c) High Uncertainty samples from DDU in blue. d) High Uncertainty samples from PRU (proposed) in blue.

increase in PRAUC (area under precision-recall curve) in bps if 10% of the most uncertain data points are removed from the evaluation set (1 bps = 0.01%) i.e. (PRAUC after filtering top uncertain 10% samples - PRAUC without filtering)* 10000. Positive value implies an improvement. While the baseline algorithms observe a drop in majority of cases. PRU outperforms and improves the AUC on selective prediction for all the scenarios. The performance of PRU is consistent across modes for the Movielens and Avazu. As we keep on increasing the modes, the sampled data for density estimator decreases and if we fit higher modes, the density estimator lose out the useful training data distribution. MC Dropout improves on the Movielens dataset, but the performance of MC Dropout significantly drops for Avazu dataset. The drop is even higher for the PRAUC-90 in Avazu. To understand the trend, we plot the TSNE feature distribution of Avazu from the feature extractor layer in Fig 5. Fig 5a represents the class distribution for clicks/not clicked. We observe overlap and imbalance in the data. The samples inside the red circle denotes the higher confident samples for the minority class (click). MC-Dropout flags all the highly confident samples of the minority class as uncertain, therefore suffers the highest drop in the PRAUC. DDU flags confident samples from the minority as well as minority as uncertain, while PRU flags the majority of the points in the overlapping region as uncertain. Therefore, PRU is able to achieve the improved performance while maintaining the latency of a single pass backbone model.

5.2 OOD Detection

For the OOD detection task, we expect OOD data samples (data distribution on which the model is not trained) to have higher uncertainty compared to the ID samples (data distribution on which the model is trained). We label OOD samples as class 1 and ID samples as class 0, and report the AUC score based on the uncertainty estimates. If the uncertainty is high, the sample is expected to be OOD.

5.2.1 Datasets: We use category information to define the OOD samples. Datasets defined in section 5.1.1 either doesn't contain category information or it is anonymized. Therefore, we use the following datasets for the OOD detection task setup.

MovieLens-1M [4] :The data consists of 1 million movie ranking instances over thousands of movies and users. Each movie has features including its title, year of release, and genres. Titles and genres are lists of tokens. Each user has features including the user's ID, age, gender, and occupation. We transform ratings into binary

[4]1 http://www.grouplens.org/datasets/movielens

Table 2: Overall Performance comparison with uncertainty baselines. AUROC on the OOD/ID detection (Backbone model: Wide&Deep). (*) indicates the statistically significant improvement over the best baseline (two-sided t-test with $p < 0.05$).

	MovieLens-1M				Taobao Ad	
	Genre#1	Genre#1 $(e+\eta)$	Genre#3	Genre#3 $(e+\eta)$	Category_id	Category_id $(e+\eta)$
MC Dropout	0.5608	0.4633	0.5695	0.4744	0.5255	0.4630
5-ensemble	0.5122	0.4597	0.5353	0.4977	0.4207	0.4680
DDU	0.3432	0.4680	0.3480	0.4658	0.4976	0.6018
SNGP	0.5317	0.5667	0.4573	0.5887	0.5782	0.5950
PRU (m=3)	0.5710	**0.6192***	0.6028	0.6088	0.5958	0.6255
PRU (m=5)	**0.6170** *	0.6171	**0.6096** *	**0.6226** *	**0.6537***	**0.6346***

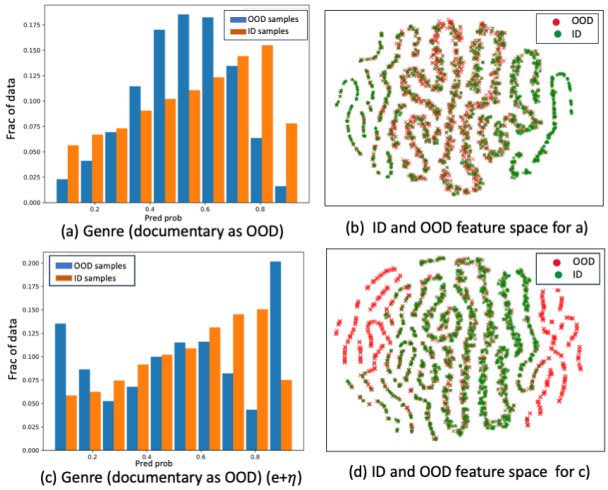

Figure 6: Feature distribution for the two settings used to create the OOD samples.

(The ratings at least 4 are turned into 1 and the others are turned into 0).

Taobao Display Ad Click [5] : It contains 1,140,000 users from the website of Taobao for 8 days of ad display / click logs (26 million records). Each ad can be seen as an item in our paper, with features including its ad ID, category ID, campaign ID, brand ID, Advertiser ID. Each user has 9 categorical attributes: user ID, Micro group ID, cms-group-id, gender, age, consumption grade, shopping depth, occupation, city level.

We use the same terminology and processing steps used in [31] to pre-process the MovieLens-1M and Taobao Ad datasets.

5.2.2 Experimentation: We use two settings to create the OOD samples. In the first setting, we hold out a subset of data based on category information (e.g., genre in MovieLens-1M and category-id in Taobao Ad). We choose the genre and categories based on their frequency in the data. For MovieLens-1M, we select the least frequent genre as Genre#1 OOD set, corresponding to the documentary genre, and pick the three least frequent genres as Genre#3 OOD set. For Taobao Ad, we pick the bottom 29 category-ids based on their frequency out of the total 139 categories, resulting in 1% of the total samples, denoted as Category_id OOD set. We don't train the model on these categories and use them as the OOD test set. We sample an equal number of samples from the rest of the data and use it as the ID test set. We then train the model on the remaining training data using a 90-10 train-val split.

In the second setting, we add noise (η) to the dense embedding (e) for 25% of the OOD test set. This results in more confident scores and maps the OOD samples away from the overall data distribution. Note that this is not an ideal setting, as the model might not experience this type of data distribution in the future. However, we obtain the OOD test samples that are mapped away from the data distribution. This second setting contains both realistic OOD samples, which the model has not seen and mapped mainly to the overlapping region, and also unrealistic OOD samples that are mapped away from the data distribution.

We plot the feature distribution of OOD/ID test set for Genre#1 MovieLens-1M dataset in Fig 6. The feature distribution for the first scenario is shown in Fig 6 (a,b). Since the model is not trained on OOD samples, we observe that OOD samples get mapped to the class overlapping region, and the model is not confident about the scores for the OOD samples compared to the ID samples. The feature distribution for the second scenario is shown in Fig 6 (c,d). We observe that the majority of the noisy embeddings $(e+\eta)$ get mapped away from the actual data distribution and obtain highly confident scores from the model.

We use Wide&Deep as the backbone network and report the AUROC performance for the OOD/ID detection in Table 2. For both settings, we observe that PRU is able to obtain better OOD detection performance compared to the other baseline methods. MC-Dropout and Deep ensemble suffer more in the presence of noisy embeddings, as they are not able to detect OOD samples with highly confident prediction scores. On the other hand, DDU and SNGP performance improve in the presence of noisy embeddings. However, the performance of DDU suffers mainly for the first setting, as DDU flags high confidence prediction ID samples as OOD because high confidence prediction samples lie in the low-density region for the recommendation system. PRU is able to handle both the cases and thereby improving the performance of OOD/ID detection in both the settings.

[5] https://tianchi.aliyun.com/dataset/56

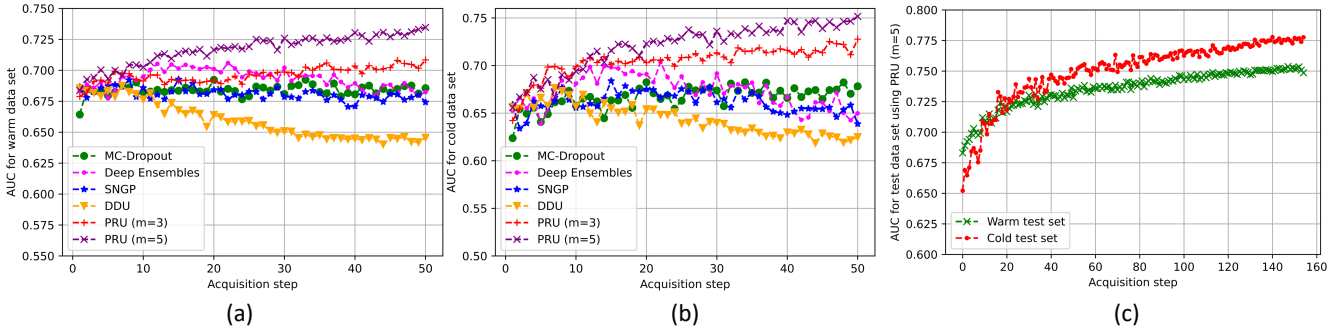

Figure 7: Active learning setup for the MovieLens-1M dataset. (a) and (b) denotes the AUC after each acquisition step of 1k samples on warm and cold test dataset. (c) AUC improvement using PRU (m=5) for 150 acquisition steps.

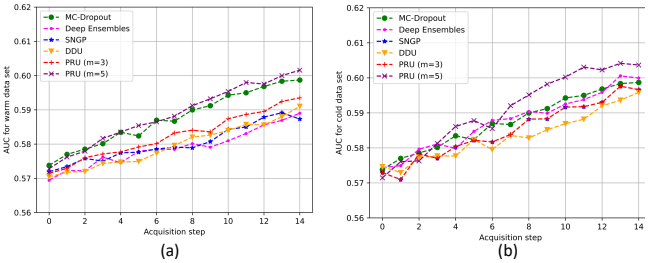

Figure 8: Active learning setup for the Taobao Ad dataset. (a) and (b) denotes the AUC after each acquisition step of 50k samples on warm and cold test dataset.

5.3 Active Learning

Data annotation is an expensive problem for deep learning models. The goal of active learning is to select a small portion of data for labelling from large pool of unlabelled data such that the model constructed with the labeled data has the optimal performance. The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. In uncertainty based active learning, the learner queries samples from the unlabelled set which is least confident (high uncertainty samples), presumably because such labels contain the most information about the downstream task. We follow the active learning setup in the recommendation setting, where at each acquisition step, top N unlabelled samples are picked based on uncertainty and the model is trained on the new generated training set. We start with an initial model trained with 10% of the randomly sampled training data and at each acquisition step push top N uncertain samples from the unlabelled set to the training data. We run it for all uncertainty techniques and report the results for MovieLens-1M and Taobao Ad dataset defined in section 5.2.1.

We divide the dataset into two groups, warm and cold based on their frequency. We use the same terminology and processing used in [31], to define the Warm items (The items whose number of labeled instances is larger than a threshold K). We use K of 200 and 2000 for Movielens-1M and Taobao Ad data. Cold item samples are sorted by timestamp and divided in four equal groups. We use the last (fourth) cold data set sorted based on timestamp as the cold test data set. We sample 10% of the Warm item data as warm test dataset. We use rest of the data as the training data pool. We use Wide&Deep as the backbone model and report the results on the warm and cold dataset. We use N=1k samples for MovieLens-1M and N=50k samples for Taobao Ad in each acquisition step which corresponds to 0.13% and 2% of the unlabelled pool of training data. We run 50 acquisition steps for MovieLens-1M and 15 acquisition steps for Taobao Ad (since each acquisition step of Taobao Ad is expensive) and report the AUC after each acquisition step in Fig 7 and 8. PRU clearly outperforms the baseline uncertainty algorithms. MC-dropout is competitive for the Taobao Ad warm test dataset but require 10 times inference time at each acquisition step. Also, note that we have to train 5 ensemble model at each acquisition step to obtain 5-ensemble uncertainty.

Further the gains are higher for the cold test data set. We plot the AUC improvement performance of PRU on warm and cold test set for extended 150 acquisition steps in Fig 7c. We observe that PRU achieve better performance for the cold test set as uncertainty can be higher for the cold samples as compared to the warm samples in the unlabelled training data. Also we obtain 95% of the baseline model performance (trained with all training data) with 96 acquisition step in MovieLens-1M that is around 13% of the unlabelled pool of data.

6 CONCLUSION

In this work, we proposed PRU (Predictive Relevance Uncertainty) for recommendation system. We showed that existing uncertainty estimation techniques suffers for recommendation problems because of the class overlap and class imbalance. We defined the notion of uncertainty for a sample as a distance from the predictive relevance sample of the training data. We showed that the proposed framework is able to correctly define uncertainty for the OOD region and the class overlapping region. We showed the efficacy of the proposed framework in the selective prediction and the downstream task of OOD detection and active learning.

REFERENCES

[1] P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *J. Mach. Learn. Res.*, 9:1823–1840, jun 2008.

[2] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[3] L. Chen and H. Shi. Dexdeepfm: Ensemble diversity enhanced extreme deep factorization machine model. *ACM Trans. Knowl. Discov. Data*, 16(5), mar 2022.

[4] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, page 7–10, New York, NY, USA, 2016. Association for Computing Machinery.

[5] W. Cheng, Y. Shen, and L. Huang. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3609–3616, 2020.

[6] V. Coscrato and D. Bridge. Estimating and evaluating the uncertainty of rating predictions and top-n recommendations in recommender systems. *ACM Transactions on Recommender Systems*, 1(2):1–34, 2023.

[7] V. Dragos. An ontological analysis of uncertainty in soft data. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1566–1573, 2013.

[8] Y. Gal et al. Uncertainty in deep learning. 2016.

[9] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[10] H. Guo, R. TANG, Y. Ye, Z. Li, and X. He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1725–1731, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016.

[12] E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[13] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc.

[15] A. Kurz, K. Hauser, H. A. Mehrtens, E. Krieghoff-Henning, A. Hekler, J. N. Kather, S. Fröhling, C. von Kalle, and T. J. Brinker. Uncertainty estimation in medical image classification: Systematic review. *JMIR Med Inform*, 10(8):e36427, Aug 2022.

[16] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

[17] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[18] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[19] J. Z. Liu, S. Padhy, J. Ren, Z. Lin, Y. Wen, G. Jerfel, Z. Nado, J. Snoek, D. Tran, and B. Lakshminarayanan. A simple approach to improve single-model deep uncertainty via distance-awareness. *J. Mach. Learn. Res.*, 24:42–1, 2023.

[20] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal. Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394, 2023.

[21] B. Qin, L. Wang, B. Hui, B. Li, X. Wei, B. Li, F. Huang, L. Si, M. Yang, and Y. Li. Sun: Exploring intrinsic uncertainties in text-to-sql parsers. In N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S.-H. Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5298–5308. International Committee on Computational Linguistics, 2022.

[22] S. Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[23] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3183–3193, Red Hook, NY, USA, 2018. Curran Associates Inc.

[24] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[25] M. Valdenegro-Toro and D. Saromo. A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement. *arXiv e-prints*, page arXiv:2204.09308, Apr. 2022.

[26] J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

[27] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

[28] R. Wang, B. Fu, G. Fu, and M. Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, ADKDD'17, New York, NY, USA, 2017. Association for Computing Machinery.

[29] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, WWW '21, page 1785–1797, New York, NY, USA, 2021. Association for Computing Machinery.

[30] X. Wei, H. Yu, Y. Hu, R. Weng, L. Xing, and W. Luo. Uncertainty-aware semantic augmentation for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2724–2735, Online, Nov. 2020. Association for Computational Linguistics.

[31] X. Zhao, Y. Ren, Y. Du, S. Zhang, and N. Wang. Improving item cold-start recommendation via model-agnostic conditional variational autoencoder. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2595–2600, New York, NY, USA, 2022. Association for Computing Machinery.

[32] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1059–1068, New York, NY, USA, 2018. Association for Computing Machinery.

[33] J. Zhu, Q. Dai, L. Su, R. Ma, J. Liu, G. Cai, X. Xiao, and R. Zhang. Bars: Towards open benchmarking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2912–2923, 2022.

[34] J. Zhu, J. Liu, S. Yang, Q. Zhang, and X. He. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2759–2769, 2021.

A APPENDIX

A.1 Additional Experiments

We experimented with DCNv2 [29] one of the recent and popular CTR backbone model and observe the similar findings in Table 3 and Table 4. We extended our experiment with Taobao Ad dataset defined in section 5.2.1 and report the findings in Table 5 and Table 6. We report AUC (area under ROC curve) metric at different coverage in below table. AUC-95 denotes the bps (increase/decrease) in AUC if 5% of the data samples are removed from the evaluation set based on high uncertainty i.e (AUC after filtering - AUC without filtering)* 10000. We report the AUC at different coverage of 95, 90, 80 by filtering 5%, 10% and 20% of the most uncertain data based on the uncertainty algorithm. AUC metric can be biased to the majority data, therefore we also report the PRAUC (area under precision-recall curve) -90. While the baseline uncertainty algorithms observe a drop in majority of cases. PRU outperforms and improves the AUC on selective prediction for all the scenarios.

Table 3: Selective prediction at different coverage on Movielens (DCNv2)

	Movielens			
	AUC-95	AUC-90	AUC-80	PRAUC-90
MC Dropout	21.17	38.18	47.26	45.32
5-ensemble	9.89	6.98	-11.53	-10.87
DDU	-40.40	-48.27	-67.81	-98.40
SNGP	-15.77	-25.95	-36.28	-38.56
PRU (m=3)	**24.54**	**52.35**	**79.51**	**84.70**
PRU (m=4)	23.05	51.22	77.86	83.60
PRU (m=5)	23.99	52.00	79.23	84.39

Table 4: Selective prediction at different coverage on Avazu (DCNv2)

	Avazu			
	AUC-95	AUC-90	AUC-80	PRAUC-90
MC Dropout	-187.49	-127.95	-76.23	-905.41
5-ensemble	-79.60	-45.86	-18.90	-178.89
DDU	-15.91	-23.40	-60.21	-31.97
SNGP	-11.70	-22.51	-37.58	-50.81
PRU (m=3)	10.51	19.87	15.81	23.63
PRU (m=4)	**12.60**	22.45	33.23	29.54
PRU (m=5)	12.50	**22.67**	**33.74**	**29.61**

Table 5: Selective prediction at different coverage on Taobao (DeepFM)

	DeepFM			
	AUC-95	AUC-90	AUC-80	PRAUC-90
MC Dropout	-35.19	-105.07	-209.13	-202.55
5-ensemble	-29.89	-36.98	-41.53	-107.87
DDU	-21.39	-17.50	-5.78	-18.40
SNGP	-13.42	-14.35	-14.03	16.14
PRU (m=3)	**22.41**	**26.94**	**28.69**	**34.60**
PRU (m=4)	20.45	24.23	25.80	33.60
PRU (m=5)	22.01	24.50	25.58	34.59

Table 6: Selective prediction at different coverage on Taobao (Wide&Deep)

	Wide&Deep			
	AUC-95	AUC-90	AUC-80	PRAUC-90
MC Dropout	-95.19	-165.07	-269.13	-215.41
5-ensemble	-39.60	-45.86	-58.90	-118.89
DDU	-24.10	-20.76	-10.47	-31.97
SNGP	-16.58	-25.65	-24.03	-20.81
PRU (m=3)	**24.10**	**28.56**	29.76	29.61
PRU (m=4)	22.35	27.66	**31.83**	**33.63**
PRU (m=5)	22.10	27.51	30.77	29.54