PLOS ONE

RESEARCH ARTICLE

Distributionally robust learning-to-rank under the Wasserstein metric

Shahabeddin Sotudian[1], **Ruidi Chen**[1], **Ioannis Ch. Paschalidis**[1,2]*

1 Division of Systems Engineering, Department of Electrical and Computer Engineering, Boston University, Boston, MA, United States of America, **2** Department of Biomedical Engineering, and Faculty of Computing & Data Sciences, Boston University, Boston, MA, United States of America

* yannisp@bu.edu

Check for updates

OPEN ACCESS

Citation: Sotudian S, Chen R, Paschalidis IC. (2023) Distributionally robust learning-to-rank under the Wasserstein metric. PLoS ONE 18(3): e0283574. https://doi.org/10.1371/journal.pone.0283574

Editor: Kathiravan Srinivasan, Vellore Institute of Technology: VIT University, INDIA

Received: October 30, 2022

Accepted: March 12, 2023

Published: March 30, 2023

Copyright: © 2023 Sotudian et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The data that support the findings of this study are openly available in Cancer Cell Line Encyclopedia, Cancer Therapeutics Response Portal, and LETOR benchmark data sets. We provided the links in the manuscript. Please refer to the "Experiment setup - Data sets" Subsection.

Funding: This research was partially supported by the National Science Foundation (NSF) in the form of grants awarded to ICP (CCF-2200052, DMS-1664644, and IIS 1914792), the Office of Naval Research (ONR) in the form of a grant awarded to

Abstract

Despite their satisfactory performance, most existing listwise *Learning-To-Rank (LTR)* models do not consider the crucial issue of robustness. A data set can be contaminated in various ways, including human error in labeling or annotation, distributional data shift, and malicious adversaries who wish to degrade the algorithm's performance. It has been shown that *Distributionally Robust Optimization (DRO)* is resilient against various types of noise and perturbations. To fill this gap, we introduce a new listwise LTR model called *Distributionally Robust Multi-output Regression Ranking (DRMRR)*. Different from existing methods, the scoring function of DRMRR was designed as a multivariate mapping from a feature vector to a vector of deviation scores, which captures local context information and cross-document interactions. In this way, we are able to incorporate the LTR metrics into our model. DRMRR uses a Wasserstein DRO framework to minimize a multi-output loss function under the most adverse distributions in the neighborhood of the empirical data distribution defined by a Wasserstein ball. We present a compact and computationally solvable reformulation of the min-max formulation of DRMRR. Our experiments were conducted on two real-world applications: medical document retrieval and drug response prediction, showing that DRMRR notably outperforms state-of-the-art LTR models. We also conducted an extensive analysis to examine the resilience of DRMRR against various types of noise: Gaussian noise, adversarial perturbations, and label poisoning. Accordingly, DRMRR is not only able to achieve significantly better performance than other baselines, but it can maintain a relatively stable performance as more noise is added to the data.

Introduction

There exist many real-world applications such as recommendation systems, document retrieval, machine translation, and computational biology where the correct ordering of instances is of equal or greater importance than minimizing regression or classification errors [1]. *Learning-to-rank (LTR)* refers to a group of algorithms that apply machine learning techniques to tackle these ranking problems. Generally speaking, LTR methods learn a scoring function that maps an instance-query feature vector to a relevance score (i.e., multi-level

ICP (N00014-19-1-2571), the National Institutes of Health (NIH) in the form of grants awarded to ICP and the Boston University Clinical & Translation Science Institute (R01 GM135930, UL54 TR004130), and by Boston University funds. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

rating/label) that is then used to rank instances for a given query. Ideally, the resulting ranked list should maximize a ranking metric [2–4]. We considered two medical applications of LTR, namely *medical document retrieval* and *drug response prediction*. Healthcare applications commonly face various challenges including: (*i*) susceptibilities in data collection due to instrument and environmental noise or data entry errors; (*ii*) ambiguous or improper data annotation; (*iii*) lack of large-scale data for training and testing of algorithms; (*iv*) imbalanced data sets; (*v*) missing data; (*vi*) divergence of training and testing data distributions (e.g., data is recorded by different hospitals using different procedures); and, more importantly, (*vii*) the threat of adversarial attacks [5–7]. Consequently, robustness is critical for the wider adoption and deployment of algorithms into healthcare systems [7].

In this work, and without loss of generality, we take document retrieval as an example to explain the concepts and formulations. The main goal of document retrieval is to rank a set of documents by their relevance to a query. A slightly different example in computational biology is drug response prediction. For instance, prescribing the right therapeutic option for each cancer patient is an intricate task since the efficacy of cancer medications varies among patients. Nevertheless, the biological differences among cancers can be used to design genomic predictors of drug responses from large panels of cancer cell lines [8]. In drug response prediction, large-scale screenings of cancer cell lines against libraries of pharmacological compounds are used to predict precise and individualized medications [8].

Existing LTR approaches fall into three categories, namely pointwise, pairwise, and listwise [9]. The pointwise approach formulates ranking as a classification or regression problem—most early LTR algorithms such as linear regression ranking [9] or RankNet [10] take a very similar approach. In the pairwise approach, a classification method is employed to classify the preference order within document pairs. Representative pairwise ranking algorithms include RankBoost [11], RankNet [10], and ordinal regression [9]. Both approaches are misaligned with the ranking utilities such as Normalized Discounted Cumulative Gain (NDCG) and do not straightforwardly model the ranking problem. The listwise models can overcome this drawback by taking the entire list of retrieved documents from a query as instances and train a ranking function through the minimization of a listwise loss function. Experimental results show that the listwise approaches generally outperform the pointwise and pairwise algorithms [12]. The literature offers a variety of approaches from deriving a smooth approximation to ranking utilities (e.g., ApproxNDCG [13] and SoftRank [14]), to constructing differentiable surrogate loss functions (e.g., ListMLE [15], LambdaMART [16], and ListNet [12]). Specifically, ListNet and ListMLE try to learn the best document permutation based on permutation probabilities via the Plackett-Luce model while SoftRank and ApproxNDCG use ranking metrics or positions to tune their loss functions. On the other hand, LambdaMART employs heuristics to compute the gradients of an unknown loss function directly.

Most existing studies on LTR achieve impressive performance but often neglect the importance of ***robustness*** [9]. Systematic noise can become part of a data set in many ways and deceive LTR models to rank an item at an incorrect position with high confidence. While Empirical Risk Minimization (ERM) has been effective to optimize loss, ERM often does not yield models that are robust to adversarially crafted samples [17]. *Distributionally Robust Optimization (DRO)* is a modeling paradigm for data-driven decision-making under uncertainty. It has been successful in handling problems with corrupted training data through hedging against the most adverse distribution within a Wasserstein ball [18]. Recently, DRO has been an active area of research owing to its robustness to adversarial examples, rigorous out-of-sample and asymptotic consistency guarantees, and excellent empirical performance [19].

In the present work, we seek to infuse robustness into LTR problems through the DRO framework. Equipped with this perspective, we make the following contributions. Unlike

other LTR frameworks, our algorithm approaches listwise ranking in a novel way and employs ranking metrics (i.e., NDCG) in its *output*. In particular, we use the notion of *position deviation* to define a vector of relevance scores instead of a scalar. We then adopt the DRO framework to minimize a worst-case expected multi-output loss function over a probabilistic ambiguity set that is defined by the Wasserstein metric. To the best of our knowledge, ours is the first study that utilizes a multi-output Wasserstein DRO framework to robustify LTR problems. We present an equivalent convex reformulation of the DRO problem, which is shown to be tighter than earlier work [18]. In experiments, our approach yields state-of-the-art results in two challenging applications of LTR, namely medical document retrieval and drug response prediction. More importantly, we evaluate our model to verify its robustness against various types of attacks including adversarial attacks and label attacks, showing that our model maintains a consistently good performance under various attack scenarios.

Notational conventions

We use boldfaced lowercase letters to denote vectors, ordinary lowercase letters to denote scalars, boldfaced uppercase letters to denote matrices, and calligraphic capital letters to denote sets. All vectors are column vectors. For space saving reasons, we write $\mathbf{x}$ to denote the column vector $(x_1, \ldots, x_{\dim(x)})$, where $\dim(\mathbf{x})$ is the dimension of $\mathbf{x}$. We use prime to denote the transpose, N for the set $\{1, \ldots, N\}$ for any integer N, $\|\cdot\|_p$ for the ℓ_p norm with $p \geq 1$, and $\mathbf{I}_K$ for the K-dimensional identity matrix. For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, we use $\|\mathbf{A}\|_p$ to denote its induced ℓ_p norm, defined as $\|\mathbf{A}\|_p \triangleq \sup_{\mathbf{x}\neq\mathbf{0}} \|\mathbf{A}\mathbf{x}\|_p / \|\mathbf{x}\|_p$.

Preliminaries

Learning-to-rank

In a ranking problem, the data consists of a set of triples (query, document, relevance score). A feature vector is used to represent a query-document pair. The relevance score indicates the degree of relevance of this document to its corresponding query. Given a ranking data set $\{(\mathbf{X}^q, \boldsymbol{\theta}^q)\}_{q=1}^T$, $q \in T$ indexes a query, and $\mathbf{X}^q$ and $\boldsymbol{\theta}^q$ represent the list of retrieved documents and corresponding relevance scores, respectively. The q-th query contains n_q documents and $\mathbf{X}^q \in \mathbb{R}^{n_q\times p}$ has rows $(\mathbf{x}_1^q, \cdots, \mathbf{x}_{n_q}^q)$, each of which is a p-dimensional document feature vector. The vector $\boldsymbol{\theta}^q = (\theta_1^q, \cdots, \theta_{n_q}^q) \in \mathbb{R}_+^{n_q}$ contains the corresponding ground-truth relevance scores, where a higher $\theta_d^q \in \mathbb{R}$ indicates that the document with features $\mathbf{x}_d^q$ is more relevant. In the learning-to-rank framework, denoting by $\mathbf{x}$ and θ the random variables that represent the document feature vector and relevance score, respectively, the goal is to learn a scoring function f that best predicts the relevance score:

$$\min_f \mathcal{L}(f) \triangleq \mathbb{E}^{\mathbb{P}^*}[\ell(\theta, f(\mathbf{x}))], \tag{1}$$

where $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function, $f : \mathbb{R}^p \to \mathbb{R}$ predicts the relevance score of each document, and $\mathbb{P}^*$ is the underlying true probability distribution of $(\mathbf{x}, \theta)$. Given that $\mathbb{P}^*$ is unknown, most existing LTR algorithms solve (1) through estimating the expected loss by its empirical substitute (2):

$$\hat{\mathcal{L}}(f) \triangleq \frac{1}{\sum_q n_q} \sum_{q=1}^{T} \sum_{d=1}^{n_q} \ell\left(\theta_d^q, f(\mathbf{x}_d^q)\right). \tag{2}$$

For a test query $\mathbf{X}^t \in \mathbb{R}^{n_t\times p}$ consisting of n_t documents, the final predicted ranking list $\hat{\boldsymbol{\pi}}$ is

simply obtained by ranking the rows in $\mathbf{X}^t$ based on their inferred ranking scores $\hat{\boldsymbol{\theta}}^t = (f(\mathbf{x}_1^t), \ldots, f(\mathbf{x}_{n_t}^t))$. Eq (2) is restrictive in the sense that: (*i*) it does not take into account the inter-dependency of scores between documents, and (*ii*) the empirical estimate is very sensitive to data perturbations.

Distributionally Robust Optimization

Distributionally Robust optimization (DRO) hedges against a set of probability distributions instead of just the empirical distribution. DRO minimizes a worst-case loss over a probabilistic ambiguity set:

$$\min_f \max_{\mathbb{Q}\in\Omega} \mathbb{E}^{\mathbb{Q}}[\ell(\theta, f(\mathbf{x}))],$$

where the ambiguity set Ω can be defined through moment constraints [20], or as a ball of distributions using some probabilistic distance function such as the Wasserstein distance [21, 22]. The Wasserstein DRO model has been extensively studied in the machine learning community; see, for example, [23, 24] for robustified regression models, [19] for adversarial training in neural networks, and [25] for distributionally robust logistic regression. These works, [18, 26, 27] provided a comprehensive analysis of the Wasserstein-based distributionally robust statistical learning framework.

Problem formulation

Next, we introduce our DRO formulation of the LTR problem. Different from the existing works where a univariate relevance score $\theta_d^q \in \mathbb{R}$ is used for each document $\mathbf{x}_d^q \in \mathbb{R}^p$, we define a Ground Truth Deviation vector $\boldsymbol{\theta}_d^q \in \mathbb{R}^K$ to characterize different levels of importance for the document $\mathbf{x}_d^q$ in the q-th query. Here, K is a constant to be defined later (cf. end of the next section). We also derive an equivalent reformulation of the DRO problem.

Ground Truth Deviation

As a popular evaluation criterion in information retrieval, Normalized Discounted Cumulative Gain (NDCG) can deal with cases that have more than two degrees of relevancy for documents [28]. Let $D(s) = 1/\log(1 + s)$ be a discount function, $G(s) = s$, a monotonically increasing gain function, and $\mathcal{Z}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ a set of documents ordered according to their ground-truth rank, with $\mathbf{x}_i$ and y_i being a document feature vector and a relevance score, respectively. Assume $\tilde{\mathcal{Z}}_n$ is a (predicted) ranked list for $\mathcal{Z}_n$; then the *Discounted Cumulative Gain (DCG)* of $\tilde{\mathcal{Z}}_n$ is defined as $\Phi(\tilde{\mathcal{Z}}_n) = \sum_{r=1}^n G(y_{\pi_r})D(r)$, where π_r is the index of the document ranked at position r of $\tilde{\mathcal{Z}}_n$. The reason for introducing the discount function is that the user cares less about documents ranked lower [29]. NDCG normalizes DCG by the Ideal DCG (IDCG), $\Phi^I(\mathcal{Z}_n)$, which is the DCG score of the ideal ranking result [30] and can be computed by $\Phi^N(\tilde{\mathcal{Z}}_n) = \Phi(\tilde{\mathcal{Z}}_n)/\Phi^I(\mathcal{Z}_n) \in [0, 1]$. Considering the q-th query $(\mathbf{X}^q, \mathbf{y}^q)$ that contains n_q documents, we define a *Ground Truth Deviation (GTD)* vector for document d as follows:

$$\boldsymbol{\theta}_d^q = \xi_I(\boldsymbol{\xi}_D \circ \boldsymbol{\xi}_\Phi), \tag{3}$$

where $\circ$ is the Hadamard product (a.k.a. the element-wise product). The vector $\boldsymbol{\theta}_d^q$ is comprised of the following three components.

NDCG deviation score ($\boldsymbol{\xi}_\Phi$). To compute this vector, first, the elements of $\mathbf{y}^q = (y_1^q, \ldots, y_{n_q}^q)$ are sorted in descending order of their ground truth individual relevance scores,

and the document feature vectors $\mathbf{X}^q = (\mathbf{x}_1^q, \ldots, \mathbf{x}_{n_q}^q)$ are also sorted correspondingly. We denote them by $\bar{\mathbf{y}}^q$ and $\bar{\mathbf{X}}^q$, respectively. The NDCG score for $\bar{\mathbf{X}}^q$ is equal to 1. If we switch two documents in $\bar{\mathbf{X}}^q$, the NDCG will decrease or in some cases may stay the same (i.e., if their relevance scores are equal). For document d in query q, we define the NDCG deviation score vector as $\boldsymbol{\xi}_\Phi = (\lambda_{d1}, \ldots, \lambda_{dn_q})$ where λ_{di} is the NDCG score of $\bar{\mathbf{X}}^q$ when we switch the ***position*** of document d with the ***document*** that is in i-th position of $\bar{\mathbf{X}}^q$ and can be formulated as follows:

$$\lambda_{di} = 1 + \frac{\dfrac{y_d - y_{\pi_i}}{\log(1+i)} + \dfrac{y_{\pi_i} - y_d}{\log(1+\pi_d^{-1})}}{\Phi^I}.$$

Here, π_d^{-1} is the position of the document d in $\bar{\mathbf{X}}^q$, π_i is the index of the document ranked at the i-th position of $\bar{\mathbf{X}}^q$, and Φ^I is the IDCG. The details about the derivation can be found in the S1 Appendix. We can perceive the i-th element of the GTD vector as a score that indicates the degree of congruence between a document and the i-th rank.

Position deviation score ($\boldsymbol{\xi}_D$). This vector is defined to further push the relevant documents to the top of the ranking list and penalize documents based on their position in the ranking list. The position deviation score works in conjunction with $\boldsymbol{\xi}_\Phi$. We define it as $\boldsymbol{\xi}_D = (\rho_{d1}, \ldots, \rho_{dn_q})$ where ρ_{di} can be calculated by

$$\rho_{di} = \frac{\alpha}{\sqrt{\left|cosh\left(min\left(\beta h_{di}, \frac{\beta}{2} h_{di}\right)\right)\right|}},$$

where $h_{di} = \pi_d^{-1} - i$. As can be seen in Fig 1, α specifies the GTD's maximum score and β regulates the magnitude of the penalty for a position deviation. Here, we use the red dashed curve for positive deviations (i.e., when a document ranked higher than its optimal position) and the black curve for negative deviations. This would induce our model to tolerate a positive deviation more than a negative one. Consequently, the model pushes the relevant documents to the top of the ranking list.

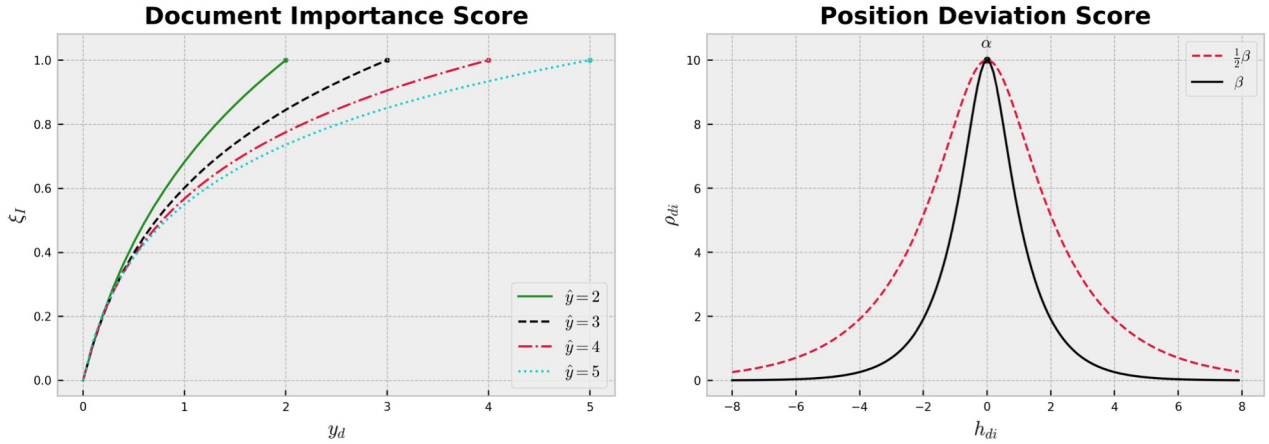

Fig 1. GTD graphs. (a) Position deviation score where $\alpha = 10$ and $\beta = 2$. (b) Document importance score for various maximum possible relevance scores.

https://doi.org/10.1371/journal.pone.0283574.g001

Document importance score (ξ_I). It is defined to place greater emphasis on highly relevant documents and can be computed as

$$\xi_I = \frac{\log(\hat{y}y_d + 1)}{\log(\hat{y}^2 + 1)},$$

where $\hat{y}$ is the maximum possible value for relevance scores. Fig 1 presents ξ_I for different values of $\hat{y}$.

Ultimately, instead of a relevance score for each document, we have a GTD vector. The GTD vector characterizes different levels of importance for a document in a query where the first element is the first level of importance, the second element is the second level of importance, and so on. Since each query may have a different number of documents, we just consider the first K elements of $\boldsymbol{\xi}_\Phi$ and $\boldsymbol{\xi}_D$ in our model, corresponding to K levels of importance. In this way, all GTD vectors are of the same length. We prefer to use a low value for K since it forces the model to focus on the most relevant documents. In case a large K needs to be used and $K > n_q$, we can simply repeat the last element of $\boldsymbol{\xi}_\Phi$ and $\boldsymbol{\xi}_D$ to pad our $\boldsymbol{\xi}_\Phi$ vector.

In a nutshell, the *NDCG deviation score* ($\boldsymbol{\xi}_\Phi$) captures the relative position of a document in a query. On the other hand, the *position deviation score* ($\boldsymbol{\xi}_D$) and the *document importance score* (ξ_I) work in conjunction to push the relevant documents to the top of the list. We used an asymmetric bell-shaped function for the position deviation score to give a maximum score to correctly ranked documents. By using a "steeper left fall," we give a lower score to a negative position deviation (i.e., when a document ranked lower than it should) compared to a positive one. Moreover, α and β enable us to control the maximum score and the magnitude of the penalty for a position deviation, respectively. In the S1 Appendix, we present an ablation study to gauge their effect on performance. We also provid an example of GTD vector calculation.

Distributionally Robust Multi-output Regression

Consider a setting where there are K levels of importance with features and importance scores distributed according to $\mathbf{x} \in \mathbb{R}^p$ and $\boldsymbol{\theta} \in \mathbb{R}^K$, respectively. We restrict our attention to linear function classes by assuming $\mathbf{f}(\mathbf{x}) = \mathbf{B}'\mathbf{x}$ where $\mathbf{B} \in \mathbb{R}^{p\times K}$. The matrix $\mathbf{B}$ characterizes the dependency structure of the different levels of importance. Nonlinearity can be introduced by applying a transformation (e.g., kernel function) on the feature $\mathbf{x}$. The Distributionally Robust Multi-output Regression Ranking (DRMRR) formulation minimizes the worst-case expected loss as follows:

$$\min_{\mathbf{B}} \max_{\mathbb{Q}\in\Omega} \mathbb{E}^{\mathbb{Q}}[\ell(\boldsymbol{\theta} - \mathbf{B}'\mathbf{x})], \tag{4}$$

where $\ell : \mathbb{R}^K \to \mathbb{R}$ is a Lipschitz continuous loss function on the metric spaces $(\mathcal{D}, \|\cdot\|_r)$ and $(\mathcal{C}, |\cdot|)$, where $\mathcal{D}, \mathcal{C}$ are the domain and co-domain of $\ell(\cdot)$, respectively. In (4), $\mathbb{Q} \in \Omega \triangleq \{\mathbb{Q} \in \mathcal{P}(\mathcal{S}) : W_1(\mathbb{Q}, \hat{\mathbb{P}}_N) \leq \varepsilon\}$ is the probability distribution of $(\mathbf{x}, \boldsymbol{\theta})$, where $\mathcal{P}(\mathcal{S})$ is the space of all probability distributions supported on $\mathcal{S}$ and $\mathcal{S}$ is the uncertainty set of $(\mathbf{x}, \boldsymbol{\theta})$, ε is a positive constant (i.e., Wasserstein ball radius), $\hat{\mathbb{P}}_N$ is the empirical distribution that assigns an equal probability to all N training samples, with $N = \sum_{j=1}^{T} n_j$, where T is the number of queries, and $W_1(\mathbb{Q}, \hat{\mathbb{P}}_N)$ is the order-1 Wasserstein distance between $\mathbb{Q}$ and $\hat{\mathbb{P}}_N$ defined as

$$W_1(\mathbb{Q}, \hat{\mathbb{P}}_N) \triangleq \min_{\Pi\in P(\mathcal{S}\times\mathcal{S})} \left\{ \int_{\mathcal{S}\times\mathcal{S}} \delta(\mathbf{z}_1 - \mathbf{z}_2)\Pi(d\mathbf{z}_1, d\mathbf{z}_2) \right\}.$$

In the distance, $\delta(\mathbf{z}_1 - \mathbf{z}_2) \triangleq \|\mathbf{z}_1 - \mathbf{z}_2\|_r$ with $\mathbf{z}_i = (\mathbf{x}_i, \boldsymbol{\theta}_i)$, $i = 1, 2$, drawn from $\mathbb{Q}$ and $\hat{\mathbb{P}}_N$,

respectively, and Π specifies the joint distribution of $\mathbf{z}_1$ and $\mathbf{z}_2$ with marginals $\mathbb{Q}$ and $\hat{\mathbb{P}}_N$. Note that the same norm is used to define the Wasserstein metric and the domain space of $\ell(\cdot)$. In the following theorem we propose an equivalent reformulation of (4) by using duality for the inner maximization problem.

Theorem 0.1. *Suppose our dataset consists of T queries* $\{(\mathbf{X}^q, \mathbf{\Theta}^q)\}_{q=1}^T$ *and each query q contains* n_q *documents,* $q \in T$, *where* $\mathbf{X}^q \in \mathbb{R}^{n_q \times p}$ *is the document feature matrix with rows* $\mathbf{x}_d^q \in \mathbb{R}^p$, $d \in n_q$, *and* $\mathbf{\Theta}^q \in \mathbb{R}^{n_q \times K}$ *is the GTD matrix with rows* $\boldsymbol{\theta}_d^q \in \mathbb{R}^K$. *Define a loss function* $\ell(\cdot) \triangleq \|\cdot\|_r$. *If the Wasserstein metric is induced by* $\|\cdot\|_r$, *the DRMRR problem (4) can be equivalently reformulated as*:

$$\min_{\mathbf{B}} \frac{1}{\sum_{e=1}^T n_e} \sum_{q=1}^T \sum_{d=1}^{n_q} \|\theta_d^q - \mathbf{B}'\mathbf{x}_d^q\|_r + \varepsilon \|\tilde{\mathbf{B}}'\|_s, \tag{5}$$

where $r, s \geq 1$; $1/r + 1/s = 1$; $\tilde{\mathbf{B}} = (-\mathbf{B}', \mathbf{I}_K)$.

The proof can be found in S1 Appendix. Thm. 0.1 establishes a connection between distributional robustness and regularization, which has also been studied by, e.g., [22, 25, 26]. However, most of the existing studies focused on a univariate output. By contrast, our work adapts the DRO framework to a multi-output setting, which is more suitable for the ranking problem. Recently, [18] studied a multi-output regression problem under the Wasserstein DRO framework. However, our results in Theorem 0.1 present a tighter reformulation than theirs (Eq. (6.2) in [18]. In the case where the Wasserstein metric is induced by the ℓ_2 norm ($r = 2$), Eq (5) yields a regularizer which is the spectral norm (largest singular value) of $\tilde{\mathbf{B}}'$, while [18] derived a regularizer in the Frobenius norm which is looser.

Score calculation

Suppose we are given a test query $\mathbf{X}^t = (\mathbf{x}_1^t{}', \ldots, \mathbf{x}_{n_t}^t{}') \in \mathbb{R}^{n_t \times p}$; we can estimate the GTD matrix as $\hat{\mathbf{\Theta}}^t = (\mathbf{B}'\mathbf{x}_1^t, \ldots, \mathbf{B}'\mathbf{x}_{n_t}^t) \in \mathbb{R}^{n_t \times K}$. In the matrix $\hat{\mathbf{\Theta}}^t$, columns correspond different ranks and rows refers to different documents. Algorithm 1 demonstrates the procedure of ranking using the output of the DRMRR algorithm where $R_K(j)$ is the remainder of dividing j by K. In the S1 Appendix, we present an intuitive toy example to illustrate this algorithm better.

Algorithm 1: **Scoring Procedure for DRMRR**

```
Input: Θ̂^t
Output: Sorted list
Let δ = 1;
for j = 1 to n_t
  Find the maximum of δ-th column of Θ̂^t;
  Assign the corresponding row/document to rank j;
  Remove the corresponding row/document;
  if R_K(j) = 0 then
    δ = 1;
  else
    δ = δ + 1;
  end if
end for
```

Experimental results

Experiment setup

Data sets. We conducted experiments on two publicly available benchmark datasets: OHSUMED(https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/), and *Drug Response Prediction (DRP)*(https://modac.cancer.gov/assetDetails?dme_data_id=NCI-DME-MS01-8088592). As a subset of the MEDLINE database (a database on medical publications), the OHSUMED corpus [31] consists of about 0.3 million records from 270 medical journals from 1987 to 1991. A query set with 106 queries on the OHSUMED corpus has been extensively used in previous works, in which each query is represented by 45 features [2]. There are in total 16,140 query document pairs with relevance judgments. LETOR [2] defined three ratings 0, 1, 2, corresponding to "irrelevant," "partially relevant," and "definitely relevant," respectively. In addition to OHSUMED, we trained and evaluated our method using the cell line data and drug sensitivity data from the *Cancer Cell Line Encyclopedia (CCLE)* [32] and the *Cancer Therapeutics Response Portal (CTRP v2)* [33]. A total of 332 cell lines (i.e., queries) and 50 drug responses were used. The "Act Area" (the area above the fitted dose-response curve) was used to quantify drug sensitivity where a lower response value indicates higher drug sensitivity. After several pre-processing steps, cell lines are represented by 251 numeric features (i.e., genes) and drug sensitivities are labeled with graded relevance from 0 to 2 (i.e., "insensitive," "sensitive," and "highly sensitive," respectively) with larger labels indicating a higher sensitivity. Further details of the data pre-processing steps can be found in S1 Appendix. Moreover, all code written in support of this publication is publicly available on a GitHub repository(https://github.com/noc-lab/DRMRR-Distributionally-robust-learning-to-rank-under-the-Wasserstein-metric). Please note that we targeted *biomedical applications with limited data*. Since the number of drug-cell line pairs is much less than the number of features, most approaches "overfit." Similarly, OHSUMED challenges ranking models due to its small sample size.

Evaluation metrics. We evaluated model performance using two metrics: NDCG@k and AP@k. NDCG@k is the top-k version of NDCG, where the discount function is $D(s) = 0$ for $s > k$. Precision at position k (P@k) is the fraction of relevant documents in the top-k. Suppose we have binary relevance for the documents in a q-query; we define P@k as $P@k = \frac{1}{k}\sum_{j=1}^{k} 1(y_{\pi_j} = 1)$ where $1(\cdot)$ is the indicator function. We define Average Precision at position k as $AP@k = \frac{1}{m}\sum_{j=1}^{k} P@j \times 1(y_{\pi_j} = 1)$, where m is the total number of relevant documents in the top-k of the ranking list. AP is a highly localized performance measure and captures the quality of rankings for applications where only the first few results matter. The main difference between AP and NDCG is that NDCG differentiates between "partially relevant" and "definitely relevant" documents while AP treats them equally. Given a set of testing queries and a performance metric, we are interested in the mean metric which is simply the mean of the performance metric for all queries. From now on, we use NDCG@k and AP@k to denote mean NDCG@k and mean AP@k, respectively.

Competing methods. Although the list of published LTR algorithms is endless, Lambda-MART_{MAP} [16], $\text{LambdaMART}_{\text{NDCG}}$ [16], and $\text{XE-MART}_{\text{NDCG}}$ [4] have been demonstrated repeatedly to outperfrom other algorithms including RankNet [10], Coordinate Ascent [34], ListNet [12], Random Forests [35], BoltzRank [36], ListMLE [15], Position-Aware ListMLE [37], RankBoost [11], AdaRank [38], SoftRank [14], ApproxNDCG [13], ApproxAP [13], and several direct optimization methods [39, 40]. Moreover, multiple comparative studies [41–43] reported that tree-based models exhibit top performance in drug response prediction.

Thus, we rely on prior research [4, 41, 44] and do not include the weaker methods in our experiments. It is important to note that the author of XE-MART$_{NDCG}$ proposed this model as a *robust alternative* to LambdaMART-based models. We also compared DRMRR against the state-of-the-art transformer-based neural ranking model [45] with different loss functions. However, since the performance of the aforementioned tree-based baselines was by far better than the latter (especially on our main application, namely DRP), we defer the presentation of the performance of the latter methods to the S1 Appendix.

Experimental settings and hyper-parameter optimization. In our experiments, we used the standard supervised LTR framework [9]. Authors of LETOR [2] partitioned the OHSUMED data set into five parts for five-fold cross-validation where three parts were used for training, one part for validation (i.e., tuning the hyperparameters of the learning algorithms), and the remaining part for evaluating the performance of the learned model. Similarly, we partitioned the drug response data set into five folds and conducted five-fold cross-validation to train, validate, and evaluate the ranking algorithms. In all experiments, the average on the test set over the 5 folds was reported. Algorithm parameters were tuned on the validation sets. We optimized the algorithm parameters to maximize NDCG@5 and NDCG@10. The details of the parameter-tuning procedure and the optimal parameters for each algorithm can be found in the S1 Appendix.

Overall comparison

We compared the performance of DRMRR on OHSUMED, and DRP data sets with baseline methods introduced in the previous sections. The results are in Table 1. The values inside the parentheses denote the *Standard Deviation (SD)* of the corresponding metrics. Bold numbers indicate the best performance among all methods for each metric. DRMRR consistently outperforms all baseline methods across all metrics. In our experiment on OHSUMED data, LambdaMART$_{NDCG}$ demonstrated a reasonably good overall performance and it is the second-best method. However, XE-MART$_{NDCG}$ was the second-best method in our experiment on the DRP data. The difference between the best and the second-best methods for the DRP data set is greater than what we obtained for OHSUMED. Due to the limited number of samples available and the specific structure of the DRP data, the performance of the baseline methods diminished significantly. On the other hand, DRMRR was able to maintain its high performance. To sum up, the proposed method is not only able to push the most relevant documents (or sensitive drugs) to the top of the ranking list, but it can put them in the right order. Furthermore, as we discuss in the Supplement, our model is more efficient (low model complexity) and generalizes better (typically, the generalization error increases with model complexity).

Table 1. Performance comparison of ranking methods.

	Algorithms	NDCG@5	NDCG@10	AP@5	AP@10
OHSUMED	LambdaMART$_{MAP}$	45.18% (5.07%)	43.65% (3.55%)	67.94% (7.26%)	64.12% (5.83%)
	LambdaMART$_{NDCG}$	46.17% (5.91%)	44.40% (5.00%)	68.53% (8.40%)	65.25% (6.47%)
	XE-MART$_{NDCG}$	44.31% (6.58%)	44.79% (5.65%)	65.25% (8.08%)	62.41% (6.76%)
	DRMRR	**47.79% (6.58%)**	**45.36% (4.84%)**	**70.84% (7.35%)**	**65.31% (7.35%)**
DRP	LambdaMART$_{MAP}$	58.11% (1.90%)	63.39% (2.17%)	83.27% (1.57%)	77.25% (1.07%)
	LambdaMART$_{NDCG}$	58.73% (2.54%)	62.87% (2.83%)	83.07% (1.99%)	76.95% (1.19%)
	XE-MART$_{NDCG}$	59.37% (1.92%)	63.51% (2.18%)	83.70% (1.28%)	77.43% (1.34%)
	DRMRR	**68.40% (1.74%)**	**71.27% (1.78%)**	**85.03% (1.10%)**	**81.03% (1.00%)**

https://doi.org/10.1371/journal.pone.0283574.t001

Robustness comparison

In this section, we empirically study the behavior of DRMRR in the presence of noise. While our overall performance analysis suggested that DRMRR should be the most "well-behaved" of the four, that analysis was performed on the clean data. The robustness of a ranking model to noise is crucial in practice, especially in the healthcare domain. We put this hypothesis to test through four types of experiments. We conducted all experiments on the OHSUMED data set since it is a popular and standard LTR data set. In all experiments, the values are the average of 5 folds.

Gaussian noise attack. We added Gaussian noise to the test documents to deliberately corrupt them; therefore, depreciating their predictability. Gaussian noise was added to 75% of the test queries randomly. Experiments were conducted using various means and a fixed standard deviation of 0.001. We used the perturbed test data to evaluate the trained models (i.e., all algorithms were tested on the same perturbed test data). Fig 2 demonstrates the performance of the algorithms on the perturbed test data. Two observations are in order: (*i*) DRMRR outperformed the baseline models at different levels of noise; and (*ii*) DRMRR demonstrated a relatively stable performance.

Universal adversarial perturbation attack. We built an adversarial model to introduce perturbations that break the neighborhood relationships by altering the input slightly. To that

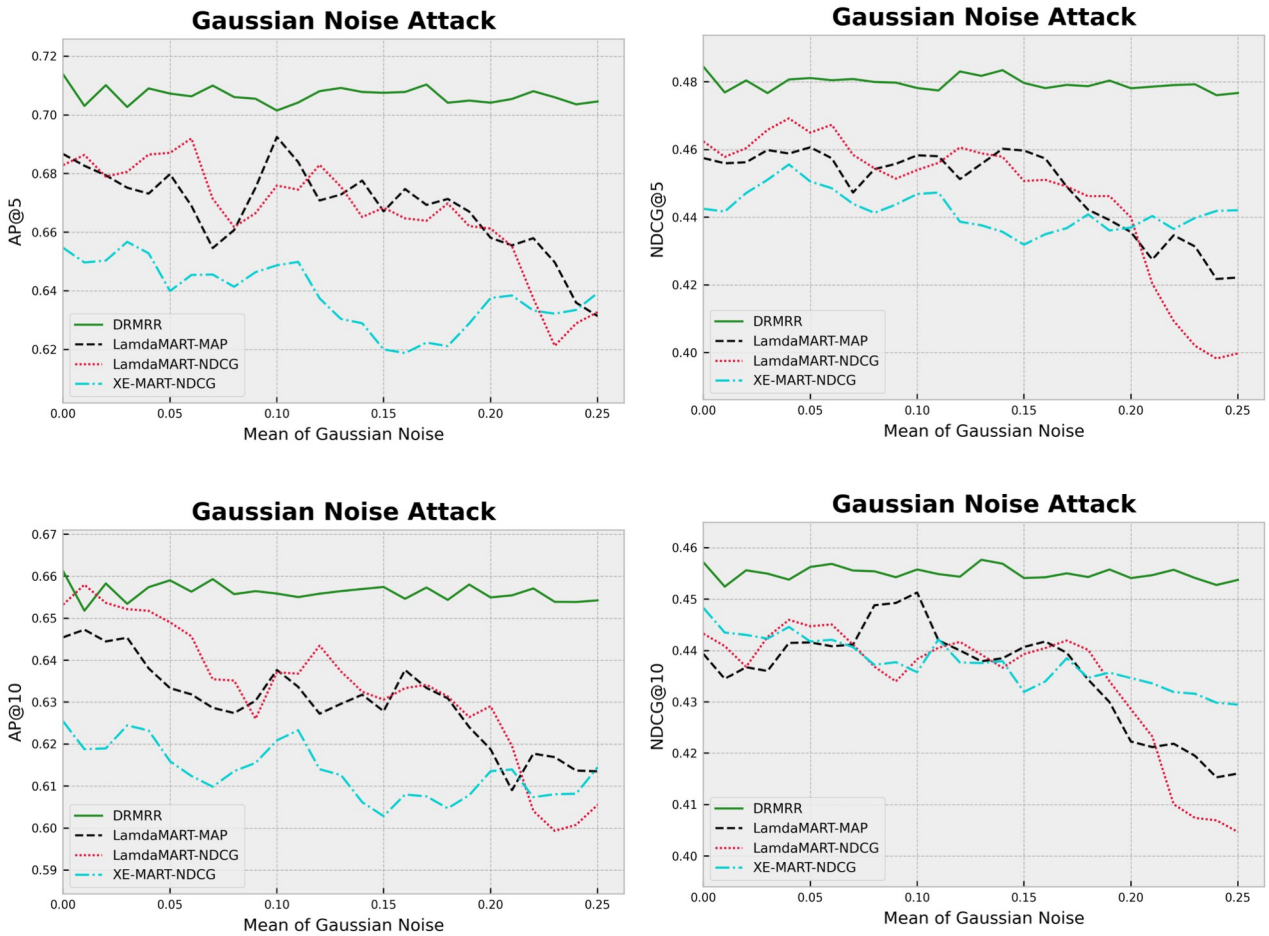

Fig 2. The impact of Gaussian noise on the performance of ranking models.

https://doi.org/10.1371/journal.pone.0283574.g002

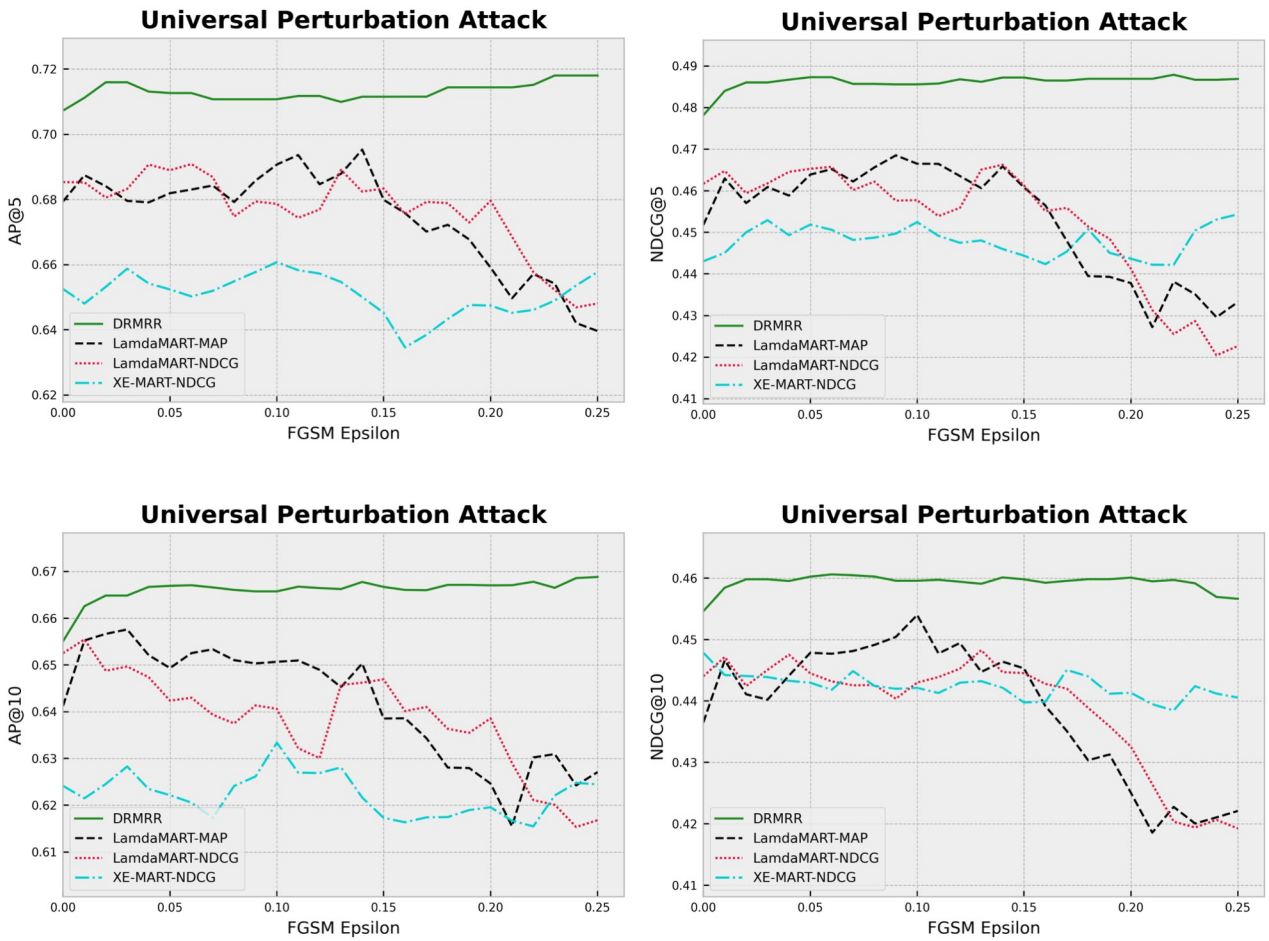

Fig 3. The impact of universal adversarial perturbation on the performance of ranking models.

https://doi.org/10.1371/journal.pone.0283574.g003

end, a pointwise linear regression ranking model was trained as the adversarial model on the clean training set. Then, 75% of the test queries were perturbed using the coefficient of the adversarial model and the Fast Gradient Sign Method (FGSM) method: $\bar{\bar{\mathbf{x}}}_q^d = \mathbf{x}_q^d + \sigma \cdot \mathrm{sgn}(\nabla_{\mathbf{x}_q^d} J(\mathbf{x}_q^d, y_q^d))$, where $\bar{\bar{\mathbf{x}}}_q^d$ is the perturbed feature vector, σ controls the magnitude of the perturbations, and J is the cost function of the adversarial model [46].

All algorithms that we trained in the "Overall Comparison" section were evaluated on the same perturbed test data. In this case, the adversary had no knowledge of the ranking models; however, it was trained on the same training data. Fig 3 shows the performance of the algorithms on the perturbed test data. As we increase the level of perturbations (i.e., σ), we can see that DRMRR is less sensitive to adversarial perturbations in comparison with the competing methods. It demonstrated a stable performance across all metrics. Among the baselines, XE-MART$_{\mathrm{NDCG}}$ that performed well in terms of NDCG@5, demonstrated poor performance in terms of AP@5.

Black-box adversarial attack. The black-box adversarial attack restricts the attacker's knowledge only to the deployed model [47]. The setting of black-box attacks is closer to the real-world scenario; therefore, this is the most practical experiment to measure the robustness of our algorithm. Please refer to [47] for more information on the black-box adversarial attacks. Since the adversary has no access to the model's weights and parameters, the adversary

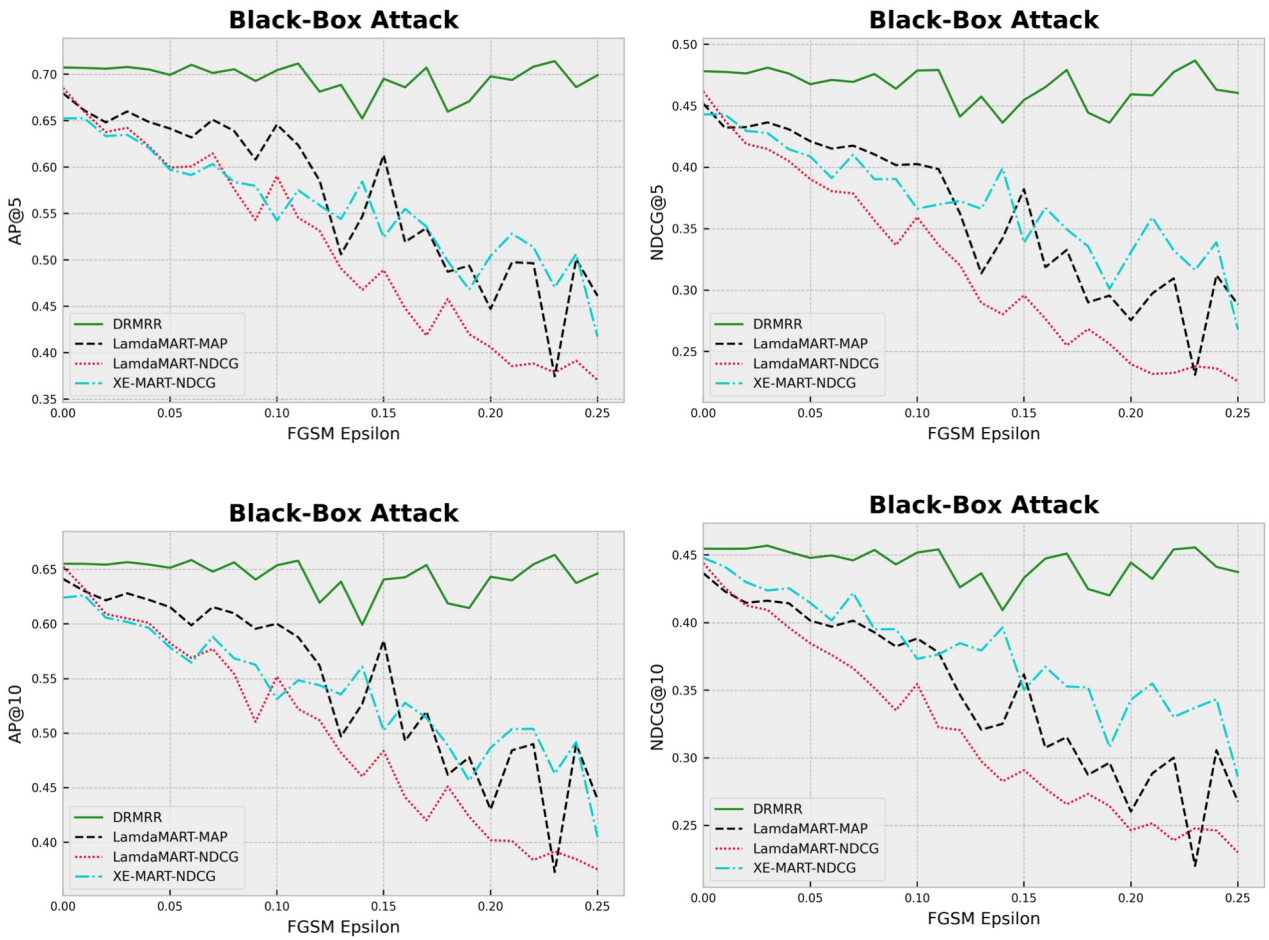

Fig 4. The impact of black-box adversarial perturbation on the performance of ranking models.

https://doi.org/10.1371/journal.pone.0283574.g004

can choose to train a parallel model called a *substitute model* to imitate the original model. Here, we use a four layers fully connected network as our substitute models (see the S1 Appendix for more details). To construct the substitute models, the training data were independently fed to each model and the output was observed. Then, for each algorithm, a Neural Network (NN) ranking model was trained as an adversarial substitute model using the training feature vectors and the observed output of that specific algorithm. Subsequently, 75% of the test queries were perturbed using the FGSM method and the parameters of the substitute model corresponding to each algorithm. We trained four substitute models corresponding to each ranking algorithm. We used the specific perturbed test data to evaluate the best trained models (i.e., each model has a different perturbed test set). Fig 4 demonstrates the performance of the algorithms on the perturbed test data. The values are the average of 5 folds. We can see that both figures show the same trend—increasing the level of perturbations (i.e., σ) leads to significant differences between DRMRR and the baselines methods. The competing methods were greatly affected by this type of noise, whereas their performance was modest in the simpler experiments, namely universal adversarial attack and Gaussian attack. We conclude that DRMRR is robust to adversarial perturbations, an important property that leads to good generalization ability.

Table 2. The error probability table.

$P(y_i^q \rightarrow y_j^q)$		y_j^q		
		0	**1**	**2**
y_i^q	**0**	e	$\frac{2}{3}(1-e)$	$\frac{1}{3}(1-e)$
	1	$\frac{1}{2}(1-e)$	e	$\frac{1}{2}(1-e)$
	2	$\frac{1}{3}(1-e)$	$\frac{2}{3}(1-e)$	e

https://doi.org/10.1371/journal.pone.0283574.t002

Label attack. In practice, the vagueness of query intent, insufficient domain knowledge, and ambiguous definition of relevance levels make it hard for human judges to assign proper relevance labels to some documents. Practically speaking, the probability of judgment errors in various relevance degrees is not equal. Even if human annotators misjudge a document, they are more probable to label it closer to its ground-truth label. Inspired by [48], we define the non-uniform error probabilities in Table 2 where entries of this table correspond to the probability that a document with ground-truth label y_i^q is prone to be labeled as y_j^q. We randomly changed the labels of the training data using the probabilities in Table 2. Then, each model was trained on the noisy training data. Clean test data were used to evaluate each model. We conducted two sets of experiments, namely low label noise (i.e., $e = 0.85$) and high label noise (i.e., $e = 0.7$). Table 3 reports the performance of the algorithms on the clean test data. The values in these figures are the average of 5 folds. For the low noise scenario, the differences between the average AP@5 and NDCG@5 of the baseline models and DRMRR were 2.53% and 1.59%, respectively. Notably, the gaps were even larger for the high noise scenario (AP@5 = 3.08%, NDCG@5=1.68%). Since noise in human-labeled data is an inevitable issue, we can argue that the baseline models are susceptible and degrade more severely as more noise is added to the training set.

Discussion and conclusion

This paper went beyond conventional listwise learning-to-rank approaches and introduced a distributionally robust learning-to-rank framework with multiple outputs, referred to as DRMRR. Unlike existing methods, the scoring function in DRMRR was designed as a multivariate mapping from a feature vector to a vector of deviation scores (a.k.a. GTD vector). The GTD vector captures local context information and cross-document interactions. Moreover, we formulated DRMRR as a min-max problem where one minimizes a worst-case expected loss over a probabilistic ambiguity set. The ambiguity set was defined as a ball of distributions using the Wasserstein metric. Notably, we presented a compact and computationally solvable equivalent reformulation of the min-max formulation of DRMRR. We compared DRMRR with the baseline models in terms of: (*a*) the overall performance on two real-world applications and (*b*) the robustness to various types and degrees of noise. In medical document

Table 3. The impact of label noise on the performance of ranking models.

	AP@5		NDCG@5		AP@10		NDCG@10	
	High	**Low**	**High**	**Low**	**High**	**Low**	**High**	**Low**
DRMRR	**70.25%**	**69.30%**	**48.43%**	**47.76%**	**66.59%**	**65.18%**	**46.67%**	**45.44%**
LambdaMART$_{MAP}$	67.97%	67.87%	47.16%	45.87%	65.06%	63.23%	46.27%	44.92%
LambdaMART$_{NDCG}$	67.82%	66.79%	46.89%	46.46%	64.51%	63.38%	45.30%	44.58%
XE-MART$_{NDCG}$	65.70%	65.65%	46.19%	46.18%	63.71%	63.50%	45.80%	44.81%

https://doi.org/10.1371/journal.pone.0283574.t003

retrieval, DRMRR outperformed state-of-the-art LTR models and established its capability in differentiating relevant documents from irrelevant ones. In drug response prediction, our results indicated that DRMRR leads to substantially improved performance when compared to the competing methods across all performance metrics. Thus, DRMRR can infer robust predictors of drug responses from patient genomic or proteomic profiles which can lead to selecting a highly effective personalized treatment. In our robustness evaluations, we conducted a comprehensive analysis to assess the resilience of DRMRR against various types of noise and perturbations. Experimental results demonstrated that DRMRR is effective against: (*i*) Gaussian noise; (*ii*) universal adversarial perturbations by a substitute model with no knowledge of the victim model; (*iii*) black-box adversarial perturbations by a substitute model with access only to the deployed victim model; and (*iv*) probabilistic perturbation of relevance labels. Interestingly, the performance of DRMRR was consistently better than the baseline methods for all levels of noise. More importantly, DRMRR showed no significant change in its performance with the increase in the noise intensity. Two attributes of DRMRR did help to enhance its performance and robustness: (*i*) efficiently capturing the contextual information and interrelationship between documents/drugs via the GTD vector; and (*ii*) the distributional robustness by hedging against a family of plausible distributions, including the true distribution with high confidence.

Even though DRMRR demonstrated promising performance, it also suffers from some limitations that can be addressed in future work. DRMRR solves a convex problem which can be done very efficiently with 1st-order gradient methods. Its computational complexity is comparable to the training of leaf nodes in tree models (or the last layer of a neural network model), where a simple regression model is being trained. However, listwise ranking models can get relatively complex compared to pointwise or pairwise approaches and DRMRR is not an exception. One possible direction is to reformulate the problem to speed up the solutions to the DRO problem considered in this paper. As for the DRP application, an interesting future direction is to incorporate the toxicity of drugs in our predictions. Since the biological dissimilarities among patients affect the side effects of medications, patients may have various side effects. Hence, we can improve our predictions by considering the side effects and toxicity of drugs.

Supporting information

S1 Appendix. The supplementary materials file.
(PDF)

Author Contributions

Conceptualization: Shahabeddin Sotudian, Ruidi Chen, Ioannis Ch. Paschalidis.

Data curation: Shahabeddin Sotudian.

Formal analysis: Shahabeddin Sotudian, Ruidi Chen, Ioannis Ch. Paschalidis.

Funding acquisition: Ioannis Ch. Paschalidis.

Investigation: Shahabeddin Sotudian, Ruidi Chen, Ioannis Ch. Paschalidis.

Methodology: Shahabeddin Sotudian, Ruidi Chen, Ioannis Ch. Paschalidis.

Project administration: Ioannis Ch. Paschalidis.

Resources: Ioannis Ch. Paschalidis.

Software: Shahabeddin Sotudian.

Supervision: Ioannis Ch. Paschalidis.

Validation: Shahabeddin Sotudian.

Visualization: Shahabeddin Sotudian.

Writing – original draft: Shahabeddin Sotudian.

Writing – review & editing: Ruidi Chen, Ioannis Ch. Paschalidis.

References

1. Ru X, Ye X, Sakurai T, Zou Q. Application of learning to rank in bioinformatics tasks. Briefings in Bioinformatics. 2021;. https://doi.org/10.1093/bib/bbaa394 PMID: 33454758
2. Qin T, Liu TY, Xu J, Li H. LETOR: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval. 2010; 13(4):346–374. https://doi.org/10.1007/s10791-009-9123-y
3. Sotudian S, Desta IT, Hashemi N, Zarbafian S, Kozakov D, Vakili P, et al. Improved cluster ranking in protein–protein docking using a regression approach. Computational and structural biotechnology journal. 2021; 19:2269–2278. https://doi.org/10.1016/j.csbj.2021.04.028 PMID: 33995918
4. Bruch S. An alternative cross entropy loss for learning-to-rank. In: Proceedings of the Web Conference 2021; 2021. p. 118–126.
5. Papangelou K, Sechidis K, Weatherall J, Brown G. Toward an understanding of adversarial examples in clinical trials. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2018. p. 35–51.
6. Finlayson SG, Bowers JD, Ito J, Zittrain JL, Beam AL, Kohane IS. Adversarial attacks on medical machine learning. Science. 2019; 363(6433):1287–1289. https://doi.org/10.1126/science.aaw4399 PMID: 30898923
7. Qayyum A, Qadir J, Bilal M, Al-Fuqaha A. Secure and robust machine learning for healthcare: A survey. IEEE Reviews in Biomedical Engineering. 2020; 14:156–180. https://doi.org/10.1109/RBME.2020.3013489
8. Sotudian S, Paschalidis IC. Machine Learning for Pharmacogenomics and Personalized Medicine: A Ranking Model for Drug Sensitivity Prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2021;.
9. Liu TY, et al. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval. 2009; 3(3):225–331. https://doi.org/10.1561/1500000016
10. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, et al. Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning; 2005. p. 89–96.
11. Freund Y, Iyer R, Schapire RE, Singer Y. An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research. 2003; 4(Nov):933–969.
12. Cao Z, Qin T, Liu TY, Tsai MF, Li H. Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine Learning; 2007. p. 129–136.
13. Qin T, Liu TY, Li H. A general approximation framework for direct optimization of information retrieval measures. Information retrieval. 2010; 13(4):375–397. https://doi.org/10.1007/s10791-009-9124-x
14. Taylor M, Guiver J, Robertson S, Minka T. Softrank: optimizing non-smooth rank metrics. In: Proceedings of the 2008 International Conference on Web Search and Data Mining; 2008. p. 77–86.
15. Xia F, Liu TY, Wang J, Zhang W, Li H. Listwise approach to learning to rank: theory and algorithm. In: Proceedings of the 25th International Conference on Machine Learning; 2008. p. 1192–1199.
16. Wu Q, Burges CJ, Svore KM, Gao J. Adapting boosting for information retrieval measures. Information Retrieval. 2010; 13(3):254–270. https://doi.org/10.1007/s10791-009-9112-1
17. Biggio B, Corona I, Maiorca D, Nelson B, Šrndić N, Laskov P, et al. Evasion attacks against machine learning at test time. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2013. p. 387–402.
18. Chen R, Paschalidis IC. Distributionally robust learning. Foundations and Trends® in Optimization. 2020; 4(1-2).
19. Sinha A, Namkoong H, Volpi R, Duchi J. Certifying some distributional robustness with principled adversarial training. arXiv preprint arXiv:171010571. 2017;.

20. Wiesemann W, Kuhn D, Sim M. Distributionally robust convex optimization. Operations Research. 2014; 62(6):1358–1376. https://doi.org/10.1287/opre.2014.1314

21. Blanchet J, Murthy K. Quantifying distributional model risk via optimal transport. Mathematics of Operations Research. 2019; 44(2):565–600. https://doi.org/10.1287/moor.2018.0936

22. Shafieezadeh-Abadeh S, Kuhn D, Esfahani PM. Regularization via mass transportation. Journal of Machine Learning Research. 2019; 20(103):1–68.

23. Blanchet J, Glynn PW, Yan J, Zhou Z. Multivariate distributionally robust convex regression under absolute error loss. Advances in Neural Information Processing Systems. 2019; 32:11817–11826.

24. Chen R, Paschalidis IC. A robust learning approach for regression models based on distributionally robust optimization. Journal of Machine Learning Research. 2018; 19(13). PMID: 34421397

25. Shafieezadeh Abadeh S, Mohajerin Esfahani PM, Kuhn D. Distributionally robust logistic regression. Advances in Neural Information Processing Systems. 2015; 28.

26. Gao R, Chen X, Kleywegt AJ. Wasserstein distributional robustness and regularization in statistical learning. arXiv e-prints. 2017; p. arXiv–1712.

27. Mohajerin Esfahani P, Kuhn D. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming. 2018; 171 (1):115–166. https://doi.org/10.1007/s10107-017-1172-1

28. Ravikumar P, Tewari A, Yang E. On NDCG consistency of listwise ranking methods. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics; 2011. p. 618–626.

29. Wang Y, Wang L, Li Y, He D, Chen W, Liu TY. A theoretical analysis of NDCG ranking measures. In: Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013). vol. 8; 2013. p. 6.

30. Valizadegan H, Jin R, Zhang R, Mao J. Learning to Rank by Optimizing NDCG Measure. In: Advances in Neural Information Processing Systems. vol. 22; 2009. p. 1883–1891.

31. Hersh W, Buckley C, Leone T, Hickam D. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In: SIGIR'94. Springer; 1994. p. 192–201.

32. CCLE. Cancer Cell Line Encyclopedia (CCLE); 2021.

33. CTRP. Cancer Therapeutics Response Portal; 2021.

34. Metzler D, Croft WB. Linear feature-based models for information retrieval. Information Retrieval. 2007; 10(3):257–274. https://doi.org/10.1007/s10791-006-9019-z

35. Breiman L. Random forests. Machine learning. 2001; 45(1):5–32. https://doi.org/10.1023/A:1010933404324

36. Volkovs MN, Zemel RS. Boltzrank: learning to maximize expected ranking gain. In: Proceedings of the 26th Annual International Conference on Machine Learning; 2009. p. 1089–1096.

37. Lan Y, Zhu Y, Guo J, Niu S, Cheng X. Position-Aware ListMLE: A Sequential Learning Process for Ranking. In: UAI; 2014. p. 449–458.

38. Xu J, Li H. Adarank: a boosting algorithm for information retrieval. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2007. p. 391–398.

39. Xu J, Liu TY, Lu M, Li H, Ma WY. Directly optimizing evaluation measures in learning to rank. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2008. p. 107–114.

40. Metzler DA, Croft WB, McCallum A. Direct maximization of rank-based metrics for information retrieval. CIIR report. 2005; 429.

41. De Niz C, Rahman R, Zhao X, Pal R. Algorithms for Drug Sensitivity Prediction. Algorithms. 2016; 9 (4):77. https://doi.org/10.3390/a9040077

42. Riddick G, Song H, Ahn S, Walling J, Borges-Rivera D, Zhang W, et al. Predicting in vitro drug sensitivity using Random Forests. Bioinformatics. 2011; 27(2):220–224. https://doi.org/10.1093/bioinformatics/btq628 PMID: 21134890

43. Ma Y, Ding Z, Qian Y, Shi X, Castranova V, Harner EJ, et al. Predicting Cancer Drug Response by Proteomic Profiling. Clinical Cancer Research. 2006; 12(15):4583–4589. https://doi.org/10.1158/1078-0432.CCR-06-0290 PMID: 16899605

44. Wang X, Li C, Golbandi N, Bendersky M, Najork M. The lambdaloss framework for ranking metric optimization. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; 2018. p. 1313–1322.

45. Pobrotyn P, Bartczak T, Synowiec M, Białobrzeski R, Bojar J. Context-aware learning to rank with self-attention. arXiv preprint arXiv:200510084. 2020;.

46. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:14126572. 2014;.

47. Bhambri S, Muku S, Tulasi A, Buduru AB. A survey of black-box adversarial attacks on computer vision models. arXiv preprint arXiv:191201667. 2019;.

48. Niu S, Lan Y, Guo J, Wan S, Cheng X. Which noise affects algorithm robustness for learning to rank. Information Retrieval Journal. 2015; 18(3):215–245. https://doi.org/10.1007/s10791-015-9253-3