Multi-Label Learning to Rank through Multi-Objective Optimization

Debabrata Mahapatra
debabrata@u.nus.edu
National University of Singapore
Singapore
Yetian Chen
yetichen@amazon.com
Amazon
Seattle, WA, USA
Chaosheng Dong∗
chaosd@amazon.com
Amazon
Seattle, WA, USA
Michinari Momma∗
michi@amazon.com
Amazon
Seattle, WA, USA

ABSTRACT
Learning to Rank (LTR) technique is ubiquitous in Information Retrieval systems, especially in search ranking applications. The relevance labels used to train ranking models are often noisy measurements of human behavior, such as product ratings in product searches. This results in non-unique ground truth rankings and ambiguity. To address this, Multi-Label LTR (MLLTR) is used to train models using multiple relevance criteria, capturing conflicting but important goals, such as product quality and purchase likelihood for improved revenue in product searches. This research leverages Multi-Objective Optimization (MOO) in MLLTR and employs modern MOO algorithms to solve the problem. A general framework is proposed to combine label information to characterize trade-offs among goals, and allows for the use of gradient-based MOO algorithms. We test the proposed framework on four publicly available LTR datasets and one E-commerce dataset to show its efficacy.

CCS CONCEPTS
• Computing methodologies → Machine learning; Multi-task learning; • Information systems → Learning to rank.

KEYWORDS
Learning to Rank, Multi-Objective Optimization

1 INTRODUCTION
The field of Learning to Rank (LTR) has seen significant growth in recent years due to the availability of large amounts of labeled data for query-item relevance, either obtained through manual labeling or user behavior tracking. LTR aims to train a scoring function that assigns a relevance score to each retrieved item in order to rank them in the final results. Initially, LTR relied on a single criterion for relevance, but this uni-dimensional approach has been criticized for its limitations, such as subjectivity and noise in relevance articulation and an inability to account for multiple goals. To address these limitations, Multi-Label Learning to Rank (MLLTR) was introduced, which utilizes a multi-dimensional approach to relevance [35]. However, this multi-dimensional aspect also presents a major challenge: different relevance criteria can sometimes be in conflict. For instance, in web search, the goals of displaying items that the user is familiar with based on their view/click history and increasing the number of serendipitous items in the top results can be at odds. Given this conflict, it is often difficult to find a scoring function that optimizes for all relevance criteria simultaneously, requiring a compromise between them.

Multi-Objective Optimization (MOO) is a fascinating area that deals with the trade-offs between multiple objectives. One of the most important concepts in MOO is the Pareto Frontier (PF), which is a set of non-dominated solutions that represent the trade-offs between the objectives. The history of MOO research is rich, with numerous methods developed to specify the trade-off between
objectives. These methods include linear scalarization, weighted Chebyshev, and the \( \varepsilon \)-constraint method, among others. Despite these many methods, recent studies on MLLTR have mainly focused on approximating the PF, rather than finding a unique non-dominated solution that represents a particular trade-off. This is because the individual solutions on the PF may not necessarily correspond to a specific trade-off [6]. At best, they may only correspond to one type of trade-off [27]. Thus, there is a need for more research to develop methods that can accurately capture the trade-offs between objectives, leading to a better understanding of the PF and the trade-offs represented by its solutions.

Approximating the entire PF without considering trade-off specifications may seem attractive, but it is not practical for MLLTR. The final result presented to the user is merely a single ranked list produced to produce the final ranking; label aggregation, where the relevance labels are combined to create a single ranking model; and Linear Scalarization (LS), where a weight is assigned to each relevance criterion, collapsing the utility into a scalar function. The state-of-the-art Stochastic Label Aggregation (SLA) method [6] has been shown to be equivalent to LS. For recommendation applications, [18] proposed a framework for MOO-based MLLTR that guarantees finding non-dominated solutions, but does not consider trade-offs. In product search applications, [27, 26] proposed multiple relevance criteria and developed an \( \varepsilon \)-Constraint MOO algorithm that enables trade-off specification as upper bounds for all objectives except one.

Recently, various gradient-based MOO algorithms have been developed for Multi-Task Learning (MTL) applications [31, 17] to approximate the PF. [23] introduced an EPO algorithm that guarantees to find solutions that correspond to trade-off specifications defined by objective priorities. [25] developed the WC-MGDA algorithm that offers the same guarantees and can improve over arbitrary reference models. Meanwhile, [13] proposed the DBGD algorithm, an \( \varepsilon \)-Constraint type method that allows for trade-off specification as upper bounds of all objectives except one. In our MLLTR framework, we facilitate trade-off specification through various MOO methods, including classic methods such as LS and modern ones like EPO.

Another related research area in LTR also examines a multi-task learning approach. However, it uses only a single relevance label and incorporates auxiliary objectives to ensure that ranking results meet specific requirements, such as scale calibration [40], fairness [52, 28, 21], and diversity [16]. Conversely, our study focuses on employing sophisticated MOO algorithms to train ranking models using multiple labels, not just one relevance label.

2 BACKGROUND

2.1 Learning to Rank

Let \( Q \) be the set of all possible queries and \( D \) be the set of all documents or items. For a given query \( q \in Q \), let \( D^q = \{ d_i \}^{n_q}_{i=1} \subset D \) be the subset of \( n_q \) matched items. Let a query-item pair \((q, d_i)\) be represented by a \( p \)-dimensional feature vector \( x_i^q \in \mathbb{R}^p \). The goal of LTR is to learn a parametric scoring function \( f_\theta : \mathbb{R}^p \rightarrow \mathbb{R} \) that can assign a score \( s_i^q \) to each \( (q, d_i) \) pair from its corresponding vector representation, i.e., \( x_i^q \mapsto s_i^q \). The items can then be ranked in descending order of scores.

For a \((q, d_i)\) pair, we denote the relevance label as \( y_i^q \in \mathbb{Y} \). The training dataset for LTR consists of several queries: \( D_{\text{LTR}} = \{(x_i^q, y_i^q)\}_{i=1}^m \), where \( m \) is the number of queries and \( n_q \) is the number of data points in each query group.

For a query \( q \), let the output of a scoring function \( f_\theta \) for all the matched items in \( D^q \) be represented by a score vector \( s^q \in \mathbb{R}^{n_q} \). Similarly, let the corresponding relevance labels be denoted by the vector \( y^q \in \mathbb{R}^{n_q} \). The training cost is given by

\[
c(\theta) = \frac{1}{m} \sum_{q=1}^m \ell(s^q, y^q), \quad \text{where } s_i^q = f_\theta(x_i^q) \quad (1)
\]

for all \( i \in [n_q] = \{1, 2, \ldots, n_q\} \), and the per-query loss \( \ell(s^q, y^q) \) quantifies the extent to which the ordering of scores disagrees with that of the relevance labels.
In the pair-wise approach of LambdaMART cost [4], the event that one item $d_i$ is more relevant than another item $d_j$ w.r.t. $q$, denoted by $d_i \triangleright_d d_j$, is probabilistically modeled as $P(d_i \triangleright_d d_j) = \frac{\epsilon}{1 + e^{-\sigma(s_i - s_j)}}$, where $\sigma$ controls the spread of the Sigmoid function. The per-query loss $\ell$ in (1) is constructed from the log-likelihood $(\ell f)$ of $\theta$ given the (presumably independent) observations in the training data:

$$
\ell(s, y') = -\ell(f(\theta), D') = \sum_{(i, j) \in D'} |\text{ANDDCG}(i, j)| \cdot \log \left(1 + e^{-\sigma(s_i - s_j)}\right),
$$

(2)

where $D' = \{(x_i^q, y_i^q)\}_{i=1}^{n_q}$ is data pertaining to the matched items $D'_i$. $f_i^q = \{ (i, j) \in \{n_q\}^2 | y_i^q > y_j^q \}$ consists of item pairs having a strict relevance order, and $\text{ANDDCG}(i, j)$ is the change of the NDCG value when two items $i$ and $j$ swap their rank positions [4].

The scoring function is modeled by GBM [12] with $N$ decision trees: $f_\theta(x) = \sum_{t=1}^{N} \eta_t T_{\theta}(x)$, where $\eta_t$ is the learning rate, $T_\theta$ is the $t$th tree, and the full model parameter is $\theta = (\theta_t)_{t=0}^{N-1}$. On the $t$th iteration, the tree $T_{\theta}$ is learnt from the following training data:

$$
D_{\theta} = \{(x_i^q, \partial c/\partial x_i^q)\}_{i=1}^{n_q} \mid i=1, q=1,
$$

(3)

where the labels are gradients of cost w.r.t. the scores. In other words, instead of updating $f_\theta$ in the parameter space, it is updated in the function space of trees: $f_{\theta(t+1)} = f_{\theta(t)} - \eta_t T_{\theta(t)}$. The function space update of GBM suffices to treat the cost as a function of scores rather than the parameters $\theta$. Henceforth, we consider the cost $c : \mathbb{R}^M \rightarrow \mathbb{R}$ as a function of $s$, and rewrite (1) as

$$
c(s) = \frac{1}{m} \sum_{q=1}^{m} \ell(s, y').
$$

(4)

### 2.2 LTR from Multiple Relevance Labels

In MLLTR, different relevance criteria are measured, providing multiple labels for each query-item pair. The goal of MLLTR is still the same as that of LTR: to learn a scoring function $f_\theta$ that assigns a scalar value to each $(q, d_i)$ pair.

The labels for $(q, d_i)$ are $y_{i,k}^q \in \mathbb{Y}_k$ for $k = 1, \cdots, K$, where $K$ is the number of relevance criteria. Similar to LTR, each label set $\mathbb{Y}_k$ could be either discrete or continuous, endowed with a total ordering relation. The training dataset for MLLTR is denoted by

$$
D_{\text{MLLTR}} = \{(x_i^q, y_{i1}^q, \cdots, y_{iK}^q) \}_{i=1}^{n_q} \mid i=1, q=1,
$$

(5)

Every relevance criterion has a training cost. Therefore, in MLLTR, the cost is a vector valued function: $c(s) = [c_1(s), \cdots, c_K(s)]^T$, naturally making it an MOO problem.

### 2.3 Multi-Objective Optimization

In MOO, the cost function $c : \mathbb{R}^M \rightarrow \mathbb{R}^K$ is a mapping from the solution space $\mathbb{R}^M$ to the objective space $\mathbb{R}^K$.

We use $\mathbb{R}^K = \{c \in \mathbb{R}^K | c_k \geq 0, \forall k \in [K]\}$, the cone of positive orthant, to define a partial ordering relation. For any two points $c^1, c^2 \in \mathbb{R}^K$, we write $c^1 \succeq c^2$, if $c^1$ lies in the positive cone pivoted at $c^2$, i.e., $c^1 \in \{c^2 + e | e \in \mathbb{R}^K\}$. In other words, $c^1 \succeq c^2 \iff c^1 - c^2 \in \mathbb{R}^K$, making $c^1_k \geq c^2_k, \forall k \in [K]$. We define $c^1 > c^2$ when there is at least one $k$ for which $c^1_k > c^2_k$, i.e., $c^1 \not\succeq c^2$.

For minimization, a solution $s \in \mathbb{R}^M$ is said to be non-dominated or Pareto optimal, if there exists no other solution $s' \in \mathbb{R}^M$ such that $c(s) < c(s')$. We call the set of all non-dominated solutions the Pareto optimal set. The image of this Pareto set under the function $c$ is the Pareto Frontier (PF), which can be a $K-1$-dimensional manifold if connected [11, 22].

### 3 A FRAMEWORK FOR MULTI-LABEL LTR

#### 3.1 Linear Scalarization Based Methods

**Linear Scalarization (LS):** The MLLTR cost function gives rise to $K$ score-gradients, $\nabla c_k(s)$ for $k \in [K]$. However, for training the GBM based scoring function, the $i$th decision tree requires exactly one score-gradient as labels in its training data (3), not $K$ score-gradients. Although the cost is upgraded to become a vector valued function in MLLTR, the scoring function remains a scalar valued function. We combine the $K$ score-gradients as

$$
\lambda = \sum_{k=1}^{K} \alpha_k \nabla c_k(s), \quad \text{s.t.} \quad \sum_{k=1}^{K} \alpha_k = 1, \quad \alpha \in \mathbb{R}^K,
$$

(6)

where $\lambda \in \mathbb{R}^M$ are the labels for training the trees in GBM and $\alpha$ are combination coefficients.

#### 3.2 Preference Direction Based Methods

**Weighted Chebyshev (WC):** In WC, the vector valued cost is scalarized to

$$
\sigma_{WC}(s) = \max_{k \in [K]} r_k c_k(s).
$$

(7)

In general, the solution $s^* = \min_{s \in \mathbb{R}^M} \sigma_{WC}(s)$ satisfies $r_1 c_1(s^*) = r_2 c_2(s^*) = \cdots = r_K c_K(s^*)$ [24], which can be deduced by analyzing the level sets, illustrated in Figure 2b. This makes the trade-off specification between the objectives stricter than the penalty approach in the LS.

"Multi-Label Learning to Rank through MOO KDD '23, August 6–10, 2023, Long Beach, CA, USA"
Gradient Combination: Only the gradient of maximum relative objective value is chosen:

\[ a_k = \begin{cases} 
1, & \text{if } k = k^*, \\
0, & \text{otherwise,} 
\end{cases} \quad \text{s.t. } k^* = \arg \max_{k \in [K]} r_k c_k(s). \tag{10} \]

The objective vector value is proportional to the \( r^{-1} \)-ray as illustrated in Figure 2b. This trade-off specification guarantees that Pareto optimal points in the PF can be reached by varying the preferences, even when the objectives are non-convex. However, in practice, the strict trade-off requirement hinders the progress in cost value reduction. When optimizing with a step size (i.e., learning rate), the iterate \( \mathbf{c}' \) (cost at \( j \)th iteration) oscillates around \( r^{-1} \)-ray.

3.2.2 Exact Pareto Optimal Search (EPO): In EPO [23, 22], the trade-off specification is the same as that of WC. Therefore, most properties of WC are inherited. However, to overcome the limitations of WC, its gradient combination is designed to avoid oscillations around the \( r^{-1} \)-ray.

Gradient Combination: The coefficients are obtained by solving a quadratic program:

\[
\begin{align*}
\min_{\mathbf{a} \in \mathbb{R}^K} & \quad \| C^T \mathbf{c} - \mathbf{a} \|_2^2 \\
\text{s.t.} & \quad \sum_{k=1}^{K} a_k = 1,
\end{align*}
\tag{11}
\]

where \( C \in \mathbb{R}^{M \times K} \) is the matrix with \( K \) gradients in its column, and \( \mathbf{a} \) is an anchor direction in the objective space that determines the first order change in cost vector: \( \mathbf{c}^s + \mathbf{c}' \approx \delta \mathbf{c} = C^T \mathbf{a} \) from Taylor series expansion of \( \mathbf{c}(s') - C \mathbf{a} \). Here, \( \mathbf{a} \) is determined by

\[ a = \begin{cases} 
\mathbf{c}' - \frac{(\mathbf{c}' \cdot r^{-1})}{\| r^{-1} \|_2}, & \text{if } \mathbf{c}' \text{ is far}, \\
\mathbf{r}^{-1}, & \text{otherwise.} 
\end{cases} \tag{12} \]

When \( \mathbf{c}' \) is far (w.r.t. cosine distance) from \( r^{-1} \)-ray, the anchor is orthogonal to the \( r^{-1} \)-ray and directs towards it, as illustrated in Figure 2b. On the other hand, when \( \mathbf{c}' \) is near \( r^{-1} \)-ray, we move the cost along the \( r^{-1} \)-ray avoiding oscillations.

3.2.3 Weighted Chebyshev MGDA (WC-MGDA): In WC-MGDA algorithm [25], the trade-off specification is similar to that of WC method, but the SOCP formulations are designed to avoid the short-comings of WC, i.e., through the preferences over the objectives. WC-MGDA aims to build models that are closer or better than the reference model.

Gradient Combination: The coefficients are obtained by:

\[
\max_{\mathbf{a} \in \mathbb{R}^K, \mathbf{x} \in \mathbb{R}^a} \quad \mathbf{a}^T (r \odot (I(x) - \mathbf{b})) - u \gamma \\
\text{s.t.} & \quad \sum_{k=1}^{K} a_k = 1, \quad \| G_r \mathbf{a} \|_2 \leq \gamma, \tag{13} \]

where \( \mathbf{b} \) is the loss of the reference model, and \( G_r = \text{diag}(\sqrt{r}) \text{diag}(\sqrt{r}) \).

WC-MGDA jointly solves WC and MGDA to ensure achieving both preference alignment and Pareto Optimality. While the WC problem tries to find solutions by minimizing weighted \( \varepsilon_{\infty} \), the norm minimization ensures Pareto Optimality.

3.2.4 Evaluation Metric for Preference Direction Based MLLTR: To quantify the performance on preference based MLLTR, we use the objective function of WC (9), which exactly captures alignment with the \( r^{-1} \)-ray and is referred to as maximum weighted loss (MWL). Figure 3 illustrates a prototypical case with 3 models. In terms of MWL, \( M_1 \) and \( M_3 \) are the same, although \( M_1 \) dominates \( M_3 \), and better than \( M_2 \). Between \( M_1 \) and \( M_3 \), we use the volume of intersection between the negative orthant area (VNO) pivoted by each model and \( \mathbb{R}^K \) (color shaded area in Figure 3) as a tiebreaker. Note, VNO should always be used as a tie breaker when the difference in MWL is insignificant in our paper. For example, MWL are 1 for both \( M_1 \) and \( M_3 \). VNO for \( M_3 \) is 1 while VNO for \( M_2 \) is \( 1 \times 0.3 = 0.3 \). Thus, \( M_3 \) is better than \( M_1 \) due to the VNO of \( M_3 \) is smaller even if MWL are the same for both models.
Suitable When ratio between objectives is given as preference direction Type of Trade-off specification constraints (upper bounds) for secondary criteria

- LS/SLA total utility / cost is known
- WC/EPO preference direction as weights on linear combination of costs
- WC/MGD a reference model is known (pretrained model), preference direction from the reference model
- EC/AL/EC-DBGD hard constraints on objectives are known

∈\[k\] where \(\phi\) where \(k\) is a control function associated with constraint \(c_k\) for objectives, where MA is applied on its gradient. It is not applicable from the momentum based first order methods [30] for single objective, to multi-objective setup. For the upper bound \(K - 1\) costs are restricted to satisfy an upper bounded constraint given by the \(\phi_k\).

Gradient Combination: [27] proposed an augmented Lagrangian form of (15) as

\[
\max_{\alpha} \sum_{k \in [K]} \alpha_k (e_k - \epsilon_k),
\]

where \(\epsilon_k\) is a smoothing factor. Note that this is different from the momentum based first order methods [30] for single objective, where MA is applied on its gradient. It is not applicable to multi-objective case, because the search direction can change significantly between consecutive iterations due to change in the \(\alpha\), even when the objective gradients do not change significantly. Therefore, we smoothen the coefficients rather than the gradients.

We illustrate three types of trade-off specifications in Figure 2, summarize the training of scoring function in algorithm 1, the MOO methods in algorithm 2, and when to use which method in table 1.

### 3.4 Non-Smooth Trajectory and Remedy by Moving Average (MA)

All the MOO methods discussed previously are first order methods, where the final search direction is formulated by adaptively combining the objective gradients. However, the step size is kept fixed (or heuristically decreased) in every iteration, instead of adapting it to the ever changing search direction. This causes the iterates to exhibit oscillatory behavior in their cost functions, as empirically verified in section 4.2 and 4.3. Note, although theoretically step size selection techniques such as Line Search methods [37] can adapt the step size, they cannot be used in practice due to high computational cost: the objective function needs to be computed several times in every iteration. Moreover, it is non-trivial to extend these methods, primarily developed for single objective, to multi-objective setup.

Not all MOO method exhibit non-smooth trajectory though. In LS, it does not happen because the search direction do not change drastically as it is the gradient of a fixed objective function. In EC-AL, there is no change in the coefficients of primary cost, and change for secondary costs are smoothed by the coefficients of previous iteration.

To mitigate this issue of non-smooth trajectory MOO methods, we propose a simple yet effective technique. We apply a moving average (MA) filter to \(\alpha\) between consecutive iterations:

\[
\alpha^t \leftarrow \nu \alpha^t + (1 - \nu) \alpha^{t-1},
\]

where \(\nu \in (0,1)\) is a smoothing factor. Note that this is different from the momentum based first order methods [30] for single objective, where MA is applied on its gradient. It is not applicable to multi-objective case, because the search direction can change significantly between consecutive iterations due to change in the \(\alpha\), even when the objective gradients do not change significantly. Therefore, we smoothen the coefficients rather than the gradients.
4 EXPERIMENTS

We evaluate the effectiveness of our MLLTR framework by examining the compliance with the trade-off specification and accuracy in approximating the PF. The performance improvement provided by the MA method, as described in Section 3.4, is also assessed for each MOO method. Lastly, we demonstrate how extended preference-based methods can be used to explore the PF around a reference objective vector, thereby updating production models.

4.1 Datasets and Experimental Settings

We test our MLLTR framework using five datasets: the Microsoft LETOR dataset (MSLR-WEB30K) [29], Yahoo! LETOR dataset [7], Istella LETOR dataset [10], Istella-S [20], and a proprietary E-commerce dataset that has similarities to the datasets used in [33, 27], but was collected in 2021. Details of these datasets can be found in Table 2.

For all datasets, we use some features as additional labels in addition to the original relevance label and remove them from the feature list to prevent data leakage. The extra labels for MSLR-WEB30K are Query-URL Click Count (Click), URL Dwell Time (Dwell), Quality Score (QS), and Quality Score2 (QS2). Unlike the MSLR dataset, feature descriptions for Yahoo! and two Istella datasets are not publicly available. Thus, we selected extra labels for them through a descriptive analysis of the features (for details, see Appendix A). The labels for the E-commerce dataset include a binary target indicating whether an item was purchased or not, historical purchases, relevance score between the query and product, brandedness scores of the product, and delivery speed of the product.

4.1.1 Trade-off Specification. For LS and preference-based methods, we set \( r \) as follows. For bi-objective cases, we generated 5 \( r^{-1} \) rays that are equally distributed in the region between the two ‘baseline cost’ vectors. A baseline cost vector was obtained by training a model for only one objective and computing costs for all objectives. For the tri-objective case, we generated 25 preference directions using PESA [34, 22], which samples equi-distributed points on the convex hull of baseline cost vectors. For the \( \varepsilon \)-Constraint methods, we set \( \varepsilon \) of the secondary objectives as follows. In a bi-objective case, the baseline cost vector corresponding to the primary objective has a sub-optimal cost value for the non-primary objective. We set 5 upper bounds by dividing this sub-optimal cost into 5 levels. Similarly, in a tri-objective case, the two sub-optimal cost values in the baseline cost vector were divided into 5 levels each, resulting in 25 pairs of upper bounds for the two non-primary objectives.

4.1.2 Hyperparameter Tuning. For each dataset, we fine-tuned hyperparameters (i.e., number of trees and learning rate) of the GBM model by conducting LTR using main relevance judgments as the single label. We selected the best configuration of hyperparameters (according to the NDCG@5), which was 600 trees and a learning rate of 0.25, after evaluating the grid of hyperparameters \( \{300, 600, 900, 1200\} \times \{0.05, 0.15, 0.25, 0.35\} \) for number of trees and learning rates, respectively.

4.2 Qualitative Evaluation

As a preliminary experiment, we applied existing methods in their original form to the (Click, Rel) pair on the MSLR dataset, including linear weighting methods (LS, SLA), preference-based methods (WC, EPO, WC-MGDA), and EC methods (EC-AL, EC-DBGD).

4.2.1 Compliance with Trade-off Specification. To visualize the compliance, we compute the cost vectors on training dataset. To assess the ranking performance, we compute the NDCG@5 metric on validation data. Figure 4 shows the result.

In Figure 4a, we observe that the final cost vectors of LS and SLA are very similar for each preference specification, which is expected as they are known to be probabilistically equivalent [6]. However, the solutions generated by LS slightly dominate those of SLA in the cost space, which translates to better performance in the NDCG space. The solutions produced by preference-based methods (WC, EPO, and WC-MGDA) have a closer alignment with the preference rays than LS and SLA. This is particularly evident for extreme preference rays near the baselines, where the blue and purple square points of LS are farther from the corresponding preference rays compared to WC, EPO, and WC-MGDA. This indicates that if the trade-off is specified not as a utility but as a ratio between objectives, LS should not be used. Nevertheless, if one considers the frontier of solutions, LS dominates every other method in the non-extreme regions of the PF. Similarly, in Figure 4b, for the \( \varepsilon \)-Constraint methods, the solutions of both EC-AL and EC-DBGD generally comply with the constraint specifications. However, the frontier of EC-AL is superior to that of EC-DBGD in most cases.

4.2.2 Issue of Non-Smooth Trajectory. Keeping the aspect of trade-off compliance aside, a surprising observation is the solutions from the simpler baselines such as LS and EC-AL seems to dominate the other methods. To understand this, we plot cost curve for several models in Figure 5. LS is the only method that has smooth behavior in the figure. SLA is a stochastic version of LA and non-smooth changes are visible, which causes inferior dominance. For WC, the oscillation is expected, as it chooses only one label that have maximum weighted cost. Although EPO does achieve lower values of the cost as compared to WC, it still has oscillations. We observed the same issues exists in WC-MGDA and EC-DBGD, across all multi-label experiments and datasets.
Figure 4: Initial results of bi-objective experiments on MSLR-WEB30K [29] dataset. Colored lines and points represent different trade-off specifications and the corresponding solutions, respectively.

Figure 5: Cost curves for (a) SLA/LS, (b) WC and (c) EPO for (Click, Rel). For WC and EPO, we also show $\alpha$ for Rel. We use light color for the original methods and dark for smoothed versions.

Figure 6: Improved results with moving average. Most of the models are close to the Pareto Front.

4.2.3 Improvement with Moving Average. We have consistently used a smoothing factor of $\nu = 0.1$, as specified in (19). The smoothed cost curves are displayed in Figure 5 using dark colors for the WC and EPO methods. The improved results in the cost/NDCG space with smoothed $\alpha$ can be seen in Figure 6, presenting a noticeable improvement when compared to the results in Figure 4. The smoothing has effectively prevented the solutions of the preference based methods from being dominated by the LS method, while at the same time aligning them with the specified preference rays. Similarly, the solution frontier of EC-DBGD is now not dominated by that of EC-AL.
We quantify the improvement of employing MA over the vanilla MOO method on two metrics: MWL (defined in section 3.2.4) for preference-based methods, and Hypervolume Indicator (HVI) [1] for all methods. The MWL quantification (lower is better) combines two aspects of an MOO method: 1) the cost vector’s alignment with preference-based methods, and 2) the overall performance of the utility vector.
Table 7: Metrics on Istella LETOR dataset for bi-objective and tri-objective experiments.

(a) Istella LETOR dataset (2-obj)  
| Metric          | Method          | MWL (test) | HVI (train cost) | HVI (test NDCG) |
|-----------------|-----------------|------------|-----------------|----------------|
|                 | orig | ma | gain (%) | orig | ma | gain (%) | orig | ma | gain (%) |
| SLA/LS          | 6.26 | 4.79 | -23.6% | 3.90 | 3.93 | 0.6% | 0.86 | 0.90 | 4.4% |
| WC              | 61.05 | 3.93 | -93.6% | 3.59 | 3.90 | 8.7% | 0.80 | 0.90 | 11.5% |
| EPO             | 31.72 | 4.31 | -86.4% | 3.87 | 3.92 | 1.2% | 0.88 | 0.90 | 2.9% |
| WC-MGDA         | 25.67 | 3.82 | -85.1% | 3.86 | 3.91 | 1.3% | 0.86 | 0.90 | 5.3% |
| EC method       | WC-MA           | - - | - | 3.84 | - | - | 0.83 | - | - |
|                 | WC-MGDA-MA      | - - | - | 3.86 | 3.89 | 0.9% | 0.84 | 0.85 | 2.1% |

(b) Istella LETOR dataset (3-obj)  
| Metric          | Method          | MWL (test) | HVI (train cost) | HVI (test NDCG) |
|-----------------|-----------------|------------|-----------------|----------------|
|                 | orig | ma | gain (%) | orig | ma | gain (%) | orig | ma | gain (%) |
| SLA/LS          | 2.79 | 2.36 | -15.5% | 7.91 | 7.93 | 0.2% | 0.96 | 0.73 | 7.8% |
| WC              | 36.64 | 2.05 | -94.4% | 7.79 | 7.92 | 1.7% | 0.58 | 0.73 | 25.7% |
| EPO             | 2.90 | 2.18 | -24.8% | 7.87 | 7.91 | 0.5% | 0.73 | 0.77 | 5.6% |
| WC-MGDA         | 3.76 | 2.03 | -46.1% | 7.91 | 7.92 | 0.1% | 0.68 | 0.74 | 8.1% |
| EC method       | EC-AL           | - - | - | 3.88 | - | - | 0.66 | - | - |
|                 | EC-DBGD         | - - | - | 7.89 | 7.90 | 0.0% | 0.68 | 0.69 | 0.5% |

Table 8: Metrics on preference with reference points.

| Metric      | E-commerce | SLA/LS | EPO | WC-MGDA-MA |
|-------------|-------------|--------|-----|-------------|
| Metric      | MWL          | HVI(ndcg) | HVI(ndcg) | HVI(ndcg) |
| orig | 1.0 | 8.0e-4 | 1.3e-2 | 3.9e-4 |
| ma   | 1.0 | 1.3e-2 | 3.9e-4 | 3.9e-4 |
| gain (%) | -37 | 50 | 16 | -19 | 21 | 14 |

Figure 7: Exploring PF from a reference model (black dot) on E-commerce dataset.

5 CONCLUSION AND FUTURE WORK

We present a comprehensive framework for Multi-Label Learning to Rank that integrates any first-order gradient-based MOO algorithm to train a ranking model. Our framework incorporates three distinct trade-off specifications and implements a systematic approach to preserve the relative ranking quality with regards to various relevance criteria. Through a thorough evaluation of multiple state-of-the-art MOO algorithms, we demonstrate the efficacy of our framework by testing it on four publicly available datasets and one E-commerce dataset.

Our framework for MLLTR can be enhanced in several ways as further research. Firstly, the current pairwise cost can be extended to list-wise cost to improve performance. Secondly, one can investigate the use of non-convex surrogates, which have been shown to approximate the NDCG metric more effectively than list-wise costs (as per [3]). Thirdly, more MOO algorithms can be integrated to further refine and optimize our framework. Moreover, we will also study the online and offline performance for our proposed algorithms in production search engine [39]. We will make the source code public available.
A EXPERIMENTS

A.1 Datasets and Experimental Settings

A.1.1 Yahoo! Learning to Rank Dataset. We experimented on Yahoo! Learning to Rank [7] challenge dataset with 36K queries. Each query-url pair is represented by 700 features. Although these features are engineered (not learnt), their descriptions, however, are not publicly released. Therefore, we selected several labels to use as additional objectives. Specifically, we selected features that have more than 5 levels of values, then chose the ones that were least correlated among each other. The selected labels are ['0', '18', '22', '39', '92']. Details can be seen in Figure 9. In total, we selected 5 objectives including the original relevance label, and created 15 bi-objectives and 10 tri-objective cases. The generation of preference and constraints follows the same strategy explained in MSLR-WEB30K. Note as we saw cost vanishing behavior coming from '203', '214'. Details can be seen in Figure 10. In total, we selected 5 objectives including the original relevance label, and created 15 bi-objectives and 10 tri-objective cases. The generation of preference and constraints follows the same strategy explained in MSLR-WEB30K. For tuning the model hyperparameters, we followed a similar strategy as in MSLR-WEB30K, and selected 600 trees and 0.25 learning rate. We used the original training and test data for our experiment.

A.1.2 Istella-S and Istella LETOR Datasets. We also ran experiments on Istella-S LETOR dataset [20] and Istella LETOR dataset [10], where the former is a smaller sample of the later dataset. Each query-url pair is represented by 200 features. Although these features are engineered (not learnt), their descriptions, however, are not publicly released. Therefore, we selected several labels to use as additional objectives. Specifically, we selected features that have more than 5 levels of values, then chose the ones that were least correlated among each other. The selected labels are ['0', '11', '194', '203', '214']. Details can be seen in Figure 10. In total, we selected 5 objectives including the original relevance label, and created 4 bi-objectives cases. The generation of preference and constraints follows the same strategy explained in MSLR-WEB30K. For tuning the model hyperparameters, we followed a similar strategy as in MSLR-WEB30K, and selected 1000 trees and 0.05 learning rate. We used the original training and test data for our experiment.

Experiment results: Figure 8 shows the results of one run for Istella LETOR dataset. We plot the preference based methods and \(\epsilon\)−Constraint method separately to avoid overcrowding the figures. Moreover, both their respective trade-off specifications are different, further justifying our decision of separate figures.

Among the preference based methods, first we observe that the performance of LS and SLA are similar, from both the aspects of dominance and preference compliance w.r.t. the other MOO methods. This is expected as they are probabilistically equivalent [6]: expected cost of SLA is same as LS. However, in practice, the solutions of LS slightly dominate that of SLA in the cost space, which translates to a significant dominance in the NDCG space.

Next, we observe a similar relation among WC-MGDA, EPO and CS. The solutions of WC-MGDA either, in most cases, dominate or stay non-dominated w.r.t. to the solutions of EPO Search and CS. The dominance is apparent in the cost space. Solutions from all methods have stricter compliance to the preferences than that of LS and SLA.
Figure 10: Objectives selection for Istella LETOR dataset. We plot the histogram of the standard deviation of the feature per query group. We then select the features that have the most diversified values per query group.

For $\epsilon$-Constraint method, an equidistributed trade-off specification (the upper bounds) in the cost space does not lead to an equidistributed solution frontier in the NDCG space. However, this discrepancy is problem dependent, and should be attributed to the loose approximation of RankNet or LambdaRank \cite{4} cost to NDCG metric, but not to the $\epsilon$-Constraint method. The MOO method complies to the given trade-off specifications in the cost space.