Regression Compatible Listwise Objectives for Calibrated Ranking with Binary Relevance

Aijun Bai
Google LLC
Mountain View, USA
aijunbai@google.com

Rolf Jagerman
Google LLC
Amsterdam, Netherlands
jagerman@google.com

Zhen Qin
Google LLC
Mountain View, USA
zhenqin@google.com

Le Yan
Google LLC
Mountain View, USA
lyyanle@google.com

Pratyush Kar
Google LLC
Paris, France
pratk@google.com

Bing-Rong Lin
Google LLC
Mountain View, USA
bingrong@google.com

Xuanhui Wang
Google LLC
Mountain View, USA
xuanhui@google.com

Michael Bendersky
Google LLC
Mountain View, USA
bemike@google.com

Marc Najork
Google LLC
Mountain View, USA
najork@google.com

ABSTRACT

As Learning-to-Rank (LTR) approaches primarily seek to improve ranking quality, their output scores are not scale-calibrated by design. This fundamentally limits LTR usage in score-sensitive applications. Though a simple multi-objective approach that combines a regression and a ranking objective can effectively learn scale-calibrated scores, we argue that the two objectives are not necessarily compatible, which makes the trade-off less ideal for either of them. In this paper, we propose a practical regression compatible ranking (RCR) approach that achieves a better trade-off, where the two ranking and regression components are proved to be mutually aligned. Although the same idea applies to ranking with both binary and graded relevance, we mainly focus on binary labels in this paper. We evaluate the proposed approach on several public LTR benchmarks and show that it consistently achieves either best or competitive result in terms of both regression and ranking metrics, and significantly improves the Pareto frontiers in the context of multi-objective optimization. Furthermore, we evaluated the proposed approach on YouTube Search and found that it not only improved the ranking quality of the production pCTR model, but also brought gains to the click prediction accuracy. The proposed approach has been successfully deployed in the YouTube production system.

CCS CONCEPTS

• **Information systems → Learning to rank**.

KEYWORDS

Learning to Rank; Calibration

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614712

ACM Reference Format:
Aijun Bai, Rolf Jagerman, Zhen Qin, Le Yan, Pratyush Kar, Bing-Rong Lin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2023. Regression Compatible Listwise Objectives for Calibrated Ranking with Binary Relevance. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3614712

1 INTRODUCTION

Learning-to-Rank (LTR) aims to construct a ranker from training data such that it can rank unseen objects correctly. It is therefore required that a ranker performs well on ranking metrics, such as *Normalized Discounted Cumulative Gain* (NDCG). It is usually the case that a ranking-centric pairwise or listwise approach, such as RankNet [3] or ListNet [29], achieves better ranking quality than a regression approach that adopts a pointwise formulation.

On the other hand, modern systems in these applications have multiple stages and downstream stages consume the predictions from previous ones. It is often desired that the ranking scores are well calibrated and the distribution remains stable. Take the application of online advertising as an example. In online advertising, the pCTR (predicted Click-Through Rate) model is required to be well calibrated because it affects the downstream auction and pricing models [6, 16, 30], though the final ranking of ads is the one that matters most for the performance. This suggests that we want the ranker to perform well not only on ranking metrics, but also on regression metrics in terms of calibrating ranker output scores to some external scale. Popular regression metrics include Mean Squared Error (MSE) for graded relevance labels and the logistic loss (LogLoss) for binary relevance labels.

Unsurprisingly, capable ranking approaches perform poorly on regression metrics, due to the fact that their loss functions are invariant to rank-preserving score transformations, and tend to learn scores that are not scale-calibrated to regression targets. Furthermore, these approaches suffer from training instability in the sense that the learned scores may diverge indefinitely in continuous

training or re-training [30]. These factors strongly limit their usage in score-sensitive applications. As a result, practitioners have no choice but to fall back on regression-only approaches even if they are suboptimal in terms of user-facing ranking metrics.

It has been shown that a standard multi-objective approach effectively learns scale-calibrated scores for ranking [16, 25, 30, 31]. However, we argue that in this standard multi-objective setting, the regression and ranking objectives are inherently conflicting, and thus the best trade-off might not be ideal for either of them. In this paper, we propose a practical regression compatible ranking (RCR) approach where the two ranking and regression components are proved to be mutually aligned. Although the same idea applies to ranking with both binary and graded relevance, we mainly focus on binary labels in this paper. Empirically, we conduct our experiments on several public LTR datasets, and show that the proposed approach achieves either the best or competitive result in terms of both regression and ranking metrics, and significantly improves the Pareto frontiers in the context of multi-objective optimization. Furthermore, we evaluated the proposed approach on YouTube Search and found it not only improved the ranking capability of the production pCTR model but also brought gains to the click prediction accuracy. The proposed approach has been fully deployed in the YouTube production system.

2 RELATED WORK

There is a long history of the study of Learning-to-Rank (LTR) [17]. The general set up is that a scoring function is trained to score and sort a list of objects given a context. The accuracy is evaluated based on ranking metrics that only care about the order of the objects, but not the scale of the scores. Existing works include designing more effective loss function [4, 10, 29, 33], learning from biased interaction data [9, 13, 27, 32], and different underlying models, from support vector machines [12], to gradient boosted decision trees [14, 28], to neural networks [1, 2, 19, 20, 23, 24]. However, almost none of the existing works have studied the calibration issue of the ranking model outputs, which limits their applicability in many applications where a calibrated output is necessary.

To the best of our knowledge, few works have studied the ranking calibration problem. Similar to classification problems, post-processing methods can be used for calibrating ranking model outputs. For example, Tagami et al. [26] used the pairwise squared hinge loss to train an LTR model for ads ranking, and then used Platt-scaling [21] to convert the ranking scores into probabilities. Recently, Chaudhuri et al. [6] compared different post-processing methods to calibrate the outputs of an ordinal regression model, including Platt-scaling and isotonic regression. Our proposed method does not rely on a post-processing step.

Another class of approaches is based on multi-objective setting where ranking loss is calibrated by a regression loss during the training time, without an additional post-processing step. Sculley [25] is an early work that combines regression and ranking. It has been used in concrete application [16, 30]. In particular, Yan et al. [30] used the multi-objective formulation in deep models to prevent models from diverging during training and achieve output calibration at the same time. The shortcoming of such an approach is that ranking accuracy can be traded for calibration because the two objectives are not designed to be compatible. Our proposed method does not sacrifice ranking accuracy to achieve calibration.

3 BACKGROUND

Learning-to-Rank (LTR) concerns the problem of learning a model to rank a list of objects given a context. Throughout the paper, we use *query* to represent the context and *documents* to represent the objects. In the so-called *score-and-sort* setting, a ranker is learned to score each document, and a ranked list is formed by sorting documents according to the scores.

More formally, let $q \in Q$ be a query and $x \in \mathcal{X}$ be a document, a score function is defined as $s(q, x; \boldsymbol{\theta}) : Q \times \mathcal{X} \to \mathbb{R}$, where Q is the query space, $\mathcal{X}$ is the document space, and $\boldsymbol{\theta}$ is the parameters of the score function s. A typical LTR dataset D consists of examples represented as tuples $(q, x, y) \in D$ where q, x and y are query, document and label respectively. Let $\mathbf{q} = \{q|(q, x, y) \in D\}$ be the query set induced by D. Let $\mathcal{L}_{query}(\boldsymbol{\theta}; q)$ be the loss function associated with a single query $q \in Q$. Depending on how $\mathcal{L}_{query}$ is defined, LTR techniques can be roughly divided into three categories: pointwise, pairwise and listwise.

In the pointwise approach, the query loss $\mathcal{L}_{query}$ is represented as sum of losses over documents sharing the same query. For example, in logistic-regression ranking (i.e. ranking with binary relevance labels), the Sigmoid Cross Entropy loss per document (denoted by SigmoidCE) is defined as:

$$\text{SigmoidCE}(s, y) = -y \log \sigma(s) - (1 - y) \log(1 - \sigma(s)), \quad (1)$$

where $s = s(q, x; \boldsymbol{\theta})$ is the predicted score of query-document pair (q, x) and $\sigma(s) = (1 + \exp(-s))^{-1}$ is the sigmoid function. SigmoidCE is shown to be scale-calibrated [30] in the sense that it achieves global minima when $\sigma(s) \to \mathbb{E}[y|q, x]$.

In the pairwise approach, the query loss $\mathcal{L}_{query}$ is represented as sum of losses over all document-document pairs sharing the same query. The fundamental RankNet approach uses a pairwise Logistic loss (denoted by PairwiseLogistic) [3]:

$$\text{PairwiseLogistic}(s_1, s_2, y_1, y_2) = -\mathbb{I}(y_2 > y_1) \log \sigma(s_2 - s_1), \quad (2)$$

where s_1 and s_2 are the predicted scores for documents x_1 and x_2, $\mathbb{I}$ is the indicator function, and σ is the sigmoid function. PairwiseLogistic achieves global minima when $\sigma(s_2 - s_1) \to \mathbb{E}[\mathbb{I}(y_2 > y_1)|q, x_1, x_2]$, which indicates that the loss function mainly considers the pairwise score differences, which is also known as the *translation-invariant* property [30].

In the listwise approach, the query loss $\mathcal{L}_{query}$ is attributed to the whole list of documents sharing the same query. The popular ListNet approach uses the Softmax based Cross Entropy loss (denoted by SoftmaxCE) to represent the listwise loss as [29]:

$$\text{SoftmaxCE}(s_{1:N}, y_{1:N}) = -\frac{1}{C} \sum_{i=1}^{N} y_i \log \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)}, \quad (3)$$

where N is the list size, s_i is the predicted score, and $C = \sum_{j=1}^{N} y_j$. The global minima will be achieved at [29]:

$$\frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)} \to \frac{\mathbb{E}[y_i|q, x_i]}{\sum_{j=1}^{N} \mathbb{E}[y_j|q, x_j]}. \quad (4)$$

Similar to PairwiseLogistic, the SoftmaxCE loss is translation-invariant, and could give scores that are arbitrarily worse with respect to regression metrics.

4 REGRESSION COMPATIBLE RANKING

In this section, we first give the motivation, then formally propose the approach to regression compatible ranking (RCR).

4.1 Motivation

It has been shown in the literature that a standard multi-objective approach effectively learns scale-calibrated scores for ranking [16, 25, 30]. Taking logistic-regression ranking as an example, Yan et al. define the multi-objective loss as a weighted sum of SigmoidCE and SoftmaxCE losses:

$$\mathcal{L}^{MultiObj}_{query}(\boldsymbol{\theta}; q) = (1-\alpha)\cdot\sum_{i=1}^{N}\text{SigmoidCE}(s_i, y_i) + \alpha\cdot\text{SoftmaxCE}(s_{1:N}, y_{1:N}), \quad (5)$$

where $\alpha \in [0,1]$ is the trade-off weight. For simplicity, we refer to this method as SigmoidCE + SoftmaxCE. It can be seen that SigmoidCE+SoftmaxCE is no longer translation-invariant, and has been shown effective for calibrated ranking. Let's take a deeper look at what scores are learned following this simple multi-objective formalization.

Given query q, let $P_i = \mathbb{E}[y_i|q, x_i]$ be the ground truth click probability further conditioned on document x_i. Recall that, SigmoidCE achieves global minima when $\sigma(s_i) \to P_i$, which means we have the following pointwise learning objective for SigmoidCE:

$$s_i \to \log P_i - \log(1-P_i). \quad (6)$$

On the other hand, SoftmaxCE achieves global minima when

$$\frac{\exp(s_i)}{\sum_{j=1}^{N}\exp(s_j)} \to \frac{P_i}{\sum_{j=1}^{N} P_j}, \quad (7)$$

or equivalently:

$$s_i \to \log P_i - \log\sum_{j=1}^{N} P_j + \log\sum_{j=1}^{N}\exp(s_j), \quad (8)$$

where the log-$\sum$-exp term is an unknown constant and has no effects on the value or gradients of the final SoftmaxCE loss.

In the context of stochastic gradient descent, Equations (6) and (8) indicate that the gradients generated from the SigmoidCE and SoftmaxCE components are *pushing the scores to significantly different targets*. This reveals the fact that the two losses in a standard multi-objective setting are inherently conflicting and will fail to find a solution ideal for both. How can we resolve this conflict?

Noticing that since $\sigma(s_i)$ is pointwisely approaching to P_i, if we replace the ground truth probabilities P_i on the right side of Equation (8) with the empirical approximations $\sigma(s_i)$ and drop the constant term, we are constructing some virtual logits:

$$s'_i \leftarrow \log\sigma(s_i) - \log\sum_{j=1}^{N}\sigma(s_j). \quad (9)$$

If we further apply SoftmaxCE loss on the new logits s'_i, we are establishing the following novel listwise learning objective:

$$\frac{\exp(s'_i)}{\sum_{j=1}^{N}\exp(s'_j)} \to \frac{P_i}{\sum_{j=1}^{N} P_j}, \quad (10)$$

which is equivalent to

$$\frac{\sigma(s_i)}{\sum_{j=1}^{N}\sigma(s_j)} \to \frac{P_i}{\sum_{j=1}^{N} P_j}. \quad (11)$$

It is easy to see that Equation (6) implies Equation (11) automatically, which means, as pointwise regression and listwise ranking objectives, they are well aligned in the sense that they achieve global minima simultaneously.

4.2 The Main Approach

Inspired by the above motivating example, we firstly define a novel Listwise Cross Entropy loss (ListCE) as follows.

Definition 1. *Let N be the list size, $s_{1:N}$ be the predicted scores, and $y_{1:N}$ be the labels. Let $T(s) : \mathbb{R} \to \mathbb{R}^+$ be a non-decreasing transformation on scores. The Listwise Cross Entropy loss with transformation T is defined as:*

$$\text{ListCE}(T, s_{1:N}, y_{1:N}) = -\frac{1}{C}\sum_{i=1}^{N} y_i \log\frac{T(s_i)}{\sum_{j=1}^{N} T(s_j)}, \quad (12)$$

where $C = \sum_{j=1}^{N} y_j$ is a normalizing factor.

For the scope of this paper, we interchangeably use ListCE with transformation T, ListCE(T), or even ListCE when there is no ambiguity. We immediately have the following propositions.

Proposition 1. ListCE(exp) *reduces to* SoftmaxCE.

Proposition 2. ListCE(T) *achieves global minima when*

$$\frac{T(s_i)}{\sum_{j=1}^{N} T(s_j)} \to \frac{\mathbb{E}[y_i|q, x_i]}{\sum_{j=1}^{N}\mathbb{E}[y_j|q, x_j]}. \quad (13)$$

Proof. Let $\overline{y} = \mathbb{E}[y|q, x]$ be the expected label of query-document pair (q, x). Applying the ListCE loss on $(x, y) \in D$ is then equivalent to applying it on $(x, \overline{y})$ in expectation. Given transformation T, and predicted scores $s_{1:N}$, with $p_i = T(s_i)/\sum_{j=1}^{N} T(s_j)$, we have:

$$\text{ListCE}(T, s_{1:N}, \overline{y}_{1:N}) = \frac{1}{\sum_{j=1}^{N}\overline{y}_j}\sum_{i=1}^{N}\overline{y}_i \log p_i, \quad (14)$$

subject to $\sum_{i=1}^{N} p_i = 1$.

Let's construct the following Lagrangian formalization:

$$\mathcal{L}(p_{1:N}, \lambda) = \frac{1}{\sum_{j=1}^{N}\overline{y}_j}\sum_{i=1}^{N}\overline{y}_i \log p_i + \lambda(\sum_{i=1}^{N} p_i 1). \quad (15)$$

Finding the extremum value of Equation (14) is then equivalent to finding the stationary points of Equation (15), which requires:

$$\frac{\partial\mathcal{L}(p_{1:N}, \lambda)}{\partial p_i} = \frac{\overline{y}_i}{p_i\sum_{j=1}^{N}\overline{y}_j} + \lambda = 0, \quad (16)$$

and

$$\frac{\partial\mathcal{L}(p_{1:N}, \lambda)}{\partial\lambda} = \sum_{i=1}^{N} p_i 1 = 0. \quad (17)$$

Note that Equations (16) and (17) give us a system of $N+1$ equations on $N+1$ unknowns. It is easy to see that the unique solution is

$$p_i = \frac{\overline{y}_i}{\sum_{j=1}^{N} \overline{y}_j}, \tag{18}$$

and $\lambda = 1$.

This indicates the unique global extremum at

$$\frac{T(s_i)}{\sum_{j=1}^{N} T(s_j)} \to \frac{\mathbb{E}[y_i|q, x_i]}{\sum_{j=1}^{N} \mathbb{E}[y_j|q, x_j]}. \tag{19}$$

It is easy to verify that this unique global extremum attributes to the global minima which concludes the proof. ∎

In logistic-regression ranking, all labels are binarized or within the range of [0, 1]. A natural pointwise objective is the SigmoidCE loss. With SigmoidCE as the pointwise component, it is then required to use the sigmoid function as the transformation such that they can be optimized simultaneously without conflict.

Definition 2. *The Regression Compatible Ranking (RCR) loss for a single query in a logistic-regression ranking task (i.e. ranking with binary relevance labels) is defined as:*

$$\mathcal{L}_{query}^{Compatible}(\boldsymbol{\theta}; q) = (1-\alpha) \cdot \sum_{i=1}^{N} \text{SigmoidCE}(s_i, y_i) + \alpha \cdot \text{ListCE}(\sigma, s_{1:N}, y_{1:N}), \tag{20}$$

where σ is the sigmoid function.

For simplicity, we refer to this method as SigmoidCE+ListCE(σ). We have the following proposition:

Proposition 3. SigmoidCE + ListCE(σ) *achieves global minima when* $\sigma(s_i) \to \mathbb{E}[y_i|q, x_i]$.

Proof. The SigmoidCE component achieves global minima when $\sigma(s_i) \to \mathbb{E}[y_i|q, x_i]$ which implies

$$\frac{\sigma(s_i)}{\sum_{j=1}^{N} \sigma(s_j)} \to \frac{\mathbb{E}[y_i|q, x_i]}{\sum_{j=1}^{N} \mathbb{E}[y_j|q, x_j]}, \tag{21}$$

which minimizes ListCE(σ) at its global minima. ∎

5 EXPERIMENTS ON PUBLIC DATASETS

To validate the proposed approach, we conduct our experiments on several public LTR datasets in this section.

5.1 Experiment Setup

5.1.1 Datasets. We compare our methods with baselines on three datasets: Web30K [22], Yahoo [5], and Istella [7]. These datasets have graded relevance labels. To study logistic-regression ranking, we simply binarize them by treating non-zero labels as 1s.

Web30K is a public dataset where the 31531 queries are split into training, validation, and test partitions with 18919, 6306, and 6306 queries respectively. There are on average about 119 candidate documents associated with each query. Each document is represented by 136 numerical features and graded with a 5-level relevance label. The percentages of documents with relevance label equal to 0, 1, 2, 3, and 4 are about 51.4%, 32.5%, 13.4%, 1.9%, and 0.8%. When being binarized, the percentages for 0 and 1 are 51.4% and 48.6%.

Yahoo LTR challenge dataset consists of 29921 queries, with 19944, 2994 and 6983 queries for training, validation, and test respectively. There are 700 numerical features extracted for each query-document pair. The average number of documents per query is 24, but some queries have more than 100 documents. The labels are numerically graded. The distribution over 0, 1, 2, 3, and 4 is 21.9%, 50.2%, 22.3%, 3.9%, and 1.7%. In binarized form, the distribution over 0 and 1 is 21.9% and 78.1%.

Istella LETOR dataset is composed of 33018 queries, with 20901, 2318, and 9799 queries respectively in training, validation, and test partitions. The candidate list to each query is with on average 316 documents, and each document is represented by 220 numerical features. The graded relevance labels also vary from 0 to 4 but with a more skewed distribution: 96.3% for 0s, 0.8% for 1s, 1.3% for 2s, 0.9% for 3s, and 0.7% for 4s. With binarization, this distribution becomes 96.3% for 0s and 3.7% for 1s.

5.1.2 Metrics. We are interested in both regression and ranking performance. For ranking performance, we adopt the popular NDCG@10 [11] as the main metric. More formally, given a list of labels $y_{1:N}$ and a list of output scores $s_{1:N}$, NDCG@k is defined as:

$$\text{NDCG@}k(s_{1:N}, y_{1:N}) = \frac{\text{DCG@}k(s_{1:N}, y_{1:N})}{\text{DCG@}k(y_{1:N}, y_{1:N})}, \tag{22}$$

where DCG@k is the so-called *Discounted Cumulative Gain* up to position k metric defined as:

$$\text{DCG@}k(s_{1:N}, y_{1:N}) = \sum_{i=1}^{N} \mathbb{I}(\pi(s_i) \le k) \frac{2^{y_i} - 1}{\log_2(\pi(s_i) + 1)}, \tag{23}$$

where $\pi(s_i)$ is the 1-based rank of score s_i in the descendingly sorted list of $s_{i:N}$.

For regression performance, we mainly look at the LogLoss metric, which is defined as:

$$\text{LogLoss}(\hat{y}_{1:N}, y_{1:N}) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i), \tag{24}$$

and where N is the total data size, y_i is the label and $\hat{y}_i$ is the predicted label. Note that for LogLoss, $\hat{y}_i = \sigma(s_i)$ is the predicted probability after sigmoid transformation.

In addition, we consider the Expected Calibration Error (ECE) [8, 18] as a universal metric for calibration. This metric is commonly used in uncertainty calibration. Following [30], we divide ranking documents in each query into M bins after we sort them by the model predictions, and compute the ECE by,

$$ECE = \frac{1}{|Q|} \sum_{q \in Q} \sum_{m=1}^{M} \frac{|B_m|}{|D_q|} \left| \frac{1}{|B_m|} \sum_{i \in B_m} y_i - \frac{1}{|B_m|} \sum_{i \in B_m} \hat{y}_i \right|. \tag{25}$$

In this work, we use $M = 10$ bins with each bin containing approximately the same number of documents with successive predictions.

5.1.3 Methods. We mainly compare the proposed SigmoidCE + ListCE(σ) method with SigmoidCE and SigmoidCE + SoftmaxCE. Additionally, we include ListCE(σ) in the comparison. We also compare with SoftmaxCE and SoftmaxCE-Platt, where the SoftmaxCE-Platt method applies the de facto Platt-scaling after a model that has been trained with SoftmaxCE.

Table 1: Comparisons on logistic-regression ranking tasks. Model selection is done on validation sets with test set results reported. Numbers with bold font indicate the best result. ▲ and ▼ indicate statistical significance with p-value=0.05 of better and worse results than the SoftmaxCE-Platt baseline.

Datasets	Web30K			Yahoo			Istella		
Metrics	NDCG@10	LogLoss	ECE	NDCG@10	LogLoss	ECE	NDCG@10	LogLoss	ECE
SigmoidCE	0.4626▲	**0.5996**▲	**0.1216**▲	0.6852▼	0.4296▲	0.1807▲	0.6560▼	**0.0612**▲	0.0275▼
ListCE(σ)	0.4528▼	1.1675▼	0.1657▼	0.6954▼	0.7652▼	0.1475▲	0.6862▲	0.0643▼	0.0248
SoftmaxCE	0.4578	23.7719▼	0.5503▼	0.6993	15.9964▼	0.2472▼	0.6839	60.5913▼	0.9556▼
SoftmaxCE-Platt	0.4578	0.6103	0.1333	0.6993	0.5036	0.1962	0.6839	0.0628	0.0246
SigmoidCE + SoftmaxCE	0.4665▲	0.6239▼	0.1509▼	0.7008▲	0.4626▲	0.1852▲	0.6861▲	0.0643▼	0.0271▼
SigmoidCE + ListCE(σ)	**0.4680**▲	0.6031▲	0.1275▲	**0.7050**▲	**0.4187**▲	**0.1550**▲	**0.6900**▲	0.0634	**0.0242**

We conduct our experiments using the TF-Ranking library [20]. In all experiments, we fix the ranker architecture to be a 3-layer Dense Neural Network (DNN) whose hidden layer dimensions are 1024, 512 and 256. The fraction of neuron units dropped out in training is set to be 0.5. We run the experiments on GPUs and use 128 as the training batch size. We use Adam [15] as the optimizer, perform an extensive grid search of learning rates (LRs) and α over $[0.01, 0.001] \times [0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 0.995, 0.999]$ for 100 epochs. For each method, model selection is done on validation sets with test sets results reported. For pointwise regression SigmoidCE, we select a model by its regression performance on LogLoss, as this is what they are optimized for, and we are interested to see what the best regression performance they can achieve; for listwise, Multi-Objective and RCR methods, we select a model by its ranking performance on NDCG@10.

To further study the behavior of the approaches in the context of multi-objective optimization, we evaluate all models on the test data, and plot the Pareto frontier for each model in comparison.

5.2 Experimental Results

5.2.1 Main Comparisons. The main results are shown in Table 1. From the results, we can make the following observations:

- The pointwise baselines consistently give either best or competitive regression performance in terms of both ECE and LogLoss; however, their ranking performance is usually inferior to other ranking-oriented methods, except in binarized Web30K where they archive better ranking performance than SoftmaxCE.
- The SoftmaxCE method can produce either best or competitive ranking performance; however, it is completely uncalibrated and performs poorly on regression metrics.
- The SoftmaxCE-Platt method performs well on regression metrics while giving the same ranking performance as SoftmaxCE. However, its regression and calibration performance is consistently inferior to the pointwise baselines.
- The standard multi-objective approach (SigmoidCE + SoftmaxCE) consistently achieves strong performance on both ranking and calibration metrics. It outperforms SoftmaxCE on all domains. This indicates that calibrated ranking scores can give better ranking performance than uncalibrated scores. In other words, the regression loss as a constraint in the multi-objective setting can help learning in ranking.
- The proposed RCR approach (SigmoidCE + ListCE(σ)) consistently achieves the best ranking performance, while having comparable or better regression metrics than the pointwise baselines. This indicates that the compatible ranking and regression components within the RCR approach may mutually benefit each other and can achieve the top result on both fronts. It is also noticed that it consistently outperforms SoftmaxCE-Platt on regression metrics.

These observations indicate the proposed RCR approach is stable and performs well in terms of both regression and ranking metrics on a variety of configurations.

5.2.2 Pareto Frontier Comparisons. In the context of multi-objective optimization, the Pareto frontier is the set of Pareto optimal solutions where there is no scope for further Pareto improvement which is defined as a new solution where at least one objective gains, and no objectives lose. For each method, we evaluate all models over the hyper parameter space on the test data, plot each result as a regression-ranking metrics data point, and draw the Pareto frontier.

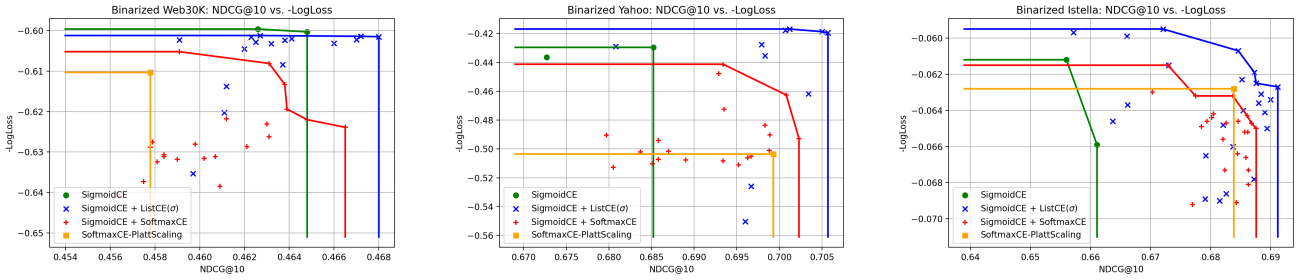
The results are shown in Figure 1. Note that we use -LogLoss in the figures, so the Pareto frontier corresponds to the maxima of a point set. From the figures, we can see that RCR consistently dominates other methods in all domains except binarized Web30k. In binarized We30k, RCR dominates all other methods except SigmoidCE which gives better regression performance; however, its ranking performance is inferior to RCR. These results suggest that RCR can improve the Pareto frontiers or give new Pareto optimal solutions (e.g. better trade-offs) in a wide range of tasks.

6 EXPERIMENTS ON YOUTUBE SEARCH

We verify our approach on the real-world YouTube Search system, through both offline evaluations and large-scale online A/B testing.

6.1 Background

In YouTube Search, real-time user interaction data, represented as item-click pairs, is streaming to the model training infrastructure in a continuous way. Our baseline is a pCTR model that is equipped with the traditional SigmoidCE loss. Recently, new data with search page information, represented as page-item-click tuples, is made available to training, which gives the opportunity to directly improve its ranking quality within a search page. However, as stated previously, a direct switch from pointwise pCTR model to listwise ranking model will not work in practice due to score calibration



(a) Binarized Web30K. **(b) Binarized Yahoo.** **(c) Binarized Istella.**

Figure 1: Pareto frontiers on the binarized Web30K, Yahoo and Istella datasets.

Table 2: Comparisons with respect to relative differences in YouTube Search with SigmoidCE as the baseline.

	AUCPR	LogLoss	NDCG@1	NDCG@5	NDCG@10
Multi-Objective: SigmoidCE + SoftmaxCE	-0.37%	+0.13%	**+0.30%**	**+0.16%**	**+0.15%**
RCR (proposed): SigmoidCE + ListCE(σ)	**+0.22%**	**+0.03%**	+0.27%	+0.13%	+0.13%

issues. It is required to improve the ranking quality of the model without affecting its original click metrics noticeably.

6.2 Experiments

In this context, we compared the standard multi-objective approach (SigmoidCE + SoftmaxCE), and our RCR approach (SigmoidCE + ListCE(σ)) following the same setting in our baseline and train them continuously over the past ~1 week of data over the same number of training steps for offline evaluation. The weight α is set to be 0.001 for both methods.

Offline Results. As in Table 2, we use LogLoss and AUCPR to measure regression accuracy and NDCG for ranking quality, where AUCPR is defined as the area under the Precision-Recall curve. Higher AUCPR or lower LogLoss indicate better regression accuracy. Note that for proprietary reasons, we only report relative numbers to our baseline (SigmoidCE). From the results, we can see that the standard multi-objective approach improved the pCTR model on the NDCG ranking metric, but it caused significant degradation in both AUCPR and LogLoss metrics. Such a degradation can significantly affect the downstream stages negatively, making the models not suitable for the system. The proposed RCR approach not only improved the ranking quality, but also brought gains to pCTR predictions in terms of AUCPR. This is because in our approach the ranking component optimizes for the ranking capability directly in a way that is compatible with the regression component and is acting as a valid and aligned in-list constraint for regression – which eventually helps the learning on regression. We also noticed that the proposed approach had a slight increase on LogLoss. This might be because the additional weight added on the ranking loss caused the learning on the regression loss to be less efficient than our baseline which is regression-only, thus the proposed approach may need more training steps for convergence.

Online A/B Testing. We further evaluated the model in a large-scale online A/B test over millions of users. The proposed model was tested against the production model. We report the following metrics in this experiment: **SearchCTR(%)** which is the percentage of clicks from search (the higher the better) and **SearchAbandonedRate(%)** which is the percentage of search queries that have 0 clicks and do not have refinements (the lower the better).

Due to data sensitivity, we only report relative performance of the experiment model over the production model. We observe that our proposed model improved SearchCTR by 0.66%, and reduced SearchAbandonedRate by 0.31% – which clearly indicates better search experiences for users. These metric gains are considered significant in the application. In comparison, the multi-objective approach doesn't qualify for production use because its negative impact on AUCPR and LogLoss. *The proposed model has now been fully deployed to the YouTube Search system.*

These results suggest that the proposed approach generalizes well to real-world production systems. The added ranking constraint not only improves ranking, but also benefits regression.

7 CONCLUSION

In this paper, we propose the practical regression compatible ranking (RCR) approach for ranking tasks with binary relevance. Theoretically, we show that the regression and ranking components are mutually aligned in the sense that they share the same solution at global minima. Empirically, we show that RCR performs well on both regression and ranking metrics on several public LTR datasets, and significantly improves the Pareto frontiers in the context of multi-objective optimization. Furthermore, we show that RCR successfully improves both regression and ranking performance of a production pCTR model in YouTube Search and delivers better search experiences for users. We expect RCR to bring new opportunities for harmonious regression and ranking and to be applicable in a wide range of real-world applications where there is a list structure. In future work, we are interested in exploring more formulations for regression compatible ranking and beyond.

REFERENCES

[1] Christopher Burges, Robert Ragno, and Quoc Le. 2007. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems*. MIT Press.

[2] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning*. 89–96.

[3] Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. *Learning* 11, 23-581 (2010), 81.

[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*. 129–136.

[5] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. *Proceedings of Machine Learning Research* 14 (2011), 1–24.

[6] Sougata Chaudhuri, Abraham Bagherjeiran, and James Liu. 2017. Ranking and Calibrating Click-Attributed Purchases in Performance Display Advertising. In *2017 AdKDD & TargetAd*. 7:1–7:6.

[7] Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Fast ranking with additive ensembles of oblivious and non-oblivious regression trees. *ACM Transactions on Information Systems (TOIS)* 35, 2 (2016), 1–31.

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[9] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 2019. To model or to intervene: A comparison of counterfactual and online learning to rank from user interactions. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 15–24.

[10] Rolf Jagerman, Zhen Qin, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2022. On Optimizing Top-K Metrics for Neural Ranking Models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2303–2307.

[11] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[12] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 133–142.

[13] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. 781–789.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. 3146–3154.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Cheng Li, Yue Lu, Qiaozhu Mei, Dong Wang, and Sandeep Pandey. 2015. Click-through Prediction for Advertising in Twitter Timeline. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1959–1968.

[17] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[18] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Vol. 29.

[19] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 499–508.

[20] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. 2019. TF-Ranking: Scalable tensorflow library for learning-to-rank. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2970–2978.

[21] John Platt. 2000. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans (Eds.). MIT Press, 61–74.

[22] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597* (2013).

[23] Zhen Qin, Zhongliang Li, Michael Bendersky, and Donald Metzler. 2020. Matching cross network for learning to rank in personal search. In *Proceedings of The Web Conference 2020*. 2835–2841.

[24] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?. In *Proceedings of the 9th International Conference on Learning Representations*.

[25] David Sculley. 2010. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 979–988.

[26] Yukihiro Tagami, Shingo Ono, Koji Yamamoto, Koji Tsukamoto, and Akira Tajima. 2013. CTR Prediction for Contextual Advertising: Learning-to-Rank Approach. In *Proceedings of the 7th International Workshop on Data Mining for Online Advertising*. Article 4, 8 pages.

[27] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 115–124.

[28] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1313–1322.

[29] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*. 1192–1199.

[30] Le Yan, Zhen Qin, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2022. Scale Calibration of Deep Ranking Models. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 4300–4309.

[31] Le Yan, Zhen Qin, Xuanhui Wang, Gil Shamir, and Mike Bendersky. 2023. Learning to Rank when Grades Matter. *arXiv preprint arXiv:2306.08650* (2023).

[32] Le Yan, Zhen Qin, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Revisiting Two-Tower Models for Unbiased Learning to Rank. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2410–2414.

[33] Xiaofeng Zhu and Diego Klabjan. 2020. Listwise learning to rank by exploring unique ratings. In *Proceedings of the 13th international conference on web search and data mining*. 798–806.