Proceedings of Machine Learning Research 204:1–19, 2023 Conformal and Probabilistic Prediction with Applications

Recommendation Systems with Distribution-Free Reliability Guarantees

Anastasios N. Angelopoulos* ANGELOPOULOS@BERKELEY.EDU
Karl Krauth* KARLK@BERKELEY.EDU
Stephen Bates STEPHENBATES@BERKELEY.EDU
Yixin Wang YIXINW@UMICH.EDU
Michael I. Jordan JORDAN@CS.BERKELEY.EDU
University of California, Berkeley
* *equal contribution*

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

When building recommendation systems, we seek to output a helpful set of items to the user. Under the hood, a ranking model predicts which of two candidate items is better, and we must distill these pairwise comparisons into the user-facing output. However, a learned ranking model is never perfect, so taking its predictions at face value gives no guarantee that the user-facing output is reliable. Building from a pre-trained ranking model, we show how to return a set of items that is rigorously guaranteed to contain mostly good items. Our procedure endows any ranking model with rigorous finite-sample control of the false discovery rate (FDR), regardless of the (unknown) data distribution. Moreover, our calibration algorithm enables the easy and principled integration of multiple objectives in recommender systems. As an example, we show how to optimize for recommendation diversity subject to a user-specified level of FDR control, circumventing the need to specify ad hoc weights of a diversity loss against an accuracy loss. Throughout, we focus on the problem of *learning to rank* a set of possible recommendations, evaluating our methods on the Yahoo! Learning to Rank and MSMarco datasets.

Keywords: Conformal prediction, distribution-free uncertainty quantification, learning to rank, recommendation systems

1. Introduction

The digitization of all manner of services has introduced recommendation systems into many aspects of our day-to-day lives. In particular, recommendation systems are now being applied to safety-critical domains such as making lifestyle recommendations to patients in healthcare (Hammer et al., 2015; Tran et al., 2021). It is therefore increasingly important that deployed recommender systems do not output recommendations devoid of uncertainty annotations. Meaningful recommendations should come with transparent and reliable statistical assessments. To date, the majority of deployed systems have fallen far short of this desideratum (Covington et al., 2016; Liu et al., 2017; Geyik et al., 2018).

Augmenting recommendation systems with internal tracking of statistical error rates would unlock new capabilities and applications. One such capability is the ability to enforce auxiliary constraints while still guaranteeing a baseline number of high-quality items in each slate of recommendations. For example, we could diversify slates whose quality we

are confident in, while leaving lower-confidence slates untouched. Furthermore, the strong guarantees provided by uncertainty quantification are a prerequisite for applying recommendation systems to safety-critical tasks such as medical diagnosis, where a misdiagnosis due to uncertain predictions can be fatal.

1.1. Our Goal

In this paper, we develop a method for quantifying uncertainty for the task of learning to rank (L2R). In particular, we consider the setting where we seek to return only items of some quality level, and can return sets of variable size. When returning variable-sized sets, a canonical notion of statistical error is the *false discovery rate* (FDR) (e.g., Efron, 2010). We will focus on this quantity in this work.

Formally, let $\{1, \ldots, K\}$ be the items under consideration, let $Y^* \subset \{1, \ldots, K\}$ be the ground-truth subset of the items that are of acceptable quality, and let $\widehat{S} \subset \{1, \ldots, K\}$ be the set of items returned by the algorithm. The false discovery rate of an algorithm is

$$\text{FDR} \;=\; \mathbb{E}\left[\frac{|\widehat{S} \setminus Y^*|}{\max(|\widehat{S}|, 1)}\right],$$

and we ask the algorithm to control this quantity at some user-specified level α.

In words, controlling for FDR means that the algorithm returns sets that are mostly items of high quality. When queries return large sets, it indicates that the model can confidently identify many high-quality items. Conversely, when queries return small sets, it indicates that the model cannot identify high-quality items with confidence. To control for FDR in recommendation systems, we propose a calibration algorithm that returns set of items with FDR guaranteed to be lower than a user-specified level with high probability; see Figure 1 for an example. This algorithm applies to any L2R model, including neural net models trained with LambdaRank (Burges, 2010). The algorithm comes with finite-sample statistical guarantees whatever the model, and these guarantees enable users to interact with recommendations with confidence.

Rigorously tracking statistical error rates opens the door to further performance improvements within a system. In this work, we will focus on the *diversity* of the items returned—a central goal when designing recommender systems. In particular, we will show how to return sets that are optimized for diversity *while maintaining finite-sample FDR guarantees*. That is, we will seek to approximately solve

$$\begin{aligned} \max_{\mathcal{D}} \quad & \mathbb{E}\Big[\text{Diversity}\big(\mathcal{D}(X)\big)\Big] \\ \text{s.t.} \quad & \mathbb{P}\Big(\text{FDR}\big(\mathcal{D}\big) > \alpha\Big) < \delta. \end{aligned}$$

Here, $\mathcal{D}$ is a model that maps an input to a set of returned items, $\widehat{S}$, and Diversity is a user-specified metric for the diversity of a set of items. Furthermore, with some abuse of notation, the FDR is taken *conditionally on the calibration data*, and we control it in high probability. See Section 2.3 for details. To our knowledge, this is the first such statistical result in the recommendation systems literature.

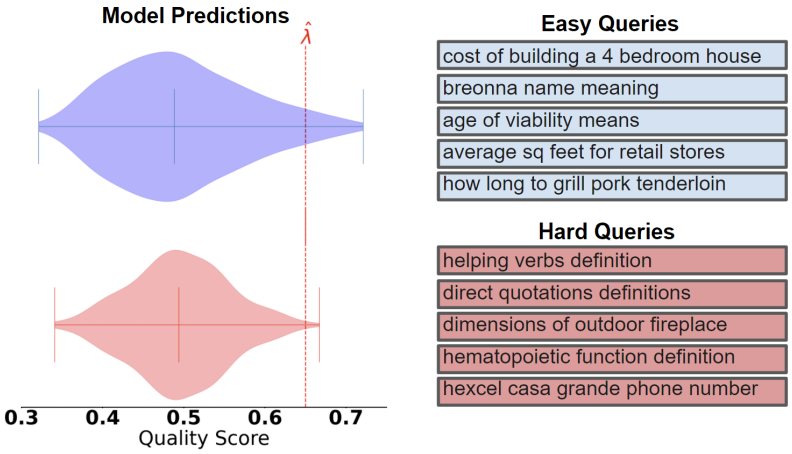

Figure 1: Examples of easy and hard queries. The violin plots show the distribution of document quality scores output by a LambdaRank model for the five queries on the right. For easy queries, the model can distinguish between the qualities of different documents, so the violin has a wide spread, and vice-versa for hard queries.

1.2. Related Work

Accurately quantifying model uncertainty has long been a desirable feature in information retrieval systems. Early work simply aimed to score the relevance of each item (Robertson and Jones, 1976; Koren et al., 2009). However, these methods are not calibrated, therefore their output scores can not be interpreted as probabilities. To remedy this issue one line of work models the problem through a Bayesian lens (Zhu et al., 2009; Freudenthaler et al., 2011; Gopalan et al., 2014; Wang et al., 2018). While these methods can improve upon their uncalibrated counterparts, they must be developed from scratch and require making strong assumptions about user interactions.

Our work aims to quantify the reliability of recommendations in the learning-to-rank setting. Recent work shows that neural learning-to-rank models suffer from poor calibration (Penha and Hauff, 2021), highlighting the importance of our goals.

Our approach is based on recent developments in *conformal prediction* (Vovk et al., 1999; Papadopoulos et al., 2002; Vovk et al., 2005; Angelopoulos and Bates, 2021) and *distribution-free uncertainty quantification* more broadly (Park et al., 2020; Bates et al., 2021a; Angelopoulos et al., 2021). This line of work provides a formal approach to defining set-valued statistical predictions and it has been applied to various learning tasks, such as distribution estimation (Vovk et al., 2020), causal inference (Lei and Candès, 2020; Jin et al., 2021), weakly-supervised data (Cauchois et al., 2022), survival analysis (Candès et al., 2021), design (Fannjiang et al., 2022), model cascades (Fisch et al., 2020; Schuster et al., 2021), the few-shot setting (Fisch et al., 2021), handling dependent data (Chernozhukov et al., 2018; Dunn et al., 2020), and handling or testing distribution shift (Tibshirani et al., 2019; Cauchois et al., 2020; Hu and Lei, 2020; Bates et al., 2021b; Gibbs and Candès, 2021; Vovk, 2021; Podkopaev and Ramdas, 2022). Most closely related to the present work, there

have been recent proposals applying conformal prediction to recommender systems. One line of work aims to apply conformal prediction to quantify the uncertainty in predicted ratings (Himabindu et al., 2018; Ayyaz et al., 2018), while another aims to quantify the uncertainty in a set of recommended items when only implicit feedback is available (Kagita et al., 2017; Penha and Hauff, 2021).

Diverging from these proposals, we go beyond conformal prediction and use the more general risk-control framework (Bates et al., 2021a; Angelopoulos et al., 2021). As a result, our work allows recommender systems to be optimized with respect to metrics other than accuracy while maintaining reliability guarantees. While we focus on diversity as a case study (Kunaver and Požrl, 2017), our work is applicable to the broader literature on alternative metrics, including reachability, serendipity and fairness (Singh and Joachims, 2018; Yao and Huang, 2017; Dean et al., 2020; Herlocker et al., 2004; Kaminskas and Bridge, 2016). As prior work has shown, focusing solely on accuracy can harm performance with respect to these alternate metrics (Adomavicius et al., 2013; Nguyen et al., 2014; Fleder and Hosanagar, 2009), underscoring the necessity of designing recommender systems that can be optimized with respect to multiple objectives.

1.3. Our Contribution

We introduce a method for calibrating learning-to-rank models to control the false discovery rate. The calibration procedure is supported by finite-sample statistical guarantees that apply to any model and dataset. Controlling the false discovery rate enables downstream tasks like optimizing recommendations for diversity; we explicitly extend our algorithm to produce sets of high diversity that are certified to control the false discovery rate. This concrete example also serves as a template for how to handle desiderata beyond diversity while providing statistical guarantees.

2. Methods

We begin by describing the learning-to-rank problem in recommendation systems. We then present a calibration algorithm for controlling the FDR in recommendation systems with provable guarantees.

2.1. Learning to Rank

The *learning-to-rank* problem refers to a task where we receive a query from a user and seek to return a list of responses ranked by their relevance. Formally, for any particular query, we have $K \in \mathbb{N}$ possible responses, *i.e.*, items that could be output. Note that K varies per query in our experiments; however, we suppress this dependence in our notation. We also have a list of *features* $X = \left\{X^{(j)}\right\}_{j=1}^{K}$ in some space $\mathcal{X}$, where $X^{(j)}$ encodes all relevant information about the jth response, including any interactions with the user's identity and query. One can think of the features as being an embedding from a neural network. Furthermore, we have a *ranking* Y that takes values in $\mathcal{Y} = S_K$, the space of permutations on K items, and determines which of the possible responses are most relevant (earlier-ranked items are more relevant). We use the notation $Y^{(j)}$ to refer to the rank of the jth response, and $Y^{(i:j)}$ to mean the subvector of Y from index i to index j, inclusive.

Finally, we have a model $\widehat{\pi}$ that takes the input X and returns a probability $\widehat{\pi}_{i,j}$, where $\widehat{\pi}_{i,j}$ is an estimate of the probability that item i is preferred to item j:

$$\widehat{\pi}_{i,j}(X) = \widehat{P}\Big(Y^{(i)} \leq Y^{(j)}\Big).$$

The model is usually trained from data to approximate the mapping from the features to the ranking.

As a motivating example of our setup, the reader can think of a search engine: K is the number of potential responses for a query, $X^{(j)}$ is the content within the jth hit, and Y is the ideal order in which the hits should be presented on the page. Since we do not know which webpages match the user's query in advance, we estimate it with a machine-learning model $\widehat{\pi}$, then select which results to display. In the next section, we propose an algorithm for returning a short list of provably high-quality responses to the user using the machine-learning model.

2.2. FDR-Controlling Sets

Based on the output of $\widehat{\pi}$, we seek to output a final set $\widehat{S} \subset \{1, \ldots, K\}$, that contains mostly good items. Formally, we will seek sets that have a low *false discovery proportion*:

$$\text{FDP}(\widehat{S}, Y) = \frac{|\widehat{S} \cap Y^{(m+1:K)}|}{\max(|\widehat{S}|, 1)}.$$

That is, for any prediction $\widehat{S}$, the FDP is the fraction of $\widehat{S}$ that does *not* fall in the top m items. Here, m is a parameter set by the analyst. For example, we might take $m = .2 \cdot K$, the top 20% of items.

A family of set-valued functions

We next explain how to produce good sets $\widehat{S}$ from the model output. Note that the $\widehat{\pi}_{i,j}$ need not be properly calibrated. Nonetheless, they do encode our model's assessment of the quality of each item. Therefore, we will create sets that include the most promising items, as judged by the model. In particular, we will rank items based on their total quality:

$$s_i(X) = \frac{1}{K-1} \sum_{j \neq i} \widehat{\pi}_{i,j}(X). \tag{1}$$

A larger s_i represents a more promising item; if the model's probabilities were correct, then s_i would be the expected fraction of other items that item i is better than. We consider sets that include only the best items, as judged by the score above:

$$\mathcal{T}_\lambda(X) = \{i : s_i(X) \geq \lambda\},$$

for $\lambda \in [0, 1]$.

Algorithm 1 The Learn then Test calibration procedure for L2R

Input: Calibration data, (X_i, Y_i), $i = 1, \ldots, n$; risk level α; error rate δ; underlying predictor $\widehat{\pi}$; step size $d\lambda > 0$.
Output: Parameter $\widehat{\lambda}$ for computing risk-controlling prediction set (RCPS).

$\lambda \leftarrow 1$
fdp $\leftarrow 1$
while fdp $\leq \alpha$ **do**
  $\lambda \leftarrow \lambda - d\lambda$
  for $i = 1, ..., n$ **do**
    $\text{FDP}_i \leftarrow \text{FDP}\big(\mathcal{T}_\lambda(X_i), Y_i\big)$
  end
  fdp $\leftarrow \frac{1}{n}\sum_{i=1}^{n} \text{FDP}_i + \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}$ ▷ Can replace with any valid UCB on the risk.
end
$\widehat{\lambda} \leftarrow \lambda + d\lambda$ ▷ Backtrack by one because we overshot.

Calibrating λ with Learn then Test

We want to find a rule $\mathcal{T}_\lambda$ that has good FDR properties. That is, we want it to control the *risk*:

$$\text{FDR}(\mathcal{T}_\lambda) = \mathbb{E}\left[\text{FDP}(\mathcal{T}_\lambda(X), Y)\right].$$

Using Learn then Test (Angelopoulos et al., 2021), we can select a parameter $\widehat{\lambda}$ that controls the risk

$$P(\text{FDR}(\mathcal{T}_{\widehat{\lambda}}) > \alpha) < \delta, \tag{2}$$

where α and δ are parameters set by the user and the FDR is taken conditionally on the calibration data — i.e., it is controlled in high probability.

We next review how to achieve the risk control in (2) with Learn then Test. Conceptually, the algorithm starts with small recommendations that are certain to control the FDR, and progressively grows them until the liminal point where making them any bigger would violate the FDR according to an upper-confidence bound (UCB). Each time we grow the set of recommendations, we must calculate a p-value telling us if the FDR is controlled, and stop if that p-value is greater than δ. Normally, calculating many p-values would incur a multiple testing penalty, but this can be avoided using a protocol known as *fixed sequence testing*; see Wiens (2003). First, we consider a discrete set of values, $\Lambda = (.99, .98, \ldots, .01)$. For each $\lambda \in \Lambda$, we consider the null hypothesis:

$$H_{0,\lambda} : \text{FDR}(\mathcal{T}_\lambda) > \alpha.$$

We test this null hypothesis using independent and identically distributed (i.i.d.) calibration data, $(X_1, Y_1), \ldots, (X_n, Y_n)$, together with a concentration result. For example, one valid test based on Hoeffding's inequality is to reject $H_{0,\lambda}$ if

$$\frac{1}{n}\sum_{i=1}^{n} \text{FDP}(\mathcal{T}_\lambda(X_i), Y_i) + \sqrt{\log(1/\delta)/(2n)} < \alpha. \tag{3}$$

That is, if the empirical risk on the calibration set is far enough below α that we can conclude that the result is not due to chance. With these tests in mind, we do the following. In decreasing order, we test each $\lambda \in \Lambda$ and stop for the first value of λ that we fail to reject, i.e., for the largest value of λ where (3) fails to hold. Then, we select $\widehat{\lambda}$ to be the preceding value: the smallest value of λ considered such that (3) holds. This procedure is valid, as stated next:

Proposition 1 (Validity of calibration (Angelopoulos et al., 2021)) *With $\widehat{\lambda}$ selected as in Algorithm 1, we have that (2) holds.*

This proposition follows from the general result of Learn then Test calibration (Angelopoulos et al., 2021).

2.3. Optimizing for Diversity While Controlling the FDR

Often, producing a high-quality recommendation involves more than simply predicting items with high ratings. For example, we may want the set to include a diversity of items—movies of different genres, search results from different sources, and so on—so the user has a more interesting set of options. Loosely, our goal might be to optimize for diversity while maintaining our rigorous FDR guarantee:

$$\begin{aligned} \max_{\lambda \in \Lambda} \quad & \mathbb{E}\Big[\text{Diversity}\big(\mathcal{D}_\lambda(X)\big)\Big] \\ \text{s.t.} \quad & \mathbb{P}\Big(\text{FDR}\big(\mathcal{D}_\lambda\big) > \alpha\Big) < \delta. \end{aligned} \tag{4}$$

The following technique will work for any diversity measure, although we describe it for one particular choice.

We work in the setting where we wish to recommend no more than $M \in \mathbb{N}$ items to a user due to, for example, constraints on the number of items that can be displayed on a webpage. Furthermore, we assume access to a set of embeddings for each possible response, $E^{(j)} \in \mathbb{R}^d$ for $j \in \{1, ..., K\}$ and $d \in \mathbb{N}$. A natural notion of diversity is based on the average distance among embeddings for responses in a prediction set $S \subseteq \{1, ..., K\}$,

$$\text{Diversity}\big(S\big) = \frac{\sum_{j,j' \in S} \big|\big|E^{(j)} - E^{(j')}\big|\big|_2}{\max\big(M, |S|\big)}. \tag{5}$$

The diversity is a deterministic function that takes as input a set of responses S and computes a positive real-valued number: the average distance between elements in S. It grows when the embeddings of the elements in S are far apart from each other. Furthermore, the diversity grows when we add more elements until we reach a set size of M, after which only the average distance matters. This particular choice of a measure for diversity is not critical; any other metric could be substituted in while maintaining the error-control guarantees below.

Next, we pick our family of sets to maximize the diversity at each choice of λ. This reduces to subsetting the original prediction set $\mathcal{T}_\lambda$ to the top M most diverse items,

$$\mathcal{D}_\lambda(X) = \underset{\substack{S \subseteq \mathcal{T}_\lambda(X) \\ |S| \leq M}}{\arg\max} \ \text{Diversity}(S).$$

Algorithm 2 The Learn then Test calibration procedure for L2R with approximate diversity optimization

Input: Calibration data, (X_i, Y_i), $i = 1, \ldots, n$; embeddings for each calibration point $E_i^{(j)}$, $j = 1, \ldots, K$; risk level α; error rate δ; underlying predictor $\widehat{\pi}$; step size $d\lambda > 0$.
Output: Parameter $\widehat{\lambda}$ for computing RCPS.

$\lambda \leftarrow 1$
fdp $\leftarrow 1$
while fdp $\leq \alpha$ **do**
  $\lambda \leftarrow \lambda - d\lambda$
  for $i = 1, ..., n$ **do**
    $\mathcal{D}_\lambda(X_i) \leftarrow \mathcal{T}_\lambda(X_i)$
    while $|\mathcal{D}_\lambda(X_i)| > M$ **do**
      leastDiverse $\leftarrow \mathcal{D}_\lambda(X_i)_1$
      leastDiversity $\leftarrow \infty$
      for $t \in \mathcal{D}_\lambda(X_i)$ **do**
        if Diversity$\big(\mathcal{D}_\lambda(X_i) \setminus t\big) \leq$ leastDiversity **then**
          leastDiverse $\leftarrow t$
          leastDiversity $\leftarrow$ Diversity$\big(\mathcal{D}_\lambda(X_i) \setminus t\big)$
        end
      end
      $\mathcal{D}_\lambda(X_i) \leftarrow \mathcal{D}_\lambda(X_i) \setminus$ leastDiverse
    end
    FDP$_i \leftarrow$ FDP$\big(\mathcal{D}_\lambda(X_i), Y_i\big)$
  end
  fdp $\leftarrow \frac{1}{n}\sum_{i=1}^{n}$ FDP$_i + \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}$ ▷ Can replace with any valid upper-confidence bound on the risk.
end
$\widehat{\lambda} \leftarrow \lambda + d\lambda$ ▷ Backtrack by one because we overshot.

In practice, we do not do the full combinatorial search over items in $\mathcal{T}_\lambda(X)$; instead, we do a greedy approximation, removing the elements that contribute least to diversity first; see Algorithm 3.

With this new family of sets, we choose $\widehat{\lambda}$ as before, replacing $\mathcal{T}$ with $\mathcal{D}$ everywhere it appears. We state the calibration procedure explicitly in Algorithm 2. Algorithm 2 first sub-selects a set of elements that are believed to be high-quality, i.e., which pass the λ threshold, and then further prunes that set by removing the least diverse elements, one at a time. It sweep λ until the point that the upper-confidence bound is violated, uncovering $\widehat{\lambda}$. Then, at test-time, Algorithm 3 does the diversity optimization for a fixed $\widehat{\lambda}$. Despite the seeming mathematical complexity of our diversity optimization, we maintain precise control over the FDR, as stated next.

Proposition 2 (Validity of calibration with approximate diversity optimization) *Let $\widehat{\lambda}$ be the result of Algorithm 2. With $\mathcal{T}_\lambda$ as the function given by Algorithm 3, we have that the FDR control in (2) holds.*

Algorithm 3 Producing calibrated predictions on a new test-point with diversity optimization in L2R

Input: Calibrated parameter $\widehat{\lambda}$; underlying predictor $\widehat{\pi}$; fresh test point (X, Y); maximum number of recommendations M.
Output: RCPS $\mathcal{D}_\lambda(X)$.

```
D_λ̂(X) ← T_λ̂(X)
while |D_λ̂(X)| > M do
    leastDiverse ← D_λ̂(X)_1
    leastDiversity ← ∞
    for t ∈ D_λ̂(X) do
        if Diversity(D_λ̂(X) \ t) ≤ leastDiversity then
            leastDiverse ← t
            leastDiversity ← Diversity(D_λ̂(X) \ t)
        end
    end
    D_λ̂(X) ← D_λ̂(X) \ leastDiverse
end
```

3. Experiments

We consider two popular L2R datasets: the Yahoo! Learning to Rank challenge (Chapelle and Chang, 2011) and the MS MARCO document re-ranking challenge (Nguyen et al., 2016). In each dataset, we take $m = \lfloor 0.2K \rfloor$; i.e., we are looking to output a prediction set that contains a high fraction of responses that are in the top 20% of the best responses. The quality of responses is determined by human raters; for example, in the Yahoo! case, there are human ratings between 1 and 5, with 5 being the best. Our metric for diversity is the distance in embedding space defined in (5). In the Yahoo! example, this is trivial to calculate, as the embeddings are given. In the MS Marco example, we use a BERT language model to calculate the embeddings; see that section for more details. In both examples, there is a tradeoff between diversity and FDR; for example, searching the word "floor" can give information related to a building or a mathematical function, the latter being less common, but more 'diverse'. Our procedure calibrates the sets to maximize diversity subject to an FDR constraint, i.e., a 'quality' constraint.

In each dataset, we use a LambdaRank neural network model (Burges, 2010) with an input size of 700, two hidden layers of size 16 and 8, ReLU activations (Nair and Hinton, 2010), and the Adam optimizer (Kingma and Ba, 2014) with its default parameters: a learning rate of 0.001, momentum parameters $\beta = (0.9, 0.999)$, and $\epsilon = 10^{-7}$. The evaluation protocol is shared between both datasets, so we describe it here. First, we randomly split the data into disjoint train and validation sets. We then train our model on the training set. Next, we repeat the following procedure 100 times:

1. Split the validation set into a calibration set and a test set.
2. Using the calibration set, compute $\widehat{\lambda}$ as in Proposition 1.

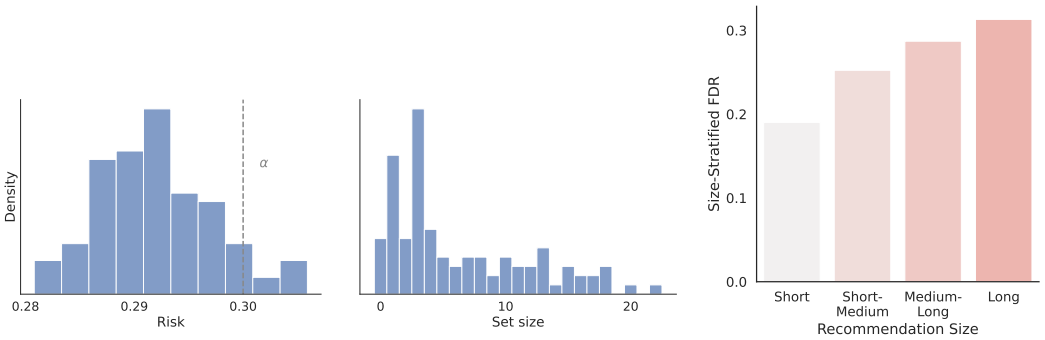

Figure 2: Results on the Yahoo! L2R dataset.

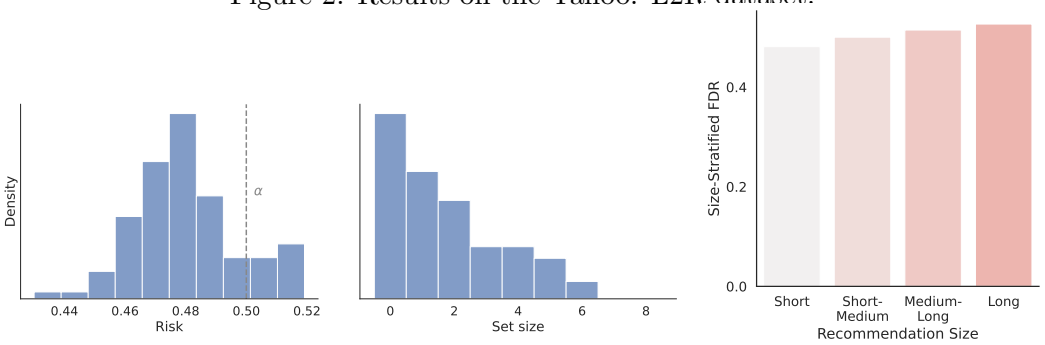

Figure 3: Results on the MS Marco Dataset.

3. Using the test set, and setting λ equal to $\widehat{\lambda}$ wherever it appears, compute the FDR risk as in 2.2, and the set size, $|\mathcal{T}_{\widehat{\lambda}}(X)|$, for a uniformly random choice of X in the validation set.

After this procedure, we have 100 values of the risk and set size, each of which was taken from a different random split of the calibration and test data. We report these values as histograms. Since we know (2) holds, we expect that the histogram of risks will not exceed the chosen value of α except with probability δ; this is indeed the case in our experiments, and the procedure is also not conservative.

In addition to providing statistical coverage, risk-controlling sets of recommendations should preferably not make systematic errors. One way to check this is by stratifying the risk along some axis and checking the risk within each stratum. If the risk is equal within each stratum, that means systematic errors are not being made along the axis of stratification. We choose to stratify along a generally applicable axis, namely set size. On the validation set, we compute the prediction sets and then compute the risk within each quartile of the set sizes. That is, letting $\left\{\left(X_i^{(\text{val})}, Y_i^{(\text{val})}\right)\right\}_{i=1}^{n'}$ be the validation examples, we form the bins

$$B_j = \left[\text{Quantile}\left(\left\{\left|\mathcal{T}_{\widehat{\lambda}}\left(X_i^{(\text{val})}\right)\right|\right\}_{i=1}^{n'}, \frac{j-1}{4}\right), \text{Quantile}\left(\left\{\left|\mathcal{T}_{\widehat{\lambda}}\left(X_i^{(\text{val})}\right)\right|\right\}_{i=1}^{n'}, \frac{j}{4}\right)\right],$$

for $j \in \{1,2,3,4\}$, where Quantile$(\{x_i\}_{i=1}^n, \beta)$ is defined as the β sample quantile of the x_i. Then, we calculate the empirical FDR within each bin, i.e.,

$$\widehat{\text{FDR}}_j = \frac{\sum_{i=1}^{n'} \text{FDP}\left(\mathcal{T}_{\hat{\lambda}}\left(X_i^{(\text{val})}\right), Y_i^{(\text{val})}\right) \mathbb{1}\left\{\left|\mathcal{T}_{\hat{\lambda}}\left(X_i^{(\text{val})}\right)\right| \in B_j\right\}}{\sum_{i=1}^{n'} \mathbb{1}\left\{\left|\mathcal{T}_{\hat{\lambda}}\left(X_i^{(\text{val})}\right)\right| \in B_j\right\}}.$$

We report each of the four stratified risks as a bar in a barplot, labeled 'Short,' 'Short-Medium,' 'Medium-Long,' and 'Long' respectively.

3.1. Yahoo! Learning to Rank

The Yahoo! Learning to Rank dataset (Chapelle and Chang, 2011) contains 36251 Yahoo! search queries, where the ith query comprises an anonymized embedding vector for each website $X_i^{(j)}$ and the ranking of the jth website, $Y_i^{(j)}$, for $j = 1, ..., K$. We subset the dataset to a smaller version with only 26090 queries and use 13045 random points for model training, $n = 8000$ for LTT calibration, and 5045 for testing.

We report the results of the FDR calibration procedure (Algorithm 1) in Figure 2, where we seek FDR control at level $\alpha = 30\%$ with a $\delta = 10\%$ tolerance level. We find that the risk is controlled and that it is nearly tight. Moreover, there is a large spread in the size of the returned set, indicating that the model is effectively discriminating between high-uncertainty and low-uncertainty inputs.

3.2. MS Marco

The MS Marco document re-ranking dataset (Nguyen et al., 2016) consists of 367013 text queries sampled from Bing. Each query has 100 documents associated with it, with each document consisting of a title and a body. This task emulates the common real-world scenario where an expressive ranking model must order documents provided to it by a lightweight nominator model. We convert each document and query into a 768-dimensional vector by passing them through a DistilBERT pre-trained model (Sanh et al., 2019) (distilbert-base-uncased in the HuggingFace library (Wolf et al., 2019)). We then concatenate the query and document vectors together for each document-query pair, which we then use as our feature vectors. We subset the dataset to include the first $20,000$ queries and use $10,000$ random points for model training, $n = 1,500$ for LTT calibration, and $8,500$ for testing.

We report the results of the FDR calibration procedure (Algorithm 1) in Figure 3, where we seek FDR control at level $\alpha = 50\%$ with a $\delta = 10\%$ tolerance level. As before, we find that the risk is controlled and that it is nearly tight. We again see a reasonable spread in the size of the returned set, as desired.

3.3. Experiments to Optimize Diversity

We also implement experiments on the Yahoo! L2R dataset to understand the properties of the diversity optimization procedure outlined in Section 2.3. As suggested by the optimization problem in (4), there is the diversity of the final set trades off with the stringency of the risk-control guarantee. For a sufficiently loose choice of α, the optimal strategy is simply to

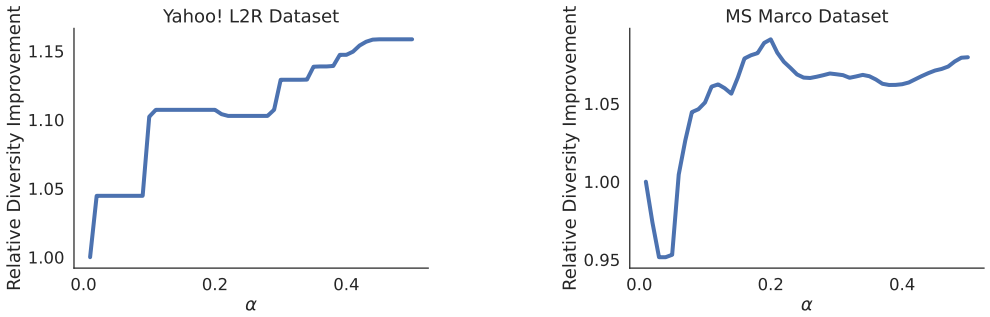

Figure 4: Diversity improvements as a function of α when optimizing for diversity subject to FDR control.

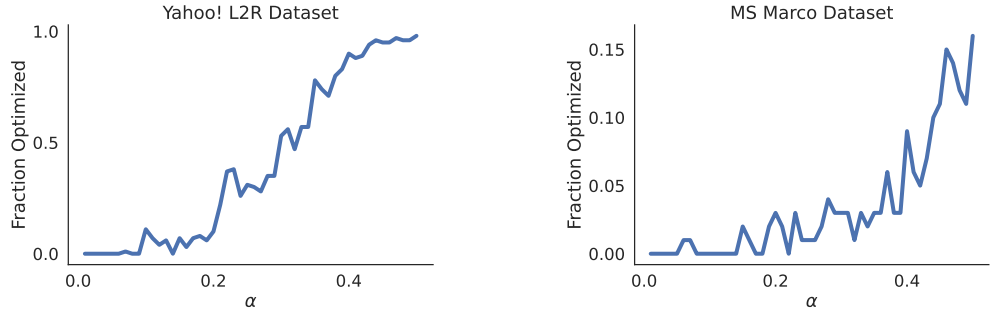

Figure 5: Fraction of elements selected for diversity optimization.

pick the M most diverse responses out of the K possible ones. As we tighten α, the ability to tolerate suboptimal responses decreases, $\mathcal{T}_\lambda$ shrinks, and the diversity optimization has fewer (but higher quality) responses to choose from.

We characterize this tradeoff by sweeping α, and for each value, estimating the relative diversity improvement

$$\mathbb{E}\left[\frac{\text{Diversity}\left(\mathcal{D}_{\widehat{\lambda}}(X)\right)}{\text{Diversity}\left(\mathcal{T}_{\widehat{\lambda}}(X)\right)}\right]. \tag{6}$$

More concretely, we repeated the following procedure 100 times.

1. Split the validation set into a calibration set and a test set.
2. Using the calibration set, compute $\widehat{\lambda}$ as in Algorithm 2.
3. Using the test set, and setting λ equal to $\widehat{\lambda}$ wherever it appears, compute the FDR risk as in 2.2, and the set size, $|\mathcal{D}_{\widehat{\lambda}}(X)|$, for a uniformly random choice of X in the validation set.

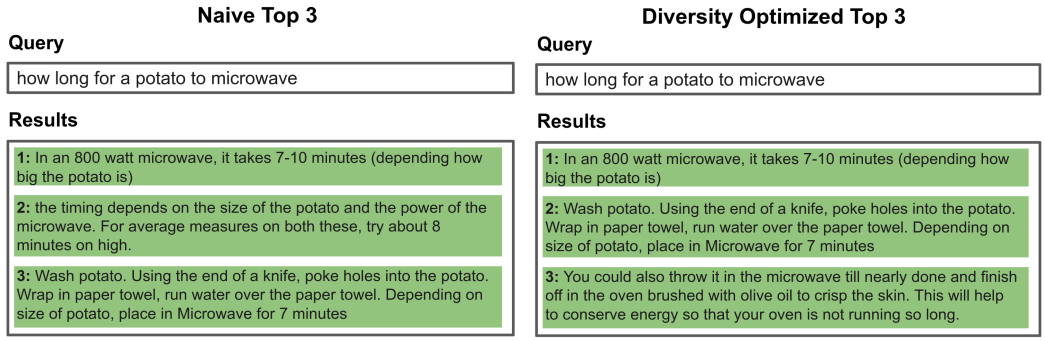

Figure 6: Comparison of the model's top three predictions against those optimized for diversity. While the naive set contains three very similar strategies to microwave the potato, the third suggestion in the diversity-optimized set includes an option to finish the potato in the oven (a unique and potentially tastier option).

4. Compute and store the values of $\text{Diversity}\big(\mathcal{T}_{\widehat{\lambda}}(\text{X})\big)$ (the diversity before optimization) and $\text{Diversity}\big(\mathcal{D}_{\widehat{\lambda}}(\text{X})\big)$ (the diversity after optimization), as well as the number of sets modified by the the diversity procedure (i.e., those such that $|\mathcal{T}_{\widehat{\lambda}}(X)| > M$).

We estimated (6) by averaging the ratio of the diversities for each set modified by the procedure. The results are shown in Figure 4 for both datasets. The non-monotone fluctuations in the curves are a consequence both of statistical noise and also of our greedy diversity optimization procedure, which does not always pick the optimal set. We also plot the fraction of sets chosen for diversity optimization, i.e., where $\mathcal{T}_{\widehat{\lambda}} > M$, as a function of α in Figure 5. More permissive choices of α allow us to optimize a larger fraction of the sets. An example of a set before and after diversity optimization is shown in Figure 6.

To query how the diversity optimization changed our previous results, we plot the risk and set size on the Yahoo! L2R dataset. The desired FDR level is $\alpha = 30\%$, with a tolerance level of $\delta = 10\%$ and $M = 3$. The plots in Figure 8 indicate the risk is still controlled and the sets have a spread, with most sets being of size three. This is not surprising, since $M = 3$, and all sets whose size was once greater than three are collapsed to size three after diversity optimization. Running the diversity optimization increased the average diversity by 15%, and 40% of the prediction sets were modified by the diversity optimization.

As a last experiment, we sweep the value of M to see how it affects the diversity of the resulting set. Keeping $\alpha = 0.3$ and $\delta = 0.1$, we vary $M \in \{2, 3, 4, 5, 6, 7, 8, 9\}$ and then estimate the relative diversity improvement in (6) with the same procedure as earlier. As before, we also report the fraction of sets changed by the optimization procedure. See the results plotted in Figure 7 for the Yahoo! L2R dataset. Increasing M excludes smaller sets (ones where the diversity optimization procedure has little room for improvement) from the diversity optimization, while also making the procedure select less aggressively for large sets.

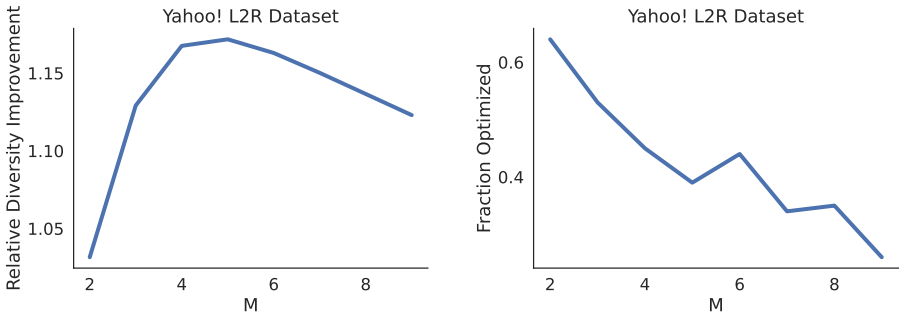

Figure 7: Diversity improvements as a function of M when optimizing diversity subject to FDR control.

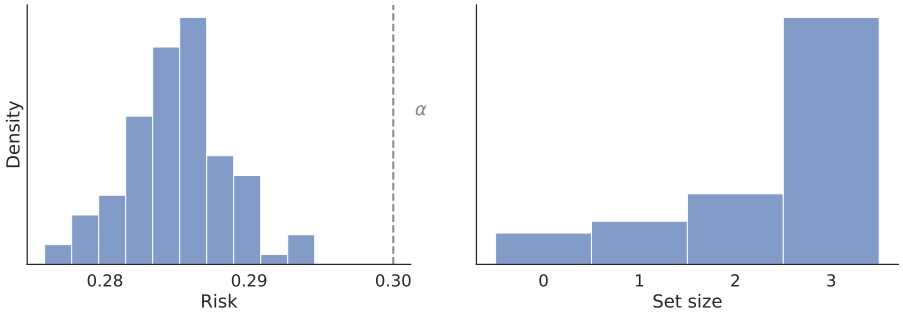

Figure 8: The risk and set size on the Yahoo! L2R dataset with diversity optimization, setting $M = 3$.

4. Discussion

The control of error rates is a critical aspect of the robust, reliable, and trustworthy deployment of learning algorithms. A key stepping stone to meeting such desiderata is to provide rigorous uncertainty quantification for a wide range of loss functions. Providing expressive uncertainty quantification also opens the door to new algorithmic capabilities. We take a step in this direction for the learning-to-rank problem, showing how to calibrate any base learning algorithm to return sets of items that control the false discovery rate. Further, we show how tracking uncertainty internally allows us to optimize for item diversity, while ensuring that the sets we return have high utility.

We wish to highlight that the framework we leverage here is entirely modular; other components can be grafted in to handle variations of the tasks we consider here. For example, the FDR notion of statistical error can be replaced with others such as the false negative rate, with minimal change to the calibration algorithm. Secondly, we could seek good performance on axes of performance other than set diversity by swapping in another performance criterion in (4). Going even farther, we could optimize for another quantify while requiring that *both* FDR and diversity are maintained at some level. Many recommender system tasks can

be handled—with finite-sample statistical guarantees—in the distribution-free risk-control framework. Finally, we note that the set construction in (1) is essentially arbitrary, and there is no reason to believe it is optimal. Just as in conformal prediction there has been a large community effort towards developing good nonconformity scores, there is likely room in the FDR control setting to consider how one might develop well-motivated and practical set constructions.

Acknowledgments

This work was supported in part by the Mathematical Data Science program of the Office of Naval Research under grant number N00014-21-1-2840 and N00014-23-1-2590, the National Science Foundation under Grant No. 2231174 and No. 2310831, and the National Science Foundation Graduate Research Fellowship.

References

Gediminas Adomavicius, Jesse Bockstedt, Shawn Curley, and Jingjing Zhang. Do recommender systems manipulate consumer preferences? A study of anchoring effects. *Information Systems Research*, 24(4):956–975, 2013.

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.

Sundus Ayyaz, Usman Qamar, and Raheel Nawaz. HCF-CRS: A hybrid content based fuzzy conformal recommender system for providing recommendations with confidence. *PloS one*, 13(10):e0204849, 2018.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6), September 2021a. ISSN 0004-5411.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279*, 2021b.

Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

Emmanuel J Candès, Lihua Lei, and Zhimei Ren. Conformalized survival analysis. *arXiv preprint arXiv:2103.09763*, 2021.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John C Duchi. Robust validation: Confident predictions even when distributions shift. *arXiv preprint arXiv:2008.04267*, 2020.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John Duchi. Predictive inference with weak supervision. *arXiv preprint arXiv:2201.08315*, 2022.

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

Victor Chernozhukov, Kaspar Wüthrich, and Zhu Yinchu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *PMLR*, pages 732–749, 06–09 Jul 2018.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

Sarah Dean, Sarah Rich, and Benjamin Recht. Recommendations and user agency: The reachability of collaboratively-filtered information. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 436–445, 2020.

Robin Dunn, Larry Wasserman, and Aaditya Ramdas. Distribution-free prediction sets with random effects. *arXiv preprint arXiv:1809.07441*, 2020.

Bradley Efron. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2010.

Clara Fannjiang, Stephen Bates, Anastasios Angelopoulos, Jennifer Listgarten, and Michael I. Jordan. Conformal prediction for the design problem. *arXiv preprint arXiv:2202.03613*, 2022.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. *arXiv preprint arXiv:2007.03114*, 2020.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. *arXiv preprint arXiv:2102.08898*, 2021.

Daniel Fleder and Kartik Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science*, 55(5):697–712, 2009.

Christoph Freudenthaler, Lars Schmidt-Thieme, and Steffen Rendle. Bayesian factorization machines. In *NIPS Workshop on Sparse Representation and Low-rank Approximation*, 2011.

Sahin Cem Geyik, Qi Guo, Bo Hu, Cagri Ozcaglar, Ketan Thakkar, Xianren Wu, and Krishnaram Kenthapadi. Talent search and recommendation systems at linkedin: Practical challenges and lessons learned. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1353–1354, 2018.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. *arXiv preprint arXiv:2106.00170*, 2021.

Prem K Gopalan, Laurent Charlin, and David Blei. Content-based recommendations with poisson factorization. *Advances in neural information processing systems*, 27, 2014.

Stephan Hammer, Andreas Seiderer, Elisabeth André, Thomas Rist, Sofia Kastrinaki, Charline Hondrou, Amaryllis Raouzaiou, Kostas Karpouzis, and Stefanos Kollias. Design of a lifestyle recommender system for the elderly: Requirement gatherings in germany and greece. In *Proceedings of the 8th ACM international conference on pervasive technologies related to assistive environments*, pages 1–8, 2015.

Jonathan Herlocker, Joseph Konstan, Loren Terveen, and John Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1): 5–53, 2004.

Tadiparthi VR Himabindu, Vineet Padmanabhan, and Arun K Pujari. Conformal matrix factorization based recommender system. *Information Sciences*, 467:685–707, 2018.

Xiaoyu Hu and Jing Lei. A distribution-free test of covariate shift using conformal prediction. *arXiv preprint arXiv:2010.07147*, 2020.

Ying Jin, Zhimei Ren, and Emmanuel J. Candès. Sensitivity analysis of individual treatment effects: A robust conformal inference approach. *arXiv preprint arXiv:2111.12161*, 2021.

Venkateswara Rao Kagita, Arun K Pujari, Vineet Padmanabhan, Sandeep Kumar Sahu, and Vikas Kumar. Conformal recommender system. *Information Sciences*, 405:157–174, 2017.

Marius Kaminskas and Derek Bridge. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems*, 7(1), 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems–A survey. *Knowledge-based systems*, 123:154–162, 2017.

Lihua Lei and Emmanuel J Candès. Conformal inference of counterfactuals and individual treatment effects. *arXiv preprint arXiv:2006.06138*, 2020.

David C Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. Related pins at pinterest: The evolution of a real-world recommender system. In *Proceedings of the 26th international conference on world wide web companion*, pages 583–592, 2017.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*, 2010.

Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. Exploring the filter bubble: The effect of using recommender systems on content diversity. In *International Conference on World Wide Web*, pages 677–686, 2014.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, 2016.

Harris Papadopoulos, Kostas Proedrou, Vladimir Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine Learning: European Conference on Machine Learning*, pages 345–356, 2002.

Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. PAC confidence sets for deep neural networks via calibrated prediction. In *International Conference on Learning Representations (ICLR)*, 2020.

Gustavo Penha and Claudia Hauff. On the calibration and uncertainty of neural learning to rank models. *arXiv preprint arXiv:2101.04356*, 2021.

Aleksandr Podkopaev and Aaditya Ramdas. Tracking the risk of a deployed model and detecting harmful distribution shifts. In *International Conference on Learning Representations*, 2022.

Stephen E Robertson and K Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. *arXiv preprint arXiv:2104.08803*, 2021.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2219–2228, 2018.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32, pages 2530–2540. 2019.

Thi Ngoc Trang Tran, Alexander Felfernig, Christoph Trattner, and Andreas Holzinger. Recommender systems in the healthcare domain: State-of-the-art and research issues. *Journal of Intelligent Information Systems*, 57(1):171–201, 2021.

Vladimir Vovk. Testing Randomness Online. *Statistical Science*, 36(4):595 – 611, 2021.

Vladimir Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, pages 444–453, 1999.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, NY, USA, 2005.

Vladimir Vovk, Ivan Petej, Paolo Toccaceli, Alexander Gammerman, Ernst Ahlberg, and Lars Carlsson. Conformal calibrators. In *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128, pages 84–99, 2020.

Chao Wang, Qi Liu, Runze Wu, Enhong Chen, Chuanren Liu, Xunpeng Huang, and Zhenya Huang. Confidence-aware matrix factorization for recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Brian L Wiens. A fixed sequence bonferroni procedure for testing multiple endpoints. *Pharmaceutical Statistics: The Journal of Applied Statistics in the Pharmaceutical Industry*, 2(3):211–215, 2003.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Jianhan Zhu, Jun Wang, Ingemar J Cox, and Michael J Taylor. Risky business: Modeling and exploiting uncertainty in information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 99–106, 2009.