Two-stage Risk Control with Application to Ranked Retrieval

Yunpeng Xu[1] , **Mufang Ying**[2] , **Wenge Guo**[3] , **Zhi Wei**[1]

[1]Department of Computer Science, New Jersey Institute of Technology
[2]Department of Statistics, Rutgers University–New Brunswick
[3]Department of Mathematical Sciences, New Jersey Institute of Technology

Abstract

Practical machine learning systems often operate in multiple sequential stages, as seen in ranking and recommendation systems, which typically include a retrieval phase followed by a ranking phase. Effectively assessing prediction uncertainty and ensuring effective risk control in such systems pose significant challenges due to their inherent complexity. To address these challenges, we developed two-stage risk control methods based on the recently proposed learn-then-test (LTT) and conformal risk control (CRC) frameworks. Unlike the methods in prior work that address multiple risks, our approach leverages the sequential nature of the problem, resulting in reduced computational burden. We provide theoretical guarantees for our proposed methods and design novel loss functions tailored for ranked retrieval tasks. The effectiveness of our approach is validated through experiments on two large-scale, widely-used datasets: MSLR-Web and Yahoo LTRC.

1 Introduction

As machine learning models become more integrated into our daily lives, the need for transparency and reliability in their predictions is crucial. Moving beyond black-box approaches, the ability to understand and quantify uncertainty in these models is increasingly important to ensure their effectiveness in real-world applications. Conformal prediction, a distribution-free and statistically valid approach that is straightforward to integrate with existing models, has emerged as a promising solution for quantifying uncertainty in machine learning [Vovk *et al.*, 2005]. Its primary objective is to generate uncertainty sets for model predictions while ensuring a specified coverage level. Recently, the conformal risk control framework [Angelopoulos *et al.*, 2024] has expanded upon traditional miscoverage control by enabling control over the expected value of any loss function. This extension greatly enhances its applicability across a wider range of contexts.

Most existing research on conformal prediction focuses on a single-stage process, where the machine learning system processes the input and generates the prediction in a single step. However, this assumption does not hold for many real-world systems, which often involve two or more concatenated stages. One notable example is ranked retrieval systems, such as search engines, where the task is to retrieve and rank documents based on their relevance to a user's query. These systems typically involve two sequential stages: (1) the retrieval stage, which identifies a set of candidate documents from a large repository, and (2) the ranking stage, which refines and orders these candidates to produce the final ranked list presented to the user [Yin and et al, 2016; Khattab *et al.*, 2020]. This two-stage approach is necessary because the massive volume of documents often exceeds the capacity of a single-stage ranking model, particularly when employing computationally intensive methods. In such two-stage problems, each stage is designed with distinct optimization objectives, and errors from one stage can propagate to the next. Consequently, the two-stage process introduces additional complexity, making it more challenging to accurately quantify and control uncertainty.

To address these challenges, we propose *two-stage* conformal prediction methods to quantify and control the uncertainty inherent in such problems. Specifically, we apply the learn-then-test framework [Angelopoulos *et al.*, 2021a] and extend the recently developed single-stage conformal risk control framework [Angelopoulos *et al.*, 2024] to a two-stage setup, where each stage has its own distinct risk control requirement. Risk control is achieved by identifying parameters that jointly satisfy the risk constraints for both stages. Furthermore, to address the specific purpose of the two stages in ranked retrieval problems, we introduce the *retrieval risk* and the *ranking risk*, respectively, and then apply our proposed two-stage risk control methods to derive their corresponding prediction sets while controlling both risks at the prespecified levels. Our proposed methods are model-agnostic and can be seamlessly integrated into existing ranked retrieval systems.

1.1 Related work

Conformal prediction Conformal prediction, originally developed by Vovk and collaborators, has recently emerged as a prominent method for uncertainty quantification in statistical machine learning [Vovk *et al.*, 1999; Papadopoulos *et al.*, 2002; Vovk *et al.*, 2005; Lei *et al.*, 2015]. A recent survey by Angelopoulos and Bates outlines the significance

and wide applications of the topic [Angelopoulos and Bates, 2021]. Our work builds upon the learn-then-test framework [Angelopoulos *et al.*, 2021a] and the recently developed conformal risk control (CRC) framework [Angelopoulos *et al.*, 2024]. The application discussed in our work shares similarities with [Angelopoulos *et al.*, 2023], which uses the LTT technique [Angelopoulos *et al.*, 2021a] to control the false discovery rate in recommender systems and optimize recommendation diversity. However, despite addressing the same ranked retrieval challenges, our work and that of [Angelopoulos *et al.*, 2023] differ in both objectives and methodologies.

Ranked retrieval Ranked retrieval has been extensively studied, with models evolving from traditional IR approaches like BM25 [Baeza-Yates and Ribeiro-Neto, 1999; Stephen and K., 1976] to modern learning-to-rank algorithms [Liu, 2009]. Recent advances in deep learning have also been successfully applied to ranked retrieval [Severyn and Moschitti, 2015; Guo *et al.*, 2016], with methods broadly categorized into pointwise [Crammer and Singer, 2001; Chu and Ghahramani, 2005], pairwise [Burges *et al.*, 2005; Freund *et al.*, 2003], and listwise approaches [Burges *et al.*, 2006; Cao *et al.*, 2007], based on their loss functions. A recent study [Wang and Joachims, 2023] also examines a similar two-stage problem in recommender systems but focuses on group fairness in the first stage, differing from our objective. Another work [Guo *et al.*, 2023] introduces stochastic ranking at inference to ensure utility or fairness in learning-to-rank models. In contrast, our work addresses a distinct focus.

2 Problem Setup

Formally, in the first stage, consider an i.i.d collection of non-increasing, right-continuous random functions $L_i^{(1)} : \Lambda \to [0,1]$, $i = 1, \ldots, n+1$, representing the associated losses. We denote by λ the tuning parameter in this stage. In the second stage, we consider another i.i.d collection of random functions, $L_i^{(2)} : \Lambda \times \Gamma \to [0,1]$, $i = 1, \ldots, n+1$, to represent the associated losses in the second stage, incorporating an additional tuning parameter γ. Here, $L_i^{(2)}(\lambda, \gamma)$ is assumed to be non-increasing and right continuous in each coordinate. Furthermore, the following conditions hold: $L_i^{(1)}(0) = 1$, $L_i^{(1)}(1) = 0$, $L_i^{(2)}(0,0) = 1$, and $L_i^{(2)}(1,1) = 0$. We use $R^{(1)}(\lambda)$ and $R^{(2)}(\lambda, \gamma)$ to denote the expected risk functions at each stage with fixed tuning parameter λ and γ, i.e., $R^{(1)}(\lambda) = \mathbb{E}L_{n+1}^{(1)}(\lambda)$ and $R^{(2)}(\lambda, \gamma) = \mathbb{E}L_{n+1}^{(2)}(\lambda, \gamma)$, and use $\widehat{R}_n^{(1)}(\lambda)$ and $\widehat{R}_n^{(2)}(\lambda, \gamma)$ to denote the empirical risk functions, i.e., $\widehat{R}_n^{(1)}(\lambda) = \frac{1}{n}\sum_{i=1}^n L_i^{(1)}(\lambda)$ and $\widehat{R}_n^{(2)}(\lambda, \gamma) = \frac{1}{n}\sum_{i=1}^n L_i^{(2)}(\lambda, \gamma)$. Without loss of generality, we assume that the parameter λ is chosen from a finite set $\Lambda = \{\lambda_i : i \in [m]\}$, and the parameter γ is chosen from a finite set $\Gamma = \{\gamma_i : i \in [m]\}$. We assume the values are ordered such that $0 \le \lambda_1 < \lambda_2 < \cdots < \lambda_m \le 1$ and $0 \le \gamma_1 < \gamma_2 < \cdots < \gamma_m \le 1$.

For pre-specified risk levels α_1, α_2, we aim to utilize random functions $\{L_i^{(1)}(\lambda)\}_{i=1}^n$ and $\{L_i^{(2)}(\lambda, \gamma)\}_{i=1}^n$ to identify data-dependent tuning parameter pairs (λ, γ) that satisfies the expected risk control guarantee

$$\mathbb{E}R^{(1)}(\lambda) \le \alpha_1 \quad \text{and} \quad \mathbb{E}R^{(2)}(\lambda, \gamma) \le \alpha_2. \tag{1}$$

When there exists a set of feasible tuning parameter pairs, $\mathcal{R}$, that satisfy (1), we are also interested in a uniform expected risk control guarantee

$$\begin{aligned} &\mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(1)}(\lambda) \le \alpha_1, \quad \text{and} \\ &\mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(2)}(\lambda, \gamma) \le \alpha_2, \end{aligned} \tag{2}$$

as this would allow us to select tuning parameter pair from $\mathcal{R}$ based on certain objective function with valid expected risk control guarantee. Note that the selection of λ and γ encodes the prediction set sizes. Throughout the paper, we assume α_1, α_2 are fixed to be in the interval $[0, 1]$, which implies feasible risk control.

With regard to two-stage risk control, one might wonder: if we can manage risk effectively in the second stage, why invest effort in controlling it in the first stage? The reason is that controlling risk in the first stage is foundational to the entire process. In the case of the ranked retrieval problem, retrieving all documents in the first stage imposes a significant computational burden on the ranking stage. Conversely, retrieving too few relevant documents in the first stage undermines the feasibility of second-stage risk control and compromises ranking quality. Thus, this paper focuses on simultaneously controlling risks at both stages.

2.1 Data Structure in Ranked Retrieval Problem

Before discussing how to achieve risk control as specified in equations (1) and (2), we first outline the structure of the data for the ranked retrieval problem. Consider a set of i.i.d calibration data points $\{(X_i, Y_i, Z_i)\}_{i=1}^n$, with $(X_i, Y_i, Z_i) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, where Y_i and Z_i are the labeled responses of feature vector X_i corresponding to the two stages respectively. In the ranked retrieval problem, X_i represents a query along with its associated candidate documents. We let $X_i = \{q_i, \{d_{i,j}\}_{j=1}^{N_i}\}$, where q_i denotes the user query and $\{d_{i,j}\}_{j=1}^{N_i}$ denotes its associated documents with size N_i. Each pair of q_i and $d_{i,j}$ is associated with a ground truth relevance score $r_{i,j} \in \{0, 1, \ldots, R\}$, where a higher value of $r_{i.j}$ indicates a higher relevance of $d_{i,j}$ to q_i. This score is only observable for the training data and is hidden for the test data. Here we denote the relevant documents with a relevance score great than 0 in the retrieval results of q_i by $Y_i = \{d_{i,j} : r_{i,j} > 0\}$. In the ranking stage, with a focus on the ranking quality of documents with a ground truth relevance level $r_0 \in [R]$ or above, the set of r_0-relevant documents for q_i: $\{d_{i,j} : r_{i,j} \ge r_0\}$ is considered. Then, Z_i denotes the ordered set of the r_0-relevant documents, sorted in descending order based on the ground truth relevance scores with ties broken arbitrarily: $Z_i = \{d_{i,(1)}, d_{i,(2)}, \ldots\}$.

Without loss of generality, we assume that each stage is associated with a model learned on the training data for all queries, denoted by $\mathcal{M}_{\text{retrieval}}$ for the retrieval model and by $\mathcal{M}_{\text{rank}}$ for the ranking model, respectively. Note that the form of the retrieval model is flexible; it can range from a simple Okapi BM25 model, which counts word occurrences, to a more complex large language model that generates embeddings for embedding-based retrieval. Typically, the model

used in the retrieval stage is more efficient but less powerful than the one in the ranking stage. For both stages, by leveraging the pre-trained model and calibration data, we can construct prediction sets $\hat{\mathcal{C}}^{(1)}(x;\lambda)$ for the unknown response $y \in \mathcal{Y}$ and $\hat{\mathcal{C}}^{(2)}(x;\lambda,\gamma)$ for the unknown response $z \in \mathcal{Z}$, given a test data point $(x,y,z) \in \mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$. By employing the loss functions $l^{(1)}$ and $l^{(2)}$ for the first and second stages, respectively, we obtain:

$$L_i^{(1)}(\lambda) = l^{(1)}(\hat{\mathcal{C}}^{(1)}(X_i;\lambda), Y_i), \quad \text{and}$$
$$L_i^{(2)}(\lambda,\gamma) = l^{(2)}(\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma), Z_i).$$

We will specify the choice of $\hat{\mathcal{C}}^{(1)}(\cdot;\lambda)$, $\hat{\mathcal{C}}^{(2)}(\cdot;\lambda,\gamma)$, $l^{(1)}$, and $l^{(2)}$ for ranked retrieval problem in Section 4.

3 Two-stage Risk Control

In this section, we present two approaches for achieving expected risk control. The first is a direct application of the LTT framework [Angelopoulos *et al.*, 2021b], included for completeness, as high-probability risk control offers a stronger guarantee than expected risk control. In the second approach, we extend the conformal risk control framework [Angelopoulos *et al.*, 2024] to accommodate the two-stage setting.

3.1 LTT Framework

In the first stage, the value of λ is determined by evaluating its risk function $R^{(1)}(\lambda)$ through the following hypothesis tests for each $\lambda_i \in \Lambda$:

$$\mathcal{H}_i^{(1)} : R^{(1)}(\lambda_i) > \alpha_1 \quad \text{vs.} \quad \mathcal{H}'^{(1)}_i : R^{(1)}(\lambda_i) \le \alpha_1.$$

Given $\lambda = \lambda_i \in \Lambda$, the value of γ is selected in the second stage by using the corresponding risk function $R^{(2)}(\lambda_i,\gamma)$. For each $\gamma_j \in \Gamma$, the following hypothesis tests are performed:

$$\mathcal{H}_{i,j}^{(2)} : R^{(2)}(\lambda_i,\gamma_j) > \alpha_2 \quad \text{vs.} \quad \mathcal{H}'^{(2)}_{i,j} : R^{(2)}(\lambda_i,\gamma_j) \le \alpha_2.$$

Let $\mathcal{F}^{(1)} = \left\{\mathcal{H}_i^{(1)} : i = 1,\dots,m\right\}$ denote the collection of all hypotheses tested in the first stage. Given $\lambda = \lambda_i \in \Lambda$, let $\mathcal{F}_i^{(2)} = \left\{\mathcal{H}_{i,j}^{(2)} : j = 1,\dots,m\right\}$ denote the collection of all hypotheses tested in the second stage for a fixed λ_i. Finally, let $\mathcal{F}^{(2)} = \bigcup_{i=1}^m \mathcal{F}_i^{(2)}$ represent the complete collection of all hypotheses tested in the second stage. To conduct hypothesis testing across both stages, we aim to control the global family-wise error rate (FWER), defined as the probability of making at least one Type I error across the two families $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$. For each individual hypothesis $\mathcal{H}_i^{(1)} \in \mathcal{F}^{(1)}$ in the first stage and $\mathcal{H}_{i,j}^{(2)} \in \mathcal{F}^{(2)}$ in the second stage, we use the Hoeffding-Bentkus inequality to compute p-values $p_i^{(1)}$ and $p_{i,j}^{(2)}$, as introduced in [Angelopoulos *et al.*, 2021a]. These p-values are valid under their respective null hypotheses and are defined as follows:

$$\begin{aligned} p_i^{(1)} =& \min(\exp\{-nh(\widehat{R}_n^{(1)}(\lambda_i)\wedge\alpha_1,\alpha_1)\},\\ & e\mathbb{P}(\mathrm{Bin}(n,\alpha_1) \le \lceil n\widehat{R}_n^{(1)}(\lambda_i)\rceil)),\\ p_{i,j}^{(2)} =& \min(\exp\{-nh(\widehat{R}_n^{(2)}(\lambda_i,\gamma_j)\wedge\alpha_2,\alpha_2)\},\\ & e\mathbb{P}(\mathrm{Bin}(n,\alpha_2) \le \lceil n\widehat{R}_n^{(2)}(\lambda_i,\gamma_j)\rceil)). \end{aligned}$$

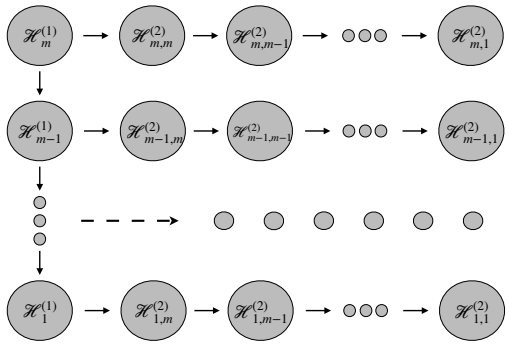

Figure 1: Graphical ordering of hypothesis tests

Here the function $h(a,b)$ is given by:

$$h(a,b) = a\log\left(\frac{a}{b}\right) + (1-a)\log\left(\frac{1-a}{1-b}\right).$$

To sequentially test the families $\mathcal{F}^{(1)}$ and $\mathcal{F}_i^{(2)}$ for each $i \in [1,m]$, one can employ the FWER controlling algorithm to obtain the tuning parameter pairs [Angelopoulos *et al.*, 2021a]. Given that the loss function in the first stage, $L_i^{(1)}(\lambda)$, is non-increasing in λ, and the loss function in the second stage, $L_i^{(2)}(\lambda,\gamma)$, is non-increasing in both λ and γ, the fixed-sequence procedure tests the hypotheses in these families in reverse order. Specifically, for $\mathcal{F}^{(1)}$, the m hypotheses are tested sequentially in the order $\mathcal{H}_m^{(1)}, \mathcal{H}_{m-1}^{(1)}, \dots, \mathcal{H}_1^{(1)}$. Similarly, for $\mathcal{F}_i^{(2)}$, the hypotheses are tested sequentially in the order $\mathcal{H}_{i,m}^{(2)}, \mathcal{H}_{i,m-1}^{(2)}, \dots, \mathcal{H}_{i,1}^{(2)}$. Building on these considerations, we propose the following procedure for simultaneously testing $\mathcal{F}^{(1)} \cup \mathcal{F}^{(2)}$, ensuring control of the global FWER at the pre-specified level δ:

Procedure:

1. **Test $\mathcal{F}^{(1)}$ using the Bonferroni procedure:**
   Apply the Bonferroni procedure to the p-values $p_i^{(1)}$ to simultaneously test $\mathcal{F}^{(1)}$ at level δ. Let $\mathcal{R}^{(1)}$ denote the index set of the rejected hypotheses:
   $$\mathcal{R}^{(1)} = \left\{i \in [m] : p_i^{(1)} \le \delta/m\right\}.$$
2. **Test $\mathcal{F}_i^{(2)}$ using the fixed-sequence procedure:**
   For each $i \in \mathcal{R}^{(1)}$, use the fixed-sequence procedure on the p-values $p_{i,j}^{(2)}$ to simultaneously test $\mathcal{F}_i^{(2)}$ at level δ/m. Let $\mathcal{R}_i^{(2)}$ denote the index set of the rejected hypotheses:
   $$\mathcal{R}_i^{(2)} = \left\{j \in [m] : p_{i,j'}^{(2)} \le \delta/m \quad \forall j' \in [j,m]\right\}.$$
3. **Determine the final set of rejected hypothesis pairs:**
   The set $\mathcal{R}$ of tuning parameter pairs that correspond to the rejected pairs of hypotheses $\left(\mathcal{H}_i^{(1)}, \mathcal{H}_{i,j}^{(2)}\right)$ is:
   $$\mathcal{R} = \left\{(\lambda_i,\gamma_j) : i \in \mathcal{R}^{(1)}, j \in \mathcal{R}_i^{(2)}\right\}.$$

The procedure described above can be viewed as a special case of the sequential graphical approach for multiple testing [Bretz *et al.*, 2009; Angelopoulos *et al.*, 2021a]. Consequently, it strongly controls the global FWER at the pre-specified level δ. We note that some other alternative global

FWER-controlling procedures can be applied; a few examples are provided in the appendix. However, determining optimal procedure for a specific setting remains an open problem. In the following, we present the expected risk control guarantee for general FWER-controlling procedure.

Theorem 1. *Let $\mathcal{R}$ denote the collection of tuning parameter pairs returned from a FWER controlling algorithm testing $\mathcal{F}^{(1)} \cup \mathcal{F}^{(2)}$ at level δ. Then, we have*

$$\mathbb{P}\left(\forall(\lambda,\gamma) \in \mathcal{R} : R^{(1)}(\lambda) \leq \alpha_1, R^{(2)}(\lambda,\gamma) \leq \alpha_2\right) \geq 1-\delta.$$

Therefore, we have

$$\mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(1)}(\lambda) \leq \alpha_1 + \delta, \quad \textit{and}$$
$$\mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(2)}(\lambda,\gamma) \leq \alpha_2 + \delta.$$

A few comments are in order regarding the comparison with multiple risks in [Angelopoulos *et al.*, 2021a]. The authors therein discuss a similar setting involving multiple risk functions. Specifically, consider the case where the sets of tuning parameters have size m, i.e., $|\Lambda| = |\Gamma| = m$. Their procedure uses the set of all parameter pairs $\{(\lambda,\gamma) : \lambda \in \Lambda, \gamma \in \Gamma\}$, as input for running FWER controlling algorithm. In contrast, our approach leverages the sequential nature of the problem and the monotonicity of the risk functions, which reduces the computational burden and enhances overall effectiveness.

3.2 Expected Risk Control

In this section, we describe methods that leverage conformal risk control framework. To proceed, we consider $\alpha_1, \alpha_2 \in (\frac{1}{n+1}, 1]$ and introduce a few additional notations. For the purpose of risk control in the first stage, define:

$$\hat{\lambda}_0^{(1)} = \inf\left\{\lambda \in \Lambda : \sum_{i=1}^{n} L_i^{(1)}(\lambda) \leq (n+1)\alpha_1 - 1\right\}.$$

Similarly, to ensure the feasibility of risk control in the second stage, define:

$$\hat{\lambda}_0^{(2)} = \inf\left\{\lambda \in \Lambda : \sum_{i=1}^{n} L_i^{(2)}(\lambda,1) \leq (n+1)\alpha_2 - 1\right\}.$$

Lastly, for any fixed $\lambda \in \Lambda$, we define:

$$\hat{\gamma}_0^{(2)}(\lambda) = \inf\left\{\gamma \in \Gamma : \sum_{i=1}^{n} L_i^{(2)}(\lambda,\gamma) \leq (n+1)\alpha_2 - 1\right\}. \tag{3}$$

When the set in equation (3) is empty, we set $\hat{\gamma}_0^{(2)}(\lambda) = 1$. To ensure risk control in both stages, we set the value of λ as a linear combination of $\hat{\lambda}_0^{(1)} \vee \hat{\lambda}_0^{(2)}$ and 1:

$$\hat{\lambda}^{(1)}(t) := \left\lceil t(\hat{\lambda}_0^{(1)} \vee \hat{\lambda}_0^{(2)}) + (1-t)\right\rceil_{\Lambda}.$$

Here $\hat{\lambda}^{(1)}(t)$ is defined to be the ceiling of $t(\hat{\lambda}_0^{(1)} \vee \hat{\lambda}_0^{(2)}) + (1-t)$, restricted to values in Λ, and $t \in [0,1]$ is a tuning parameter. This formulation ensures that $\hat{\lambda}^{(1)}(t) \geq \hat{\lambda}_0^{(1)} \vee \hat{\lambda}_0^{(2)}$ for any t. Using $\hat{\lambda}^{(1)}(t)$, we determine the value of γ by defining

$$\hat{\gamma}^{(2)}(t) := \hat{\gamma}_0^{(2)}\left(\hat{\lambda}^{(1)}(t)\right).$$

With the above definitions, we are ready to state our results for expected risk control.

Theorem 2. *For any $t \in [0,1]$, tuning parameter pair $(\hat{\lambda}^{(1)}(t), \hat{\gamma}^{(2)}(t))$ achieves first-stage risk control at level α_1 with a finite-sample guarantee and achieves second-stage risk control at level α_2 asymptotically, i.e.,*

$$\mathbb{E}R^{(1)}(\hat{\lambda}^{(1)}(t)) \leq \alpha_1, \quad \textit{and}$$
$$\limsup_{n\to\infty} \mathbb{E}R^{(2)}(\hat{\lambda}^{(1)}(t), \hat{\gamma}^{(2)}(t)) \leq \alpha_2.$$

Since Λ and Γ are finite sets, we further have the following corollary.

Corollary 1. *Uniform risk control for both stages can be achieved in the set*

$$\mathcal{R} = \left\{(\lambda,\gamma) \in \Lambda \times \Gamma : \lambda = \hat{\lambda}^{(1)}(t), \gamma \geq \hat{\gamma}_0^{(2)}(\lambda), t \in [0,1]\right\},$$

i.e.,

$$\mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(1)}(\lambda) \leq \alpha_1, \quad \textit{and}$$
$$\limsup_{n\to\infty} \mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(2)}(\lambda,\gamma) \leq \alpha_2.$$

Finite-sample second-stage risk control

To ensure finite-sample guarantee for the second stage, a data-splitting approach can be employed. Let calibration data be divided into two non-overlapping parts with index sets $\mathcal{I}_1$ and $\mathcal{I}_2$. Let $\alpha_1, \alpha_2 \in (1/(1+|\mathcal{I}_1|), 1]$. Define

$$\tilde{\lambda}_0^{(1)} = \inf\left\{\lambda \in \Lambda : \sum_{i\in\mathcal{I}_1} L_i^{(1)}(\lambda) \leq (|\mathcal{I}_1|+1)\alpha_1 - 1\right\}.$$

To proceed, we need to impose the following additional assumption to ensure finite-sample risk control in the second stage:

Assumption 1. *There exists a known constant $\lambda_0 \leq 1$ such that $L_i^{(2)}(\lambda_0, 1) \leq \alpha_2$ for $i \in [n+1]$.*

The above Assumption 1 enables feasibility of finite-sample second-stage risk control. Next, for $\lambda \in \Lambda \cap [\lambda_0, 1]$ define

$$\tilde{\gamma}_0^{(2)}(\lambda) = \inf\left\{\gamma \in \Gamma : \sum_{i\in\mathcal{I}_2} L_i^{(2)}(\lambda,\gamma) \leq (|\mathcal{I}_2|+1)\alpha_2 - 1\right\}.$$

We define $\tilde{\gamma}_0^{(2)}(\lambda) = 1$ when the set is empty. For $t \in [0,1]$, we then define:

$$\begin{aligned} &\tilde{\lambda}^{(1)}(t) := \left\lceil t(\tilde{\lambda}_0^{(1)} \vee \lambda_0) + (1-t)\right\rceil_{\Lambda}, \quad \text{and} \\ &\tilde{\gamma}^{(2)}(t) = \tilde{\gamma}_0^{(2)}(\tilde{\lambda}^{(1)}(t)). \end{aligned} \tag{4}$$

Theorem 3. *For any $t \in [0,1]$, tuning parameter pair $(\tilde{\lambda}^{(1)}(t), \tilde{\gamma}^{(2)}(t))$ achieves finite-sample first-stage and second-stage risk control at level α_1, α_2, respectively, i.e.,*

$$\mathbb{E}R^{(1)}(\tilde{\lambda}^{(1)}(t)) \leq \alpha_1 \quad \textit{and} \quad \mathbb{E}R^{(2)}(\tilde{\lambda}^{(1)}(t), \tilde{\gamma}^{(2)}(t)) \leq \alpha_2.$$

Note that for any $t \in [0,1]$, we have $\mathbb{E}R^{(2)}(\tilde{\lambda}^{(1)}(t), \tilde{\gamma}^{(2)}(t)) \leq \alpha_2$. However, there is no guarantee that

$$\mathbb{E} \sup_{t\in[0,1]} R^{(2)}(\tilde{\lambda}^{(1)}(t), \tilde{\gamma}^{(2)}(t)) \leq \alpha_2,$$

as no monotonic relationship exists. Therefore, to obtain a finite-sample uniform risk control in the second stage, we define

$$\bar{\gamma}^{(2)} = \inf\left\{\gamma \in \Gamma : \sum_{i\in\mathcal{I}_2} L_i^{(2)}(\tilde{\lambda}^{(1)}(1), \gamma) \leq (|\mathcal{I}_2|+1)\alpha_2 - 1\right\}$$

Corollary 2. *Uniform risk control for both stages can be achieved in the set*

$$\mathcal{R} = \left\{(\lambda,\gamma) \in \Lambda \times \Gamma : \lambda = \tilde{\lambda}^{(1)}(t), \gamma \geq \bar{\gamma}^{(2)}, t \in [0,1]\right\},$$

i.e.,

$$\mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(1)}(\lambda) \leq \alpha_1 \quad \textit{and} \quad \mathbb{E} \sup_{(\lambda,\gamma)\in\mathcal{R}} R^{(2)}(\lambda,\gamma) \leq \alpha_2.$$

We remark that our data-splitting approach, which ensures a uniform finite-sample risk control guarantee, can be regarded as a special case of the method proposed in Section 4.3 of [Angelopoulos *et al.*, 2024].

4 Application to Ranked Retrieval

In this section, we apply our proposed methods to ranked retrieval problem. We begin by introducing the loss functions defined for each stage.

4.1 Loss Function for Retrieval Stage

As defined in the Section 2, Y_i is the set of relevant documents with respect to the query q_i, i.e., the set of documents with ground truth relevance scores greater than 0. To represent the set of documents fetched by $\mathcal{M}_{\text{retrieval}}$ used in the retrieval stage, we define the retrieved document set $\hat{\mathcal{C}}^{(1)}(X_i;\lambda)$ for the retrieval stage as:

$$\hat{\mathcal{C}}^{(1)}(X_i;\lambda) = \{d_{i,j} : \mathcal{M}_{\text{retrieval}}(q_i, d_{i,j}) \geq 1-\lambda\},$$

where $\lambda \in [0,1]$ and $\mathcal{M}_{\text{retrieval}}(q_i, d_{i,j})$ denotes the model score for the query-document pair $(q_i, d_{i,j})$, as computed by the retrieval model $\mathcal{M}_{\text{retrieval}}$. Note that a good retrieved document set $\hat{\mathcal{C}}^{(1)}(X_i;\lambda)$ associated with the query q_i should aim to cover as many relevant documents in Y_i as possible. To measure the miscoverage of Y_i by $\hat{\mathcal{C}}^{(1)}(X_i;\lambda)$, we define retrieval loss as:

$$L_i^{(1)}(\lambda) = 1 - \frac{|Y_i \cap \hat{\mathcal{C}}^{(1)}(X_i;\lambda)|}{|Y_i|}. \tag{5}$$

Note that the loss function defined in equation (5) is non-increasing, right-continuous in λ, and bounded within $[0,1]$. This form of loss function quantifies the missed fraction of relevant documents retrieved by model $\mathcal{M}_{\text{retrieval}}$. By choosing an appropriate λ, we aim to control the risk at level α_1.

4.2 Loss Function for Ranking Stage

Similarly, in the ranking stage, we first define the prediction set $\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma)$ as:

$$\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma) = \{d_{i,j} : \mathcal{M}_{\text{rank}}(q_i, d_{i,j}) \geq 1-\gamma\} \cap \hat{\mathcal{C}}^{(1)}(X_i;\lambda). \tag{6}$$

In contexts like information retrieval, search engines, and recommendation systems, nDCG [Järvelin and Kekäläinen, 2000] is widely used to evaluate the ranking quality of algorithms or systems. Motivated by the intuition from nDCG that a ranked list can be evaluated by rewarding relevance while considering ranking positions, we slightly modify the definition of nDCG to suit our setting, and define:

$$\text{DCG}_{\text{mod}}\left(\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma), Z_i\right) = \sum_{j=1}^{|Z_i|} \frac{\mathbf{1}_{\{d_{i,(j)} \in \hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma)\}}}{\log(j+1)}.$$

Notably, when $\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma)$ contains all the r_0-relevant documents, DCG_{mod} attains its maximum value. To normalize the DCG_{mod}, we define the modified Ideal Discounted Cumulative Gain (iDCG_{mod}) for query q_i as:

$$\text{iDCG}_{\text{mod}}(Z_i) = \sum_{j=1}^{|Z_i|} \frac{1}{\log(j+1)}.$$

Correspondingly, we define the modified Normalized Discounted Cumulative Gain (nDCG_{mod}) as

$$\text{nDCG}_{\text{mod}}\left(\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma), Z_i\right) = \frac{\text{DCG}_{\text{mod}}\left(\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma), Z_i\right)}{\text{iDCG}_{\text{mod}}(Z_i)}.$$

The loss function for the ranking stage is then defined as

$$L_i^{(2)}(\lambda,\gamma) = 1 - \text{nDCG}_{\text{mod}}(\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma), Z_i). \tag{7}$$

The proposed loss function in equation (7) addresses the ranking order within the ground truth set Z_i. When selecting an appropriate γ that satisfies risk control guarantee, greater weight is assigned to documents in Z_i with higher ranking positions, ensuring that the most relevant documents are prioritized for inclusion in the prediction set. Note that, given λ specified in the first stage, the ranking loss function is non-increasing, right-continuous in γ, and bounded within $[0,1]$. Our goal is to determine γ to control the ranking loss at specified level α_2.

4.3 Parameter Pair Selection via Empirical Set Size Minimization

Given a collection of tuning parameter pairs that achieve risk control at both stages, we determine the tuning parameter pair $(\hat{\lambda}, \hat{\gamma})$ through optimization with a objective function $\mathcal{L}$:

$$(\hat{\lambda}, \hat{\gamma}) = \text{argmin}_{(\lambda,\gamma)\in\mathcal{R}} \mathcal{L}\left(\lambda, \gamma; \{X_i\}_{i=1}^n\right). \tag{8}$$

In many applications (recommendation in mobile devices etc.), it is desirable to produce a smaller prediction set in the second stage. Therefore, in this paper, we consider objective function

$$\mathcal{L}\left(\lambda, \gamma; \{X_i\}_{i=1}^n\right) = \frac{1}{n}\sum_{i=1}^n |\hat{\mathcal{C}}^{(2)}(X_i;\lambda,\gamma)\}|.$$

We comment that, the choice of the loss function should be driven by the specific problem at hand. For instance, in scenarios where the objective is to reduce computational cost during the ranking stage, a smaller prediction set size in the retrieval stage reduces the number of documents evaluated by the computationally intensive ranking model. To address this, the term $\frac{1}{n}\sum_{i=1}^{n}|\hat{\mathcal{C}}^{(1)}(X_i;\lambda)|$ can be incorporated into the loss function $\mathcal{L}$, thereby systematically mitigating the computational burden.

4.4 Experiments

We address the ranked retrieval problem using three approaches: the learn-then-test framework (LTT), two-stage conformal risk control (tCRC), and two-stage conformal risk control with data splitting (tCRC-s[1]). The tuning parameter pairs $(\hat{\lambda}, \hat{\gamma})$ for these methods are determined by equation (8). Accordingly, these selected tuning parameter pairs ensure expected risk control guarantees, as established in Theorem 1, Corollary 1, and Corollary 2, respectively. Our experiments are conducted on two datasets: MSLR-Web, and Yahoo LTRC. For each dataset and experiment, we split the data into a calibration set and a test set, and then apply different methods to the data. The calibration set is used to determine the tuning parameter pair, while the test set is used for evaluation. Using the test data and the tuning parameter pair computed from the calibration data, we compute the following quantities for evaluation: empirical risks in two stages (Risk (1) represents empirical risk in the first stage, Risk (2) represents empirical risk in the second stage), average prediction set size in the second stage, average recall for documents with a relevance level greater than 2, average recall for documents with a relevance level equal to 1, and average precision for documents with relevance level greater than 1. Then, we replicate each experiment 10 times and report their averages.

In Table 1, we summarize results for the dataset MSLR. From the table, we observe that both Recall (≥ 2) and Recall (1) exhibit relatively high values, indicating that the prediction sets effectively cover a substantial proportion of relevant documents. Additionally, Recall (≥ 2) consistently achieves higher values than Recall (1). This confirms that our proposed ranking loss effectively prioritizes documents with higher relevance levels, resulting in prediction sets that are more likely to include highly relevant documents. In the table, we vary the risk levels and consider $(\alpha_1, \alpha_2) \in \{(0.1, 0.1), (0.01, 0.1), (0.1, 0.2)\}$. Notably, LTT exhibits the lowest Risk (1) and Risk (2) values, indicating that LTT is the most conservative approach. This can also be explained by the fact that the size of the feasible set $\mathcal{R}$ for LTT is, on average, the smallest. Conversely, we observe that tCRC achieves the smallest prediction set size after tuning parameter selection via equation (8), attributed to its larger feasible set $\mathcal{R}$. Notably, compared with tCRC-s, tCRC maintains a larger set size in the second stage but achieves a smaller Recall (≥ 2). While this seems counterintuitive, it can be explained by that fact that tCRC-s has a larger risk in the first stage. This is

[1]While the required information λ_0 for tCRC-s is unknown in practice, we use the calibration data with index set $\mathcal{I}_1$ to obtain an estimate.

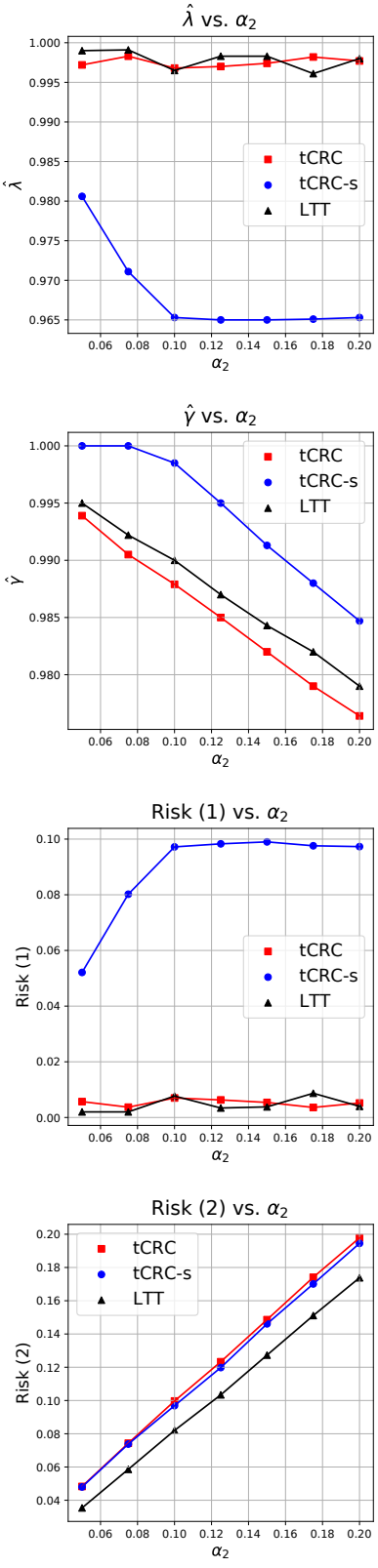

Figure 2: $(\hat{\lambda}, \hat{\gamma})$ and Risk under fixed α_1 and varying α_2

(α_1, α_2)	Method	Risk (1)	Risk (2)	Set size	Recall (≥ 2)	Recall (1)	Precision
$(0.1, 0.1)$	tCRC	0.0069	0.0982	67.86	0.9537	0.8550	0.7504
	tCRC-s	0.0986	0.0969	77.41	0.9279	0.8790	0.6673
	LTT	0.0003	0.0772	70.25	0.9642	0.8857	0.7463
$(0.01, 0.1)$	tCRC	0.0058	0.0994	68.08	0.9532	0.8533	0.7511
	tCRC-s	0.0078	0.0967	68.44	0.9537	0.8577	0.7503
	LTT	0.0002	0.0765	70.66	0.9646	0.8864	0.7467
$(0.1, 0.2)$	tCRC	0.0051	0.2006	57.50	0.9084	0.7017	0.7578
	tCRC-s	0.0972	0.1971	58.28	0.8825	0.7300	0.7575
	LTT	0.0046	0.1718	60.22	0.9221	0.7440	0.7578

Table 1: Summary table for dataset MSLR

(α_1, α_2)	Method	Risk (1)	Risk (2)	Set size	Recall (≥ 2)	Recall (1)	Precision
$(0.1, 0.1)$	tCRC	0.0053	0.1001	27.50	0.9458	0.8157	0.9414
	tCRC-s	0.0985	0.0974	28.80	0.9246	0.8577	0.9069
	LTT	0.0052	0.0787	28.40	0.9572	0.8556	0.9383
$(0.01, 0.1)$	tCRC	0.0025	0.0984	27.46	0.9466	0.8187	0.9414
	tCRC-s	0.0078	0.0966	27.56	0.9464	0.8245	0.9405
	LTT	0.0019	0.0748	28.43	0.9599	0.8610	0.9381
$(0.1, 0.2)$	tCRC	0.0144	0.2014	23.36	0.8885	0.6325	0.9480
	tCRC-s	0.0966	0.1991	23.60	0.8698	0.6682	0.9477
	LTT	0.0047	0.1722	24.49	0.9070	0.6823	0.9472

Table 2: Summary table for dataset Yahoo

achieved by using a smaller $\hat{\lambda}$, which results in fewer documents with higher relevance levels being retrieved in the first stage, ultimately impacting the performance in the second stage. When α_1 is reduced from 0.1 to 0.01, the metrics of tCRC and LTT exhibit limited variation, whereas tCRC-s demonstrates greater sensitivity to this change. This increased sensitivity can be attributed to the construction of the feasible set by tCRC-s. Additionally, when α_2 is increased from 0.1 to 0.2, we observe that Risk (2) slightly exceeds the target risk level of 0.2. This deviation can be explained by approximation error and the asymptotic validity of tCRC, as established in Corollary 1.

Figure 2 illustrates $(\hat{\lambda}, \hat{\gamma})$ and the corresponding risks over two stages, with $\alpha_1 = 0.1$ and α_2 varying in the set $0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2$. Further visualizations are included in the appendix. All three methods effectively manage risk within the desired levels. Notably, as α_2 varies, tCRC achieves the smallest prediction set size in the second stage, LTT attains the highest recall rate, and both tCRC and LTT demonstrate superior precision. In practice, we recommend using LTT or tCRC. When the sample size is small, tCRC-s may suffer from lower sample efficiency, compounded by the challenge of the unknown parameter λ_0. Lastly, we present similar results for the Yahoo dataset in Table 2, with the corresponding figures deferred to the appendix.

5 Conclusion & Discussion

In this paper, we study expected risk control under a two-stage setup. We propose methods aimed at simultaneously controlling risk, and establish theoretical guarantees for these methods. The effectiveness of our approach is validated through experiments on two large-scale, widely used datasets: MSLR-Web and Yahoo LTRC. Below, we discuss several extensions of the current framework.

Non-monotone risk function When risk function is non-monotone, the following monotonization procedure can be applied. At the first stage, the loss function is modified as follows:

$$\tilde{L}_i^{(1)}(\lambda) = \sup_{\lambda' \in [\lambda, 1] \cap \Lambda} L_i^{(1)}(\lambda').$$

At the second stage, we define the modified loss function as

$$\tilde{L}_i^{(2)}(\lambda, \gamma) = \sup_{\substack{\lambda' \in [\lambda, 1] \cap \Lambda, \\ \gamma' \in [\gamma, 1] \cap \Gamma}} L_i^{(2)}(\lambda', \gamma').$$

With these modifications to the loss functions, the results from the previous sections can be directly applied.

Multiple-stages We extend the problem to the scenario where there are $K > 2$ stages. Let $\theta \in \mathbb{R}^K$ denote the parameter of interest. At the k-th stage, the loss function for sample i is denoted as $L_i^{(\mathrm{k})}(\theta_{1:k})$, where $\theta_{1:k}$ is shorthand for $(\theta_1, \ldots, \theta_k)$. The sequential nature of the problem determines that loss function at stage k only involves parameters $\theta_{1:k}$. The goal is to determine parameter set θ that satisfies

$$\mathbb{P}\left(\forall k \in [K], L_i^{(\mathrm{k})}(\theta_{1:k}) \leq \alpha_k, \right) \geq 1 - \delta,$$

or alternatively,

$$\mathbb{E} L_i^{(\mathrm{k})}(\theta_{1:k}) \leq \alpha_k \quad \text{for all } k \in [K].$$

When loss functions at different stages are monotonic in each parameter θ_i with $i \in [K]$, one can develop analogous approaches to achieve risk control with high probability and expected risk control.

Contribution Statement

Yunpeng Xu and Mufang Ying contributed equally in this work.

References

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

[Angelopoulos *et al.*, 2021a] 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.

[Angelopoulos *et al.*, 2021b] 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.

[Angelopoulos *et al.*, 2023] A.N. Angelopoulos, K. Krauth, S. Bates, Y. Wang, and M.I. Jordan. Recommendation systems with distribution-free reliability guarantees. In *Symposium on Conformal and Probabilistic Prediction with Applications (COPA), 2023*, 2023.

[Angelopoulos *et al.*, 2024] Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *ICLR*, 2024.

[Baeza-Yates and Ribeiro-Neto, 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[Bretz *et al.*, 2009] Frank Bretz, Willi Maurer, Werner Brannath, and Martin Posch. A graphical approach to sequentially rejective multiple test procedures. *Statistics in medicine*, 28(4):586–604, 2009.

[Burges *et al.*, 2005] C. J. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, 2005.

[Burges *et al.*, 2006] Christopher J.C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *Proceedings of NIPS conference, 2006*, 2006.

[Cao *et al.*, 2007] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *MSR-TR-2007-40*, 2007.

[Chu and Ghahramani, 2005] W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, 2005.

[Crammer and Singer, 2001] Koby Crammer and Yoram Singer. Pranking with ranking. In *Proceedings of NIPS conference*, 2001.

[Freund *et al.*, 2003] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Journal of Machine Learning Research*, 2003.

[Guo *et al.*, 2016] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 39th International ACM SIGIR conference*, 2016.

[Guo *et al.*, 2023] Ruocheng Guo, Jean-François Ton, Yang Liu, and Hang Li. Inference-time stochastic ranking with risk control. *arXiv e-prints*, pages arXiv–2306, 2023.

[Järvelin and Kekäläinen, 2000] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. *Proceedings of the 23rd international ACM SIGIR conference*, 2000.

[Khattab *et al.*, 2020] Omar Khattab, Mohammad Hammoud, and Tamer Elsayed. Finding the best of both worlds: Faster and more robust top-k document retrieval. *Proceedings of the 43rd International ACM SIGIR Conference*, 2020.

[Lei *et al.*, 2015] Jing Lei, Alessandro Rinaldo, and Larry Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 2015.

[Liu, 2009] Tie-Yan Liu. Learning to rank for information retrieval. *Proceedings of the 33rd international ACM SIGIR conference*, 2009.

[Papadopoulos *et al.*, 2002] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *ECML*, 2002.

[Severyn and Moschitti, 2015] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference*, 2015.

[Stephen and K., 1976] Robertson. Stephen and Jones. K., Sparck. Relevance weighting of search terms. journal of the association for information science and technology. *27(3):129-146. doi: 10.1002/ASI.4630270302*, 1976.

[Vovk *et al.*, 1999] Vladimir Vovk, Alex Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. *Sixteenth International Conference on Machine Learning (ICML-1999)*, 1999.

[Vovk *et al.*, 2005] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[Wang and Joachims, 2023] Lequn Wang and Thorsten Joachims. Uncertainty quantification for fairness in two-stage recommender systems. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 940–948, 2023.

[Yin and et al, 2016] Dawei Yin and et al. Ranking relevance in Yahoo search. *Proceedings of the ACM SIGKDD Conference*, 2016.