Learning to Rank for Uplift Modeling

Floris Devriendt$, Jente Van Belle$, Tias Guns, and Wouter Verbeke$

Abstract—Causal classification concerns the estimation of the net effect of a treatment on an outcome of interest at the instance level, i.e., of the individual treatment effect (ITE). For binary treatment and outcome variables, causal classification models produce ITE estimates that essentially allow one to rank instances from a large positive effect to a large negative effect. Often, as in uplift modeling (UM), one is merely interested in this ranking, rather than in the ITE estimates themselves. In this regard, we investigate the potential of learning to rank (L2R) techniques to learn a ranking of the instances directly. We propose a unified formalization of different binary causal classification performance measures from the UM literature and explore how these can be integrated into the L2R framework. Additionally, we introduce a new metric for UM with L2R called the promoted cumulative gain (PCG). We employ the L2R technique LambdaMART to optimize the ranking according to PCG and show improved results over the use of standard L2R metrics and equal to improved results when compared with state-of-the-art UM. Finally, we show how L2R techniques can be used to specifically optimize for the top-\(k\) fraction of the ranking in a UM context, however, these results do not generalize to the test set.

Index Terms—Learning to rank, uplift modeling, causal classification, performance measures

1 INTRODUCTION

Causa| classification models estimate for each instance the causal effect of a treatment on an outcome variable of interest, i.e., the individual treatment effect (ITE) [1]. This causal inference task is encountered in the literature under various names, e.g., heterogeneous treatment effect estimation [2], individualized treatment rule learning [3], conditional average treatment effect estimation [4], and uplift modeling [5]. Causal classification models have been applied in various domains to maximize the effectiveness of e.g., personalized medicine [6] and marketing campaigns [7]. Another application is determining which customers to target with a retention campaign to maximize reduction in churn while also minimizing the use of resources [8].

In this work, we focus on binary causal classification in that we consider both treatment and outcome to be binary variables. In this setting, ITE estimates allow one to rank instances from a large positive effect to a large negative effect. Often, one is merely interested in this ranking rather than in the ITE estimates themselves. This specific ranking objective is typically encountered in the uplift modeling (UM) literature [5]. Learning to rank (L2R) techniques, which stem from the information retrieval community, comprise techniques specifically designed to optimize the quality of predicted rankings directly, rather than the quality of predicted values that serve to rank instances [9]. This work investigates whether L2R techniques can be successfully used in the context of UM.

While existing UM techniques are in fact already approaches to L2R since standard classification techniques can be considered as ‘pointwise’ approaches to L2R (see Section 3), we present different ways to formulate UM as an L2R problem. As this is directly linked to the different evaluation measures proposed to assess the quality of produced rankings in UM, we first provide an overview of these measures. We then consider how suited existing L2R measures are for UM and introduce a new L2R metric for this purpose, called the promoted cumulative gain, by translating the uplift metric Area Under the Uplift Curve to an L2R measure which can be directly used together with the LambdaMART L2R technique.

Finally, in both UM and information retrieval, often only the top-\(k\) fraction of the ranking is of interest to the user [5]. However, while L2R comprises techniques and measures designed to optimize for the top-\(k\) specifically, current UM techniques aim to optimize the entire ranking. Therefore, we will investigate whether specifically optimizing for the top-\(k\) by using L2R techniques can be successfully applied to UM. To the best of our knowledge, optimizing for the top-\(k\) has not been investigated before in UM.

Our contributions are:

- We investigate and experimentally compare the main UM evaluation measures in use today, and propose a unified formalization in which all measures can be unambiguously written.
- We explore the different ways in which UM can be formulated as an L2R problem.
- We introduce a new metric for UM with L2R, called the promoted cumulative gain.
- We empirically evaluate the different L2R formulations for UM on multiple datasets.
- We investigate optimizing specifically for the top-\(k\) in UM by using L2R techniques, though the benefit of top-\(k\) learning is shown to be limited as the results do not generalize to the test set.
- We compare different state-of-the-art UM techniques with our best performing L2R formulation and show

---

$ Floris Devriendt, Jente Van Belle, and Tias Guns are with the Data Analytics Laboratory, Solvay Business School, Vrije Universiteit Brussel, 1050 Brussels, Belgium. E-mail: {floris.devriendt, Jente.Van.Belle, tias.guns}@vub.be.

$ Wouter Verbeke is with the Faculty of Economics and Business, Katholieke Universiteit Leuven, 3000 Leuven, Belgium. E-mail: wouter.verbeke@kuleuven.be.

Manuscript received 20 April 2020; revised 14 October 2020; accepted 16 December 2020. Date of publication 31 December 2020; date of current version 12 September 2022.

(Corresponding author: Wouter Verbeke.)

Recommended for acceptance by C. Li Ph.D.

Digital Object Identifier no. 10.1109/TKDE.2020.3048510
that L2R is a viable alternative to the existing UM methodology.

The rest of the paper is organized as follows: in Section 2 we formally introduce causal classification and UM, discuss related work and elaborate on the evaluation measures used in the UM literature. Section 3 covers L2R and its measures, how UM can be formulated as an L2R problem and introduces a new metric for the L2R framework. In Section 4, we describe the experiments and report the results. Section 5 discusses the results and finally Section 6 presents our conclusions.

2 CAUSAL CLASSIFICATION & UPLIFT MODELING

Causal classification is about estimating the class of an instance in function of the treatment that is applied. Hence, it is equivalent to estimating the ITE, i.e., the causal effect of a treatment on an outcome of interest at the instance level [1]. Therefore, causal classification models can be used to optimize treatment assignment to instances with the aim to maximize (or, depending on the application at hand, to minimize) the outcome of interest. In this work, we focus on binary causal classification in that we consider both treatment and outcome to be binary variables.

Suppose that we have a dataset \( \mathcal{D} = \{ (x, y, t) \} \) of \( n \) instances with each a feature vector \( x \in \mathcal{X} \), response variable \( y \in \{ 0, 1 \} \) and treatment indicator \( t \in \{ 0, 1 \} \). Following the Neyman-Rubin potential outcomes framework [10], the ITE can be formally defined as:

\[
\tau(x) = E[Y(1) - Y(0) | x],
\]

(1)

where \( Y(1) \) and \( Y(0) \) are the two potential outcomes and correspond to the response \( y \) of an instance belonging to the treatment \( (t = 1) \) and control \( (t = 0) \) group, respectively. Note that it is generally assumed that some instances respond \( (y = 1) \) without being treated \( (t = 0) \) (e.g., natural healing or subscribing to a product independent of advertising). The main difficulty in ITE estimation is that \( \tau(x) \) is not directly observable since we can only observe one of the potential outcomes for a particular instance. That is, we can observe the outcome after treating or not treating, but cannot know what the outcome would have been for the opposite treatment choice. This is commonly known as The Fundamental Problem Of Causal Inference [11].

In this work, we focus on estimating the ITE in randomized controlled trial (RCT) settings. In such setting, the treatment was administered randomly and independent of \( x \) and \( \tau(x) \) can be estimated as [4], [12]

\[
\hat{\tau}(x) = P(y = 1|x, t = 1) - P(y = 1|x, t = 0).
\]

(2)

The ITE is thus defined as the difference between the probability of an instance to respond if treated minus the probability of an instance to respond if not treated.

As mentioned in the introduction, in case of binary treatment and outcome variables, ITE estimates allow one to rank instances from a large positive effect to a large negative effect. In many cases, one is merely interested in this ranking rather than in the ITE estimates themselves. This specific setting, which is the focus of this work, lies at the core of the UM literature [5].

UM has as goal to rank an unseen set of instances, e.g., customers, by their estimated \( \hat{\tau}(x) \) and to target a highly ranked fraction of this set of instances. For example, in marketing or churn prediction, the size of the fraction is determined by the campaign budget, e.g., the top 1,000 or 10,000 customers. Given a limited budget, these customers are expected to be most likely to respond when targeted with the campaign.

2.1 Related Work

UM techniques can be grouped into data preprocessing and data processing approaches. In the data preprocessing approaches, after pre- or postprocessing data and outcomes, existing out-of-the-box learning methods are used. In the data processing approaches, new learning methods and methodologies are developed that aim to estimate \( \tau(x) \) more directly. For an in depth discussion on UM techniques we refer to [5], [13].

A popular and generic data preprocessing approach is the flipped label approach, also called the class transformation approach [14], [15]. In this approach, a new target variable \( z \in \{ 0, 1 \} \) is created where \( z = 1 \) if either: \( t = 1 \) and \( y = 1 \) or \( t = 0 \) and \( y = 0 \); and \( z = 0 \) otherwise. Due to this class transformation, estimation of \( \tau(x) \) is converted into a binary classification problem with label \( z \), allowing us to adopt any standard classification technique [15].

Other data preprocessing approaches extend the set of features to allow for the estimation of \( \tau(x) \). An example is grouping together the instances from both treatment and control group and including a dummy variable that denotes to which group an instance belongs. A model is then developed from: (1) the original features; (2) the added dummy variable; and (3) interaction variables between the features and the dummy variable [16], [17]. This model can then be used to estimate two response probabilities for a new instance, once as if the instance belongs to the treatment group and once as if it belongs to the control group. Subtracting the probabilities returns \( \hat{\tau}(x) \).

Data processing approaches comprise both indirect and direct estimation approaches. Indirect estimation approaches include the two model approach. This approach builds two separate models to predict the response probabilities, one for the treatment group and one for the control group. For a new instance, the probability of responding is estimated with each model. Afterwards, the probabilities are subtracted to obtain \( \hat{\tau}(x) \).

Direct estimation approaches are typically adaptations of decision tree algorithms such as CART [18] or CHAID [19]. Proposed adaptations include modified splitting criteria and dedicated pruning techniques. Examples of tree-based techniques include significance-based uplift trees [20], decision trees with information theory-inspired splitting criteria [21] and uplift random forest and causal conditional inference trees [22]. However, there is also a group of direct estimation techniques that builds on support vector machines [23], [24], [25].

Finally, note that the overview above focuses on techniques to estimate ITE in RCT settings. However, ITE estimation based on observational data with treatment selection bias is also an active research area in the field of causal machine learning. For recent works on this subject, one may refer to [1], [26] and references therein.

Our work differs from all the above in that we focus on the link between UM and L2R, both when treatment and
2.2 Performance Measures for Evaluating Uplift Models

Because $\pi(x)$ is unobservable, we can not directly measure the quality of the estimated $\hat{\pi}(x)$ values. However, since in UM one is rather interested in the ranking that results from ranking instances by $\hat{\pi}(x)$, the norm is to evaluate the quality of this ranking. [20] propose to do this by computing and plotting the cumulative incremental gains for an incrementally larger subgroup of the ranked population, i.e., the lift in response rate as a result of treating more instances or the cumulative incremental treatment effect.

Fig. 1 shows an example, where the blue line represents the cumulative incremental gains as a function of the selected fraction of the ranked population. The black line represents the expected cumulative incremental gains for a random subsample of that size, called the random baseline. A good uplift model ranks instances likely to respond when treated higher, leading to higher estimated cumulative incremental gains in the early parts of the plot.

Interestingly, many different ways for computing and visualizing the cumulative incremental gains have been proposed in the literature. Two variants exist with a different name: the Qini Curve and the Uplift Curve. In this work, we focus on the difference of how the cumulative incremental gain values are computed for both variants. However, note that also other differences between the curves occur in the literature, e.g., what values are plotted, and how a single measure such as Area Under the Curve is derived.

Even for the Qini and Uplift variants, there are no unique definitions. We analyzed the literature, and identified two main differences. The first one relates to whether the ranking is computed for each group (i.e., treatment and control group) separately or for one joint group. If the ranking is computed separately, the top 10 percent instances are the top 10 percent instances of the treatment group and the top 10 percent instances of the control group. However, if the ranking is computed jointly, the top 10 percent instances originate from both groups. Hence, in the joint setup, the proportions of instances of each group represented in the ranking can differ from the global proportions of treatment and control groups. The second difference is linked to the potential imbalance of the treatment and control groups as it relates to whether the cumulative incremental gains are computed in (rebalanced) absolute numbers of instances or in relative terms.

An overview of the different definitions occurring in the literature, structured according to the identified differences discussed above, is provided in Table 2. Before discussing differences between the definitions in more detail, we first introduce notation allowing us to formalize the different definitions.

For evaluation, we assume the presence of a predictive model $\hat{u}$ that allows us to rank a dataset $D$ in decreasing order, i.e., instances are ranked from high to low $\hat{\pi}(x)$ (see Section 2.1). Note that $\hat{u}$ does not necessarily estimates $\pi(x)$ based on the probabilities as in Equation (2) directly. We denote the total number of treated ($t = 1$) and control ($t = 0$) instances among the top-$k$ ranked instances by $N_T(D, k)$ and $N_C(D, k)$, and the number of treated and control responders ($y = 1$) among the top-$k$ ranked instances by $R_T(D, k)$ and $R_C(D, k)$, respectively.

In the literature, the lift in response rate is often computed and compared for both the treatment and control group separately. To formalize this, we introduce the subsets $T$ and $C$, with $T$ the subset of treated instances and $C$ the subset of instances that belong to the control group. In line with the notations introduced in the previous paragraph for rankings covering both the treatment and control group, we denote the number of treated and control responders among the top-$k$ ranked instances per group by $R(T, k)$ and $R(C, k)$.

To clarify the notation introduced, consider Table 1 which shows a minimal example of a ranked dataset. For the number of treated instances among the top 1 and top 3 instances in the dataset, one would write $N_T(D, 1) = 1$ and $N_T(D, 3) = 2$, respectively. Likewise, to obtain the number of treated responders among the top 1 and top 3 instances in the dataset, one would write $R_T(D, 1) = 0$ and $R_T(D, 3) = 1$, respectively. The above notations relate to the joint scenario, in which a ranking is computed for the treatment and control instances as one group. In the separate scenario, on the other hand, both the treatment and control group have their own ranking. Therefore, to obtain the number of treated responders among the top 1 and top 3 instances, one would write $R(T, 1) = 0$ and $R(T, 3) = 2$, respectively. Notice the difference in the numbers obtained for the top 3 case.

Table 2 shows the different evaluation measures for UM proposed in the literature. For the separate scenarios, we use $p$ to denote a percentage, where $p|T|$ and $p|C|$ are then

![Fig. 1. A cumulative incremental gains curve (blue) and the expected gains from random treatments (black).](image-url)
the corresponding absolute numbers of instances being considered from the treatment and control groups, respectively. The value function $V()$ returns the cumulative incremental gains value for the first $p$-percent of the population, or for all instances up to and including instance $k$.

The first curve that occurs in the literature is the Qini Curve [28]. This curve plots the absolute number of incremental responses of the treated group compared to as when there is no treatment. To obtain a balanced comparison, however, the number of responders among the top $p$-percent of the control group is adjusted to neutralize the effect of different treatment and control group sizes. The values for the Qini Curve are then obtained by

$$V(p) = R(T, p|T|) - R(C, p|C|).$$

The Qini Curve has since been used and modified by several researchers to evaluate the performance of uplift models.

An alternative to the Qini Curve is the Uplift Curve. The Uplift Curve is obtained by subtracting two separate lift curves, one for the treatment and one for the control group, using the same model [21]. In [30] the authors measure the cumulative incremental gains by subtracting the gain obtained from the first $p$-percent of ranked instances of the control group from the gain obtained from the first $p$-percent of ranked instances of the treatment group. A normalization factor, i.e., dividing by the respective group sizes, is added to account for overall imbalance in treatment and control groups. The corresponding formula is then

$$V(p) = \frac{R(T, p|T|)}{|T|} - \frac{R(C, p|C|)}{|C|}.$$  \hspace{1cm} (4)

Note that from a modeling point of view, the above equation is equivalent to Equation (3) as it can be obtained by dividing Equation (3) by the constant $|T|$. Hence, both measures will give rise to the same conclusions when used for comparing different models.

In [24], the authors follow the same reasoning that was used to obtain Equation (4), however, the difference is that they measure the cumulative incremental gains in absolute terms. The authors do not mention any normalization applied in calculating the cumulative incremental gains

$$V(p) = R(T, p|T|) - R(C, p|C|).$$  \hspace{1cm} (5)

All above definitions of Qini and Uplift curves evaluate uplift models in a separate manner, i.e., the top $p$-percent of the treatment group is compared with the top $p$-percent of the control group. A different approach is to consider both the treatment and control groups as one joint group and evaluate targeting the top-$k$ from that group, which is closer to how uplift models are to be used in practice. In the joint relative setup, there is no clear distinction between the Qini and Uplift curves, as they are both obtained as follows [12], [36]:

$$V(k) = \frac{R^T(D, k)}{|T|} - \frac{R^C(D, k)}{|C|}.$$  \hspace{1cm} (6)

For the joint absolute setting, however, both Qini and Uplift variants can be distinguished. In the first variant, the cumulative incremental gains are expressed in rebalanced absolute numbers [34]

$$V(k) = \frac{R^T(D, k) - R^C(D, k)}{N^T(D, k)}.$$  \hspace{1cm} (7)

Note that this measure rebalances the responder counts based on the numbers of treated and control instances among the top-$k$ ranked instances, instead of the overall imbalance $|T|/|C|$. The second variant is proposed by [35] and uses a different approach to obtain the cumulative incremental gains in rebalanced absolute numbers

$$V(k) = \left( \frac{R^T(D, k)}{N^T(D, k)} - \frac{R^C(D, k)}{N^C(D, k)} \right) \times (N^T(D, k) + N^C(D, k)).$$  \hspace{1cm} (8)

Here, the responder counts are divided by their respective number of instances among the top-$k$ ranked instances before subtraction, of which the result is then multiplied by the total number of instances considered in the ranking.
In addition to constructing a Qini or Uplift Curve according to one of the definitions above, we can also calculate the Area Under the Qini or Uplift Curve, hereinafter referred to as AUUC, to obtain a single numerical quantity to measure and compare the performance among uplift models. Sometimes the quantity obtained is reduced by the AUUC of the baseline [12], [28], however, in this paper we ignore this constant.

For any of the joint setting definitions from Table 2, the AUUC is defined as

\[
AUUC = \int_0^1 V(x) \, dx = \sum_{k=1}^n V(k). \tag{9}
\]

For the separate setting definitions, we make the following approximation over 100 intervals:

\[
AUUC = \int_0^1 V(x) \, dx \approx \sum_{p'=1}^{100} V\left(p = \frac{p'}{100}\right). \tag{10}
\]

Note that in the literature, it is common to use only 10 groups (or deciles), though this does not provide much granularity.

The different evaluation measures are empirically compared in Section 4.2. However, as our goal is not only to evaluate rankings but also to optimize rankings, we turn to L2R in the next section.

3 LEARNING TO RANK (L2R)

L2R finds its origin in the Information Retrieval (IR) domain. IR is defined as finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [37]. Ranking is a core problem in IR, as it is an important part of many IR problems (e.g., document retrieval, collaborative filtering and product rating) [9]. In what follows, we use document retrieval as example to illustrate how L2R works and to establish the connection between L2R and UM.

A search engine is the most common example of a document retrieval system. The web consists of an extremely large amount of documents (i.e., webpages) and finding relevant documents is a difficult task. A search engine has multiple components, but one of its most crucial ones is the ranker. The ranker is responsible for matching the request of the user (i.e., the query) with relevant indexed documents. The goal of a ranking algorithm is to produce a ranked list of documents according to its relevance to a given query [9].

L2R algorithms can essentially be categorized into three groups: pointwise, pairwise and listwise approaches [9]. The pointwise approach predicts the relevance of each document and uses these final scores to rank all documents considered. Standard classification techniques can be considered as pointwise approaches to L2R as they can be used to discriminate between ‘relevant’ and ‘not relevant’ documents. One limitation of a pointwise approach is that the interdependency between documents is not taken into consideration, meaning that the loss function used does not consider the documents’ final place in the ranking. The pairwise approach takes as input pairs of documents and outputs for each pair which of the two documents is preferred in terms of relevance by relying on a classification model. For a entire list of documents, one thus obtains as output relative orderings for pairs of documents. However, deriving the position of all documents in the final ranking from this output is a difficult problem. Finally, the listwise approach takes as input the entire list of documents and directly outputs a ranking of all documents. This approach is closest to the L2R ideology as there is no mismatch between the learning stage and the final output of the algorithm (as is the case for pointwise and pairwise approaches).

One of the most well-known and versatile techniques in L2R is LambdaMART [38], [39]. LambdaMART combines two methods previously proposed in the field, namely LambdaRank and Multiple Additive Regression Trees (MART). A core idea of LambdaRank, which is itself an extension of RankNet, is that it can directly optimize a ranking measure, even if it is non-differentiable, by relying on so-called \( \lambda \)-gradients. These gradients are determined heuristically by multiplying the gradients of a pairwise loss function by the difference obtained in the ranking metric under consideration due to swapping the pair’s positions in the ranking. Combining LambdaRank gradients with the learning algorithm Multiple Additive Regression Trees, which is a gradient boosted tree algorithm, gives us LambdaMART. For more information, one may refer to [38], [39].

Whilst a pairwise loss function is used in obtaining \( \lambda \)-gradients, the second factor depends on the global structure of the entire list of documents. Therefore, LambdaMART can be considered a listwise approach. Next to research on non-parametric methods such as LambdaMART, there is also research on parametric (model-based) listwise L2R methods, such as ListNet [40] and Plackett-Luce ranking models [41]. However, as the general objective of this work is to explore the potential of L2R for UM, in what follows we focus on adopting the LambdaMART L2R technique for UM. A broader exploration of L2R methods for UM is identified as a prime topic for further research.

3.1 Performance Measures for Evaluating L2R Models

Commonly used performance metrics in L2R are [42], [43], [44]: Precision (P), Mean Average Precision (MAP), Cumulative Gain (CG) and (Normalized) Discounted Cumulative Gain (DCG). Different metrics are used depending on whether relevance values are binary or graded, e.g., graded according to a five-star rating system. Below, we briefly discuss each of the above measures.

3.1.1 Binary Relevance

When working with binary relevance values, an instance \( i \) (i.e., a document in document retrieval) is either ‘relevant’ or ‘not relevant’, i.e., \( rel_i \in \{0, 1\} \).

The Precision at \( k \) (\( P(k) \)) corresponds to the proportion of relevant instances (\( rel_i = 1 \)) among the top-\( k \) ranked instances

\[
P(k) = \frac{\sum_{i=1}^{k} rel_i}{k}. \tag{11}
\]

The Average Precision of a query \( q \), consisting of \( |q| \) instances, sums over all instances in the query and computes the average of the \( P(k) \) values at every \( k \) where a relevant instance is positioned
\[
V(p) = R(T, p) - R(C, p); \\
V(p) = \frac{R(T, p)}{R(C, p)}; \\
V(p) = \frac{R(T, p) - R(C, p)}{R(C, p)}
\]

Relevance Values for Separate Queries for Each Value Function Definition

| Value Function | Treatment Responder \( rel_{TR} \) | Treatment Non-Responder \( rel_{TNR} \) | Control Responder \( rel_{CR} \) | Control Non-Responder \( rel_{CNR} \) |
|----------------|----------------------------------|-----------------------------------|---------------------------------|-----------------------------------|
| \( V(p) = R(T, p) - R(C, p) \) | 1                                | 0                                | -1                              | 0                                |
| \( V(p) = \frac{R(T, p)}{R(C, p)} \) | 1                                | 0                                | -\frac{1}{2}                    | 0                                |
| \( V(p) = \frac{R(T, p) - R(C, p)}{R(C, p)} \) | 0                                | 0                                | -\frac{1}{2}                    | 0                                |

Query 1

\[
\text{AvgP}(q) = \frac{\sum_{i=1}^{n} P(i)}{\sum_{i=1}^{n} rel_i}.
\]

The Mean Average Precision (MAP) is the mean of Average Precisions over a set of \( Q \) queries.

\[
\text{MAP}(Q) = \frac{\sum_{q \in Q} \text{AvgP}(q)}{|Q|}.
\]

3.1.2 Graded Relevance

Here, the relevance is assumed to be given as a graded score, e.g., \( rel_i \in \{1, 2, \ldots, 5\} \), where a higher value means higher relevancy. Evaluation metrics can be modified accordingly, as highly relevant instances are more valuable than marginally relevant instances, which in turn are more valuable than not relevant instances.

The Cumulative Gain of a ranked list is the sum of all relevance values of the ranked list of a single query \( q \), up to point \( k \).

\[
CG(k) = \sum_{i=1}^{k} rel_i.
\]

A limitation of CG is that it does not consider the connection between ranking position and instance relevancy [42].

An alternative that takes this connection into account, i.e., that considers whether the instances with highest relevancy do appear at the top of the ranking, is the Discounted Cumulative Gain. To achieve this, graded relevance values are discounted logarithmically proportional to the ranking positions [42].

\[
DCG(k) = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i + 1)}.
\]

An alternative formulation for DCG does exist, which places more emphasis on the relevance values by using \( 2^{rel_i} - 1 \) instead of \( rel_i \) in the numerator [45]. When relevance values are binary, both formulations are equal [46].

Different queries can relate to different numbers of instances. To fairly compare a rankers’ performance among multiple queries of different sizes, one can normalize the DCG of each query to obtain scores in the range of [0,1]. The Normalized Discounted Cumulative Gain (NDCG) metric normalizes DCG by dividing the achieved DCG of each query by its Ideal Discounted Cumulative Gain (IDCG), where the IDCG is obtained by sorting all instances according to their relevance values resulting in the maximum possible DCG [42].

\[
IDCG(k) = \frac{\sum_{i=1}^{k} \text{rel}'_i}{\log_2(k + 1)}.
\]

\[
NDCG(k) = \frac{\text{DCG}(k)}{\text{IDCG}(k)},
\]

where \( \text{rel}'_i \) is the best possible relevance value for instance \( i \) as it results from the best possible ranking.

By averaging over a set of queries \( Q \), one can obtain the mean DCG and NDCG (similar to Equation (13)).

3.2 Uplift Modeling as Learning to Rank

To cast UM as an L2R problem, we first determine appropriate relevance values for each of the UM performance measures presented in Table 2. These relevance values allow us to use existing L2R performance measures in combination with an L2R technique such as LambdaMART to learn a ranking. Next, as an alternative to using existing L2R measures, we propose a new metric for UM with L2R, called the promoted cumulative gain (PCG), which can be used to learn a ranking that directly optimizes the AUUC.

3.2.1 Relevance Values for UM With L2R

In UM, an instance belongs to either the treatment group \((t = 1)\) or the control group \((t = 0)\). As shown in Table 2, for evaluating uplift models, we can rely on separate and joint performance measures, depending on whether a ranking is computed for each group separately or for one joint group. Likewise in L2R, one or multiple queries can be considered. To be more specific, for L2R in the separate UM setting, we consider both the treatment and control groups as separate queries, i.e., an L2R technique will be run to optimize the rankings of the two queries separately. In the joint UM setting, only one query is considered, covering instances of both the treatment and control groups.

We now determine relevance values for each of the UM performance measures from Table 2. The relevance value of an instance takes over the role of \( \tau(x) \) which is used in traditional UM. Recall that an instance can belong to one of the following four categories: Treatment Responder (TR), Treatment Non-Responder (TNR), Control Responder (CR) and Control Non-Responder (CNR). To obtain relevance values for each UM performance measure, we check the effect of each of the above categories on their value functions.

Table 3 shows the different value functions and the corresponding relevance values for the separate queries. The effect of a TR or TNR is assessed by looking at the left components of the subtractions in the value functions. A TR increases the overall values obtained, whereas a TNR does...
not impact the value functions at all. The size of the effect of a TR, however, depends on the value function considered. Similarly, the right components are used to assess the effects of a CR and CNR. An increase in the right components lowers the overall values obtained. Therefore, a negative relevance is assigned to a CR, however, the relevance value changes depending on the value function considered. A CNR, on the other hand, does not affect any of the value functions considered.

Table 4 shows the different value functions and the corresponding relevance values for the joint queries. For the relative value function, the relevance values are identical to those for the relative value function for the separate queries. However, for the absolute value functions, TNRs and CNRs no longer have neutral effects on the overall values as the responder counts are rebalanced using the number of treated and control instances among the top-k ranked instances. Consider the Joint Absolute Qini Curve for which the responder count of the control group is rebalanced by multiplication of \( \frac{(N^T(D, k))/(N^C(D, k))}{1} \). A TNR thus increases this ratio, leading to an increase in the right component of the value function, and hence a lower overall value. A CNR, on the other hand, results in a decreased value for the right component of the value function and hence an increase in the overall value. Quantifying the exact increases and decreases in the overall values, however, is not trivial. Instead, we observe that a TR increases the overall value the most, while a CNR only causes a small decrease and a CR causes a larger decrease in the overall value. We simply encode this relation by using relevance values 3, 2, 1 and 0 for a TR, CNR, TNR and CR, respectively. For the Joint Absolute Uplift Curve, a similar analysis gives rise to the same insights (and therefore we do not include it in Table 4).

The various sets of relevance values presented can all be used in combination with any L2R metric that accepts graded relevance values.

### 3.2.2 Transforming AUUC into an L2R Metric

The goal of this section is to come up with an L2R measure that is most similar to the AUUC, and that can be readily optimized using existing L2R techniques. To this purpose, we use the relative definitions of the Uplift Curve as the experiment in Section 4.2 shows that these are most robust to differences in group sizes.

Equation (9) shows that the AUUC is a summation of the value function over all possible \( k \) values. If we insert the value function definition for the Joint Relative Uplift Curve (Equation (6)), we obtain

\[
\text{AUUC} = \sum_{k=1}^{n} V(k) = \sum_{k=1}^{n} \left( \frac{R^T(D, k)}{|T|} - \frac{R^C(D, k)}{|C|} \right).
\]

Let us introduce the following helper function \( g(i) \):

\[
g(i) = \begin{cases} 
0 & \text{if } y = 0 \\
1/|T| & \text{if } y = 1 \text{ and } t = 1 \\
-1/|C| & \text{if } y = 1 \text{ and } t = 0
\end{cases}
\]

where the three cases are mutually exclusive and cover all possible assignment combinations for \( y \) and \( t \).

Recall from Section 2.2 that \( R^T(D, k) \) and \( R^C(D, k) \) denote the number of treated (\( t = 1 \)) and control (\( t = 0 \)) responders \( (y = 1) \) among the top-\( k \) ranked instances. As these quantities are obtained by a summation over the top-\( k \) instances, Equation (18) can be reformulated in terms of \( g(i) \), where \( i \) represents a single instance, as follows:

\[
\text{AUUC} = \sum_{k=1}^{n} \sum_{i=1}^{k} g(i) = \sum_{i=1}^{n} \sum_{i=1}^{k} g(i) = \sum_{i=1}^{n} (n - i + 1)g(i).
\]

Based on the similarities between the above expression for AUUC and the DCG measure from Equation (15), we introduce a new metric for UM with L2R called the **promoted cumulative gain**:

\[
\text{PCG}(k) = \sum_{i=1}^{k} (n - i + 1)g(i).
\]

In the DCG measure, each element has a relevance \( \text{rel} \), which is represented by \( g(i) \) in our case, and instead of discounting by \( 1/\log_2(i + 1) \) we promote each element by \( (n - i + 1) \). Finally, along the lines of the DCG measure, PCG allows optimizing the ranking of the top-\( k \) instead of the full dataset as well.

A similar analysis can be made for the separate setting, i.e., when the treatment and control groups are ranked separately, based on the Separate Relative Uplift Curve. The PCG can then be obtained as follows (see Appendix A for details), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2020.3048510:

\[
\text{PCG}(k_T, k_C) = \text{PCG}_T(k_T) + \text{PCG}_C(k_C)
\]

\[
\quad = \sum_{i=1}^{k_T} (|T| - i + 1)g(i) + \sum_{i=1}^{k_C} (|C| - i + 1)g(i),
\]

with \( k_T = p|T| \) and \( k_C = p|C| \) for some percentage \( p \). Hence, the PCG can be computed on both subsets separately and
summed up. In an L2R setup, that means (i) creating two queries, one for the treatment and one for the control group, (ii) using the PCG measure for each query, and (iii) aggregating the resulting PCG values. Note that the $k$ is different for each query, which requires a small modification to the learning systems.

### 3.2.3 Summary

In this section we summarize the different steps needed to use L2R for UM.

**Step 1.** Choose one of the two settings: the separate or joint setting. Note that the latter is closer to how uplift models are to be used in practice as producing a ranking for new data implies that none of these new instances has yet been treated.

**Step 2.** Select a target UM evaluation measure from Table 2 (conditional on the choice made in Step 1) or use PCG (see Section 3.2.2) to assign relevance values to (training) instances (see Tables 3 and 4 and Section 3.2.2).

**Step 3.** Use an existing L2R technique such as LambdaMART to optimize the full ranking or the top-$k$ according to a selected L2R metric, for either one or two queries, depending on the choice made in Step 1.

## 4 Experiments

In this section, we present several experiments to investigate whether and how L2R techniques can be successfully used in the context of UM. In Section 4.1, we first provide information on the datasets and software used. Section 4.2 covers the first experiment in which we analyze the differences between the value function definitions of the UM performance measures (Table 2) through simulation. This experiment provides useful insights to guide the selection of a target UM evaluation measure for UM with L2R (see Step 2 in Section 3.2.3). In Section 4.3, we compare the performance of a traditional pointwise UM approach, i.e., the flipped label approach, to those of similar listwise L2R approaches. Section 4.4 looks at how the use of different relevance values than those presented in Tables 3 and 4 for a specified target UM evaluation measure affects the performance of the L2R approach (see Step 2 in Section 3.2.3). In Section 4.5, we investigate the effectiveness of optimizing rankings for the top-$k$ instead of the full dataset (see Step 3 in Section 3.2.3). Finally, in Section 4.6 we compare the performance of the best performing L2R setup to state-of-the-art UM techniques.

### 4.1 Experimental Setup

#### 4.1.1 Datasets

We use three publicly available datasets originating from randomized controlled trials for the experiments. An overview of the characteristics of the datasets is provided in Table 5.

**Table 5**

| Description | Information | Hillstrom | Criteo |
|-------------|-------------|-----------|--------|
| Channel     | Insurance   | Online clothing | Marketing |
| Channel     | E-mail      | E-mail    | Advertisement |
| Total size  | 10,000      | 64,000    | 25,309,483     |
| # Treatment observations | 4,972 | 21,387 | 21,409 |
| # Control observations | 5,028 | 21,306 | 3,901 |
| # Variables | 68          | 10        | 14        |
| Response variable (binary) | Purchase | Visit | Visit |
| Treatment-to-control size ratio | 0.99:1 | 1:1 | 5.48:1 |
| Treatment positive rate | 20.37% | 15.14% | 4.41% |
| Control positive rate | 19.55% | 10.62% | 2.61% |
| Uplift initial campaign | 0.82% | 4.52% | 1.80% |

The first dataset is part of the Information R package. The data relates to a marketing campaign in the insurance industry and the response variable indicates whether or not a purchase happened. The second dataset is published on the website MineThatData and contains data on an e-mail marketing campaign concerning clothing merchandise. The dataset includes three response variables: ‘visit’, ‘purchase’ and ‘conversion’. The first two are binary variables, while the third is a numerical response variable that represents the amount of money spent. The dataset includes 64,000 observations with $1/3$ targeted with an e-mail campaign concerning men’s clothing, $1/3$ targeted with an e-mail campaign concerning women’s clothing and $1/3$ not targeted. For this dataset, in line with [17] to facilitate comparison, the ‘visit’ variable is selected as the response variable of interest and the selected treatment is the e-mail campaign for women’s clothing (reducing the dataset to 42,693 observations). The last dataset is obtained from the Criteo AI Lab [12] and contains data resulting from several incrementality tests in advertising. The dataset consists of more than 25 million observations, but due to computational reasons, a random subsample of 0.1 percent is used, reducing the dataset size to 25,310 observations.

#### 4.1.2 Software

The L2R technique employed in the experiments is LambdaMART, for which we use the implementation from the open-source RankLib package which is implemented in Java. As LambdaMART relies on gradient boosted trees (see Section 3), for fair comparison, we also use gradient boosted trees for all traditional UM approaches. For this, we rely on the implementation in the xgboost R package [47].

Random stratified sampling was applied to the treatment and control groups of each dataset to split them into 80 percent training and 20 percent test data while

---

1. https://cran.r-project.org/web/packages/Information/index.html
2. https://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html
3. https://ailab.criteo.com/criteo-uplift-prediction-dataset/
4. https://sourceforge.net/p/lemur/wiki/RankLib/
preserving the overall response rate within each group. All reported results are on the test set, unless explicitly stated otherwise. Parameter tuning was done upfront based on the performance on a validation set containing 20 percent of the data and hyperparameters are kept identical for all experiments in the paper (500 trees and a learning rate of 0.01). Each experiment is repeated 10 times. Reported results are averages of these different runs, and in plots we visualize the range between the minimum and the maximum value as a shaded area.

4.2 Experiment 1: Comparing UM Performance Measures Through Simulation

To optimize L2R techniques for UM, we need to select a UM evaluation measure from Table 2. From the table, one can see that the normalization used to account for possible differences in treatment and control group sizes is a differentiating factor. To better understand these differences, we simulate data for three different scenarios with regard to the sizes of the treatment and control groups, produce a ranking for each scenario, and compare how these rankings are evaluated by the different UM evaluation measures presented in Table 2.

The simulation consists of two populations: the treatment group with a response rate of 7 percent and the control group with a response rate of 5 percent. Each instance is assigned a value for \( \hat{t}(x) \) between 0 and 1 depending on the category the instance is in (i.e., TR, TNR, CR or CNR). To simulate uplift, we use the following procedure: for a TR and CNR, we uniformly sample from the interval \([0, 1.0]\), while for a CR and TNR we sample from the interval \([0.0, 0.8]\). These intervals ensure that TR and CNR instances will appear higher in the overall ranking than CR and TNR instances. In a next step, we sample from both treatment and control groups to create three scenario’s: (1) a balanced setting with an equal number of instances in both groups \(|T| = |C|\), (2) an imbalanced setting with a larger treatment group \(|T| = 9|C|\) and (3) an imbalanced setting with a larger control group \(|C| = 9|T|\).

Because each instance is assigned a value for \( \hat{t}(x) \), we can create a ranking for each scenario and compare how these rankings are evaluated by the different UM metrics. To this end, we visualize the different curves (and corresponding baselines) obtained for both the absolute and relative definitions in Figs. 2 and 3, respectively. We split up the analysis for the absolute and relative definitions as they have different units of measurement.

For the absolute definitions, we observe in Fig. 2 that the Qini curves behave quite similar for the separate and joint setting at first sight. However, a closer inspection shows that higher a Qini Curve, and thus a higher AUUC, is obtained for the separate setting. For the Qini curves, AUUC values are all positive but differ significantly depending on the sizes of the groups. For the Uplift curves, on the other hand, the behavior of the curves in separate and joint setting, as well as for the different scenarios, is very different. Moreover, for the Separate Absolute Uplift Curve, AUUC values turn negative when the control group is significantly larger than the treatment group,
which is due to the fact that there is no normalization between the treatment and control groups for this UM performance measure.

For the two relative definitions in Table 2, we observe in Fig. 3 that there are no differences in the curves (and the corresponding AUUC values) for the different scenarios in the separate setting. Recall from Section 2.2 that the Separate Relative Uplift Curve is equivalent to the Separate Absolute Qini Curve up to the constant scaling factor $|T|$, which implies that the curves in Fig. 3a are rescaled versions of the ones in Fig. 2a, and that the same ranking would be obtained when these measures are used for optimization. For the joint setting in Fig. 3b, in contrast to the separate setting, we observe that a different scenario results in a shift of the Uplift Curve. A possible explanation is that one group is overrepresented in the top fraction of the ranking, causing the uplift to be one-sided. However, the AUUC values are very similar. Finally, by comparing the separate and joint settings for the balanced scenario in Fig. 3c, we observe that we get a higher Uplift Curve for the separate setting, and thus a higher AUUC. This is also true for the other scenarios. The reason is that, in the separate setting, the number of TR instances in the top fraction is much higher than in the joint setting, in which we see relatively more CNR instances in the top fraction, which heavily influences the Uplift curves in the early parts of the plot.

Given the above comparisons for the simulated rankings, we consider the relative definitions to be more robust to differences in treatment and control group sizes as these UM evaluation measures are (rather) stable under the three different scenarios. As this can be considered a useful property, in the experiments that follow, we use the relative UM measures, both in separate and joint setting.

### 4.3 Experiment 2: Pointwise UM Versus Listwise L2R

Recall that the flipped label approach can be considered as a pointwise L2R method in which a single model has to rank TRs higher than TNRs and CNRs higher than CRs. Hence, both TRs and CNRs are assigned a new label $z = 1$, whereas the rest is labeled $z = 0$ with $z \in \{0, 1\}$. For the flipped label approach, there is thus no need to separate instances in different groups. In this experiment, we compare this baseline uplift model with the listwise L2R technique LambdaMART. In this regard, recall from Section 4.1.2 that we use gradient boosted trees as learning algorithm for all traditional UM approaches to ensure a fair comparison with LambdaMART.

In the flipped label approach, no distinction is made between TRs and CNRs, and TNRs and CRs. In L2R context, this is similar to the separate setting as the relevant instances of each query have no relation to each other. Therefore, we compare the flipped label approach to L2R by using LambdaMART for two separate queries, one for the treatment group and one for the control group. As for the relevance values we use binary values in accordance with the labels used in the flipped label approach.

Typically, LambdaMART is used to optimize only the top-$k$ of the ranking (as users of for example search engines typically only focus on the first $k$ results), however, in order to compare with the baseline uplift model we use LambdaMART to optimize over all instances. Therefore, in this experiment, $k$ is set equal to the number of training instances in the group considered (either treatment or control). Finally, we use LambdaMART to optimize four different metrics: MAP, DCG, NDCG and PCG.

Fig. 4 shows the relative Uplift curves for the baseline uplift model and the different LambdaMART setups for the three different datasets. On the Information dataset (Fig. 4a), the pointwise UM approach performs better than the standard LambdaMART setups based on DCG, NDCG and MAP. However, for LambdaMART optimized with our PCG metric, the Uplift curve shows higher cumulative incremental gains in the early parts of the plot when compared to the pointwise UM approach. On the Hillstrom dataset (Fig. 4b), LambdaMART with MAP, DCG, NDCG and PCG perform equally well compared to the pointwise UM approach, and perform significantly better for proportions of the population treated above 40 percent. Finally, on the Criteo dataset (Fig. 4c), all techniques achieve high cumulative incremental gains in the early parts of the plot (first 10 percent of the population treated). However, for proportions of the population treated between 10 and 40 percent, the pointwise UM approach performs worse than the listwise L2R approaches.

We further analyze the results by examining the AUUC values presented in Table 6. The pointwise UM approach performs better than the listwise L2R approaches on the Information dataset with only LambdaMART PCG coming close. However, the listwise L2R approaches marginally perform better on both the Hillstrom and Criteo datasets.

In summary, the results of this experiment indicate that listwise L2R approaches can be viable alternatives to pointwise UM approaches. Further, we observe that, for the datasets considered, using the proposed PCG metric to optimize LambdaMART always leads to a better result compared to using one of the other L2R metrics considered.
4.4 Experiment 3: Different Sets of Relevance Values

In this experiment, we investigate how the use of different relevance values than those presented in Tables 3 and 4 for a specified target UM evaluation measure affects the performance of the L2R approach. In this regard, recall that in Section 3.2.1 we determined appropriate relevance values for each UM evaluation measure. Based on these values, presented in Tables 3 and 4, and the relevance values used for the flipped label approach in Section 4.3, we can create the four different sets of relevance values presented in Table 7. We test these different relevance value sets for both the separate and joint setting and use DCG, NDCG and PCG as optimization metrics for LambdaMART. These metrics are chosen because they can handle graded relevance values.

For the use of separate queries, the results are reported in Table 8 in terms of AUUC values of the Separate Relative Uplift Curve. We observe that in all possible settings, PCG consistently performs best when compared to other metrics. On the Information dataset, the NDCG performs significantly worse compared to the other approaches, with the ‘absolute relevance 3’ setting being the exception, however, on the Hillstrom and Criteo datasets, the performance of the NDCG metric is fairly equal to that of the DCG metric. More interestingly, despite the fact that PCG with ‘relative relevance’ labels is exactly the same as optimizing AUUC, using one of the absolute relevance label sets performs marginally better on the Information and Hillstrom datasets. For the use of a joint query, the results are reported in Table 9 in terms of AUUC values of the Joint Relative Uplift Curve. Also in this setting, we observe that PCG consistently outperforms the other metrics. Furthermore, now an absolute relevance value set performs best for all datasets and metrics (while also here the use of ‘relative relevance’ labels in combination with PCG is equivalent to optimizing AUUC). Finally, also note that the NDCG shows improved and nearly equal results to those of DCG on all datasets, including the Information dataset. This is as expected, as in theory, NDCG and DCG should produce equal results when there is only one query.

In summary, the above results are somewhat surprising. While they do confirm that PCG is better suited to the task, the use of less theoretically motivated relevance values is shown to be able to produce good results too.

4.5 Experiment 4: Optimizing Rankings for the Top-\(k\)

In UM, as in information retrieval, often only the top-\(k\) fraction of the ranking is of interest to the user. For example, in

![Experiment 2. Relative Uplift curves in separate setting. Black color is used to represent random treatment assignment, blue represents the baseline uplift model and the other colors represent the different LambdaMART setups.](image)

**TABLE 6**

Experiment 2. AUUC Values of the Separate Relative Uplift Curve for the Baseline Uplift Model and the Different LambdaMART Setups

| Technique               | Information | Hillstrom | Criteo |
|-------------------------|-------------|-----------|--------|
| Flipped label approach  | 0.02052     | 0.02858   | 0.01479|
| LambdaMART MAP          | 0.01237     | 0.03038   | 0.01556|
| LambdaMART DCG          | 0.01520     | 0.02960   | 0.01522|
| LambdaMART NDCG         | 0.00935     | 0.03032   | 0.01523|
| LambdaMART PCG          | 0.01938     | 0.03077   | 0.01578|

*Values in bold: best value on that dataset.*

**TABLE 7**

Experiment 3. Different Sets of Relevance Values

| Set                      | TR  | TNR | CR  | CNR |
|--------------------------|-----|-----|-----|-----|
| Absolute relevance 1     | 1   | 0   | 0   | 1   |
| Absolute relevance 2     | 1   | 0   | -1  | 0   |
| Absolute relevance 3     | 3   | 1   | 0   | 2   |
| Relative relevance       | 1/3 | 0   | -1/3| 0   |

**TABLE 8**

Experiment 3. AUUC Values of the Separate Relative Uplift Curve

| Set                      | DCG  | NDCG | PCG  | DCG  | NDCG | PCG  | DCG  | NDCG | PCG  |
|--------------------------|------|------|------|------|------|------|------|------|------|
| Absolute relevance 1     | 0.01520 | 0.00935 | 0.01938 | 0.02960 | 0.03032 | 0.03077 | 0.01522 | 0.01523 | 0.01578 |
| Absolute relevance 2     | 0.01520 | 0.00678 | 0.01938 | 0.02960 | 0.02893 | 0.03077 | 0.01522 | 0.01555 | 0.01578 |
| Absolute relevance 3     | 0.01520 | 0.01524 | 0.01938 | 0.02960 | 0.02953 | 0.03077 | 0.01522 | 0.01538 | 0.01578 |
| Relative relevance       | 0.01382 | 0.00677 | 0.01829 | 0.02961 | 0.02893 | 0.03055 | 0.01549 | 0.01555 | 0.01601 |

*Values in bold: best value in that column. Underlined values: best value on that dataset.*
marketing or churn prediction, the size of the fraction is determined by the campaign budget. One of the properties of LambdaMART and other L2R systems is their ability to optimize for the top-$k$ specifically. Current UM techniques, however, all aim to optimize the entire ranking. Therefore, in this experiment, we investigate whether specifically optimizing for a specific top-$k$ by using L2R can be successfully applied to UM. In the experiments above, we always optimized the rankings for the entire datasets by setting $k$ equal to the number of training instances in the group considered (either treatment, control or both).

In this experiment, we optimize LambdaMART for $k$ values equal to 10, 30 and 50 percent. For each of these $k$ values, we then check the Uplift curves and AUUC values for both separate and joint settings. Additionally, we compare the results with our previous experiments in which we optimized the rankings for the entire datasets. We present the results of LambdaMART optimized with our PCG metric. We further use for relevance values the ‘relative relevance’ set as this is closest to directly optimizing AUUC. We also carried out this experiment with the DCG and NDCG metrics used for optimization, and with relevance values as in the ‘absolute relevance 3’ setting, however, these settings provided similar insights and therefore the results are not reported for brevity.

For each model, we visualize the Uplift curves from both an optimization and generalization perspective. The optimization perspective shows the results of the models when evaluated on the training set (which is only indicative of the effect of training). The generalization perspective shows the results of the models when evaluated on the test set (the proper way to evaluate the model). The results are shown in Figs. 5 and 6 for the separate and joint settings, respectively.

The plots from an optimization perspective for both the separate and joint setting show that the L2R techniques can optimize for a specific $k$ value. We see that these Uplift curves have change points, after which their behavior changes. These effects are most clearly visible in the joint setting, where it is especially pronounced for $k = 10\%$, for which we observe peaks around the 10 percent marks, followed by declines in the cumulative incremental gains.

When investigating the effect of optimizing for a specific top-$k$ on the test set, i.e., how well the results generalize to new data, the figures provide a less clear answer. Table 10 provides a closer look by presenting the AUUC values of the relative Uplift curves from the generalization perspective at the different $k$ values. We observe that optimizing for a specific $k$ value shows better performance than optimizing for the entire dataset only in some cases (mainly on the Criteo dataset). However, in general, we see that there is no direct relation

![Table 9](image)

**Table 9.** Experiment 3. AUUC Values of the Joint Relative Uplift Curve

| Set                | Information | Hillstrom | Criteo |
|--------------------|-------------|-----------|--------|
|                    | DCG         | NDCG      | PCG    |
| Absolute relevance 1 | 0.01396     | 0.01452   | 0.01940 |
| Absolute relevance 2 | 0.01101     | 0.01116   | 0.01536 |
| Absolute relevance 3 | 0.01563     | 0.01473   | 0.02300 |
| Relative relevance  | 0.01052     | 0.01031   | 0.01573 |

Values in bold: best value in that column. Underlined values: best value on that dataset.

For each model, we visualize the Uplift curves from both an optimization and generalization perspective.
between optimizing for a specific \( k \) value and the results obtained for that value on the test set, nor for other values. Hence, this is a negative result: while learning to optimize AUUC up to a specific cutoff is possible, there is no significant benefit compared to optimizing for the entire dataset (and total AUUC) on the three datasets used in our experiments.

### 4.6 Experiment 5: LambdaMART PCG versus State-of-the-Art UM Techniques

In this last experiment we compare the L2R LambdaMART technique, used in combination with our PCG metric, to state-of-the-art UM techniques. In experiment 2 (Section 4.3), we focused on comparing the performance of different LambdaMART setups to that of the flipped label approach. By contrast, in this experiment we focus on comparing the performance of the best LambdaMART setup to that of the multiple state-of-the-art UM techniques. Next to the flipped label approach, we also consider the dummy treatment approach, the two model approach, and the uplift random forest (see Section 2.1). Of all these techniques, the uplift random forest shows the most consistent performances according to a previous benchmark study [5]. For the L2R setup, we opt for LambdaMART with PCG and ‘relative relevance’ values in a separate setting. We also carried out the experiment with the ‘absolute relevance’ values, however, we do not report the results as they lead to the same conclusions.

![Relative Uplift Curves](image)

**Fig. 6.** Experiment 4. Relative Uplift curves in joint setting at multiple cutoffs. The first row is tested on the training set. The second row is tested on the test set.

**TABLE 10**

| Experiment 4. AUUC Values of Relative Uplift Curves at Multiple Cutoffs on the Test Set |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
|                                 | Separate Relative AUUC at cutoff |                                 | Joint Relative AUUC at cutoff |
|                                 | 10% | 30% | 50% | 100% | 10% | 30% | 50% | 100% |
| **Information**                 |     |     |     |      |     |     |     |      |
| PCG @ 100%                      | 0.00037 | 0.00384 | 0.00900 | 0.02178 | 0.00034 | 0.00271 | 0.00616 | 0.01573 |
| PCG @ 10%                       | 0.00026 | 0.00159 | 0.00368 | 0.01150 | 0.00040 | 0.00232 | 0.00495 | 0.01264 |
| PCG @ 30%                       | 0.00030 | 0.00299 | 0.00731 | 0.01819 | 0.00030 | 0.00246 | 0.00611 | 0.01608 |
| PCG @ 50%                       | **0.00049** | **0.00419** | **0.00942** | **0.02143** | 0.00021 | 0.00257 | 0.00646 | 0.01690 |
| **Hillstrom**                   |     |     |     |      |     |     |     |      |
| PCG @ 100%                      | 0.00037 | 0.00332 | 0.00947 | 0.03060 | 0.00037 | 0.00319 | 0.00905 | 0.03027 |
| PCG @ 10%                       | 0.00032 | 0.00318 | 0.00901 | 0.02980 | 0.00029 | 0.00296 | 0.00853 | 0.02920 |
| PCG @ 30%                       | 0.00037 | 0.00323 | 0.00934 | 0.03045 | 0.00035 | 0.00313 | 0.00887 | 0.03001 |
| PCG @ 50%                       | **0.00038** | **0.00321** | **0.00935** | **0.03043** | 0.00021 | 0.00257 | 0.00666 | 0.01690 |
| **Criteo**                      |     |     |     |      |     |     |     |      |
| PCG @ 100%                      | 0.00059 | 0.00354 | 0.00671 | 0.01575 | 0.00088 | 0.00380 | 0.00691 | 0.01554 |
| PCG @ 10%                       | 0.00063 | 0.00353 | 0.00681 | 0.01577 | 0.00087 | 0.00392 | 0.00725 | 0.01599 |
| PCG @ 30%                       | 0.00062 | 0.00368 | 0.00706 | 0.01608 | 0.00092 | 0.00383 | 0.00689 | 0.01545 |
| PCG @ 50%                       | 0.00058 | 0.00353 | 0.00676 | 0.01580 | 0.00088 | 0.00377 | 0.00684 | 0.01533 |

The left half is in the separate setting and the right half in the joint setting. Values in bold: performance of PCG @ \( k \) is higher than PCG @ 100 percent in that cell. Underlined values: best value in that cell.
technique. However, on the Hillstrom and Criteo datasets, LambdaMART PCG shows improved performance over the UM techniques. On the Hillstrom dataset, LambdaMART PCG achieves 4 percent cumulative incremental gains as the only technique at 50 percent of the population treated, whereas the other techniques only achieve this when treating almost everyone. For the Criteo dataset, we observe that the flipped label approach performs well when treating only 10 percent of the population, however, for higher treatment percentages LambdaMART PCG outperforms the UM techniques. Also notice that some UM techniques even perform worse than the baseline for very high treatment percentages.

We further analyze the results by examining the AUUC values presented in Table 11. On the Information dataset, all UM techniques perform better in terms of AUUC than LambdaMART PCG. However, on both the Hillstrom and Criteo datasets, LambdaMART PCG performs better in terms of AUUC than the UM techniques. Additionally, we present the AUUC results for the relative joint setting, however, based on the results the same conclusions can be drawn. In summary, these results thus show that the L2R LambdaMART technique can compete with existing state-of-the-art UM techniques, and in some scenarios can even do better.

5 DISCUSSION

In our first experiment, we compared different definitions of the Qini and Uplift curves by analyzing performance results on simulated rankings. The results showed that the sizes of the treatment and control groups heavily influence the cumulative incremental gains when expressed in absolute terms. Expressing the cumulative incremental gains in relative terms results in a more robust evaluation. Based on these results, we continued our experiments by focusing on the relative evaluation measures (and their corresponding AUUC values) for both the separate and joint settings, in which we rank the treatment and control groups separately or as one joint group, respectively. Note that in this work we only consider the binary treatment case, i.e., an instance is either treated or not treated. However, this could be extended to multiple treatments by creating a query for each treatment as this is very straightforward in L2R.

In the second experiment, listwise L2R techniques are compared to the pointwise UM flipped label approach in the separate setting. We test LambdaMART with both standard L2R metrics, such as DCG and NDCG, and with our own AUUC-centric metric, PCG. The results show that LambdaMART with PCG performs equal to or better than the flipped label approach which already indicates that listwise L2R techniques can be viable alternatives to pointwise UM techniques.

In the third experiment, we investigated whether the use of different sets of relevance values significantly affects the performance of the L2R approaches. The results of this experiment clearly indicate that our PCG metric outperforms standard L2R metrics in the different settings considered. When optimizing PCG with the ‘relative relevance’ values, we directly optimize the AUUC. However, the results show that the use of less theoretically motivated relevance values also produces good results, and even marginal improvements in performance. In the separate setting, this can be explained by the fact that changing the relevance values does not affect the preference orders. However, in the joint setting, changing the relevance values does affect the preference orders between the four categories (TR, TNR, CR, and CNR). Future research could investigate the effects of different sets of relevance values. Moreover, this also paves the way to future research on multipartite ranking methods in a UM context.

| Technique                        | Information | Hillstrom | Criteo |
|----------------------------------|-------------|-----------|--------|
| LambdaMART PCG                   | 0.01829     | 0.03055   | 0.01601|
| Dummy treatment approach         | 0.02392     | 0.02935   | 0.01165|
| Two model approach               | **0.02610** | 0.02820   | 0.01213|
| Flipped label                    | 0.02052     | 0.02858   | 0.01479|
| Uplift random forest             | 0.02210     | 0.02744   | 0.01287|

| Technique                        | Information | Hillstrom | Criteo |
|----------------------------------|-------------|-----------|--------|
| LambdaMART PCG                   | 0.01971     | 0.03057   | 0.01677|
| Dummy treatment approach         | 0.02283     | 0.02950   | 0.01181|
| Two model approach               | **0.02578** | 0.02840   | 0.01224|
| Flipped label                    | 0.02050     | 0.02865   | 0.01418|
| Uplift random forest             | 0.02163     | 0.02746   | 0.01174|

Values in bold: best value on that dataset.
The fourth experiment investigated the potential of optimizing rankings for the top-k in UM. From an optimization perspective, we often did observe a change in behavior as change points in performance were clearly identified around the k value used in optimization. This effect was observed in both the separate and joint settings, however, it is more distinguishable in the joint setting, indicating that the joint setting is more suitable for optimization. However, these results did not generalize to the test set. This experiment thus indicates that optimizing rankings for top-k fractions of the population is possible, but that there is no significant benefit compared to optimizing for the entire dataset.

Finally, in the fifth experiment, we compared the performance of LambdaMART PCG, which is the best performing L2R setup, to those of existing state-of-the-art UM techniques. The UM techniques show varying performances on all datasets. On the Information dataset, the UM techniques do seem to perform better than LambdaMART PCG. However, on the Hillstrom and Criteo datasets, the results show that LambdaMART PCG outperforms the UM techniques in terms of AUUC. These results indicate that LambdaMART PCG is able to better identify the most impactful instances for smaller proportions of the population treated compared to the UM techniques. Therefore, we can conclude that L2R techniques can be added to the UM toolbox of techniques.

6 Conclusion

Causal classification models estimate for each instance the causal effect of a treatment on an outcome variable of interest, i.e., the individual treatment effect. If both treatment and outcome are binary variables, ITE estimates allow one to rank instances from a large positive effect to a large negative effect. In uplift modeling, one is exactly interested in this ranking rather than in the ITE estimates themselves, as the aim is to identify the instances that are most likely to respond as an effect of being treated. Uplift models estimate the ITE and then use it to build a ranking from which a fraction is selected for treatment. On the other hand, Learning to Rank techniques comprise techniques specifically designed to optimize the quality of predicted rankings directly, rather than the quality of predicted values that serve to rank instances. This paper explores the possibility of using L2R techniques, more specifically the well-known LambdaMART technique, in a UM context.

Before UM was cast as an L2R problem, an analysis of the current evaluation metrics of UM was done. This analysis shows conflicting definitions in the literature. The main differences among definitions are (1) whether the treatment and control groups are considered as separate groups or as one joint group and (2) whether the cumulative incremental gains are expressed in absolute units or in relative percentages.

Our experiments show that standard L2R techniques can be viable alternatives to UM techniques by comparing their performances in terms of Area Under the Uplift Curve (AUUC) for two UM metric definitions selected (the Separate Relative Uplift Curve and Joint Relative Uplift Curve). With the promoted cumulative gain we have created a new L2R metric which promotes relevance values of instances earlier in the ranking, instead of discounting relevance values of instances further in the ranking. The PCG is exactly the AUUC metric from UM and can be readily used by the LambdaMART L2R technique. Moreover, using the proposed PCG metric to optimize LambdaMART is shown to produce better results in terms of AUUC than using standard L2R metrics and to achieve equal or better results than the baseline uplift model (i.e., the flipped label approach).

This paper also tested the effectiveness of optimizing rankings for the top-k instead of the full dataset in a UM context. Models were trained to optimize their rankings for the top 10, 30, 50 and 100 percent. While the results show that learning to optimize rankings for a specific top fraction of the population is possible, from a generalization perspective, no significant benefits can be observed compared to optimizing for the entire dataset.

Finally, with the last experiment we have shown that LambdaMART PCG can compete with existing state-of-the-art UM techniques and even lead to improved performances in terms of AUUC. Overall, our results confirm that L2R can be regarded as a viable alternative to the existing UM methodology by focusing on modeling the ranking directly instead of predicting values that are used to produce a ranking. This work brings up new research questions for future research:

- Future research could look into other listwise L2R techniques.
- Further investigating the effects of using different relevance values for CNRs and TNRs is another possibility.
- The potential of multipartite ranking methods [48] could be explored in UM context, which is related to the previous point.
- One of the open questions in UM is the use case of having multiple treatments [21]. The L2R framework readily allows one to plug in multiple treatments by considering each treatment as a separate query.
- Finally, as illustrated with the PCG metric, the L2R framework allows optimizing rankings for custom metrics. This creates the opportunity to include profit-centric metrics into the modeling phase. Future research could for example look into integrating the Maximum Profit Uplift (MPU) metric, specifically created for UM [13], into the L2R framework with the aim of obtaining rankings that result in higher profits.

References

[1] S. Wager and S. Athey, “Estimation and inference of heterogeneous treatment effects using random forests,” J. Amer. Statist. Assoc., vol. 113, no. 523, pp. 1228–1242, 2018.
[2] K. Imai et al., “Estimating treatment effect heterogeneity in randomized program evaluation,” The Ann. Appl. Statist., vol. 7, no. 1, pp. 443–470, 2013.
[3] M. Qian and S. A. Murphy, “Performance guarantees for individualized treatment rules,” Ann. Statist., vol. 39, no. 2, 2011, Art. no. 1180.
[4] U. Shalit, F. D. Johansson, and D. Sontag, “Estimating individual treatment effect: Generalization bounds and algorithms,” in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 3076–3085.
[5] F. Devriendt, D. Moldovan, and W. Verbeke, “A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics,” Big Data, vol. 6, no. 1, pp. 13–41, 2018.
[6] J. M. Gubela, S. Lessmann, and S. Jaroszewicz, “Uplift modeling with survival data,” in Proc. ACM SIGKDD Workshop Health Inform., 2014.
[7] R. M. Gubela, S. Lessmann, and S. Jaroszewicz, “Response transformation and profit decomposition for revenue uplift modeling,” Eur. J. Oper. Res., vol. 283, no. 2, pp. 647–661, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0377221719309415
[8] E. Ascarza, “Retention futility: Targeting high-risk customers might be ineffective,” J. Marketing Res., vol. 55, no. 1, pp. 80–98, 2018.

[9] T.-Y. Liu et al., “Learning to rank for information retrieval,” Found. Trends® Inf. Retrieval, vol. 3, no. 3, pp. 225–331, 2009.

[10] D. B. Rubin, “Causal inference using potential outcomes: Design, modeling, and identification,” J. Amer. Statist. Assoc., vol. 100, no. 469, pp. 322–331, 2005.

[11] H. Nassif, F. Kuusisto, E. S. Burnside, and J. W. Shavlik, “Decision trees for uplift modeling,” in J. Amer. Statist. Assoc., vol. 1, no. 3, pp. 14–21, 2007.

[12] H. Nassif, F. Kuusisto, E. S. Burnside, and J. W. Shavlik, “Uplift modeling with ROC: An SRL case study,” in Proc. ILP (Late Breaking Papers), 2013, pp. 40–45.

[13] M. Soza, “Reinforcement learning for marketing campaigns,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2015, pp. 399–407.

[14] J. Yoon, J. Jordon, and M. van der Schaar, “GANITE: Estimation of counterfactual outcomes,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases, 2018, arXiv:1807.07909.

[15] G. V. Kass, “An exploratory technique for investigating large quantities of categorical data,” J. Roy. Statist. Soc.: C (Appl. Statist.), vol. 29, no. 2, pp. 119–127, 1980.

[16] M. Jaskowski and S. Jaroszewicz, “Uplift modeling for clinical decision-making techniques, a new proposal, and an insurance case study,” in Proc. 22nd Int. Conf. Neural Inf. Process. Syst., 2009, pp. 315–323. [Online]. Available: https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/

[17] A. C. R. Anscombe, “Significance tests in regression and correlation,” J. Roy. Statist. Soc. B, vol. 30, no. 1, pp. 99–115, 1968.

[18] D. B. Rubin, “Statistics and causal inference,” in New York, NY, USA: John Wiley & Sons, Inc., 1974.

[19] D. B. Rubin, “Causal inference using potential outcomes: Design, modeling, and identification,” J. Amer. Statist. Assoc., vol. 100, no. 469, pp. 322–331, 2005.

[20] P. Rzepakowski and S. Jaroszewicz, “Uplift modeling in direct marketing,” in J. Telecommun. Inf. Technol., vol. 2, pp. 43–50, 2012.

[21] C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[22] J. M. Reina, “Cumulated gain-based evaluation of IR techniques,” in ACM Trans. Inf. Syst., vol. 20, no. 4, pp. 422–446, Oct. 2002. [Online]. Available: http://dl.acm.org/doi/10.1145/582415.582418

[23] W. Chen, T. van Liu, Y. Lan, Z. Ming Ma, and H. Li, “Ranking measures and loss functions in learning to rank,” in Proc. 22nd Int. Conf. Neural Inf. Process. Syst., 2009, pp. 115–121. [Online]. Available: http://papers.nips.cc/paper/3708-rankalingmeasures-and-loss-functions-in-learning-to-rank.pdf

[24] B. McFee and G. R. Lanckriet, “Metric learning to rank,” in Proc. 27th Int. Conf. Mach. Learn., 2010, pp. 775–782.

[25] C. Burges, T. Serre, and J. Puhringer, “Learning to rank using gradient descent,” in Proc. 22nd Int. Conf. Mach. Learn., 2005, pp. 99–106. [Online]. Available: http://dl.acm.org/doi/10.1145/1012351.1012363

[26] W. B. Croft, D. Metzler, and T. Strohman, Search Engines: Information Retrieval in Practice, vol. 520, Boston, MA, USA: Addison-Wesley Reading, 2010.

[27] T. Chen et al., XGBoost: Extreme Gradient Boosting, 2016. [Online]. Available: https://github.com/dmlc/xgboost

[28] J. Färm, T. Heistermann, M. Heistermann, and T. Engelmann, “A survey of modern data mining methods for direct marketing,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases, 2009, pp. 359–374.

Fioris Devriendt received the master’s degree in computer science from Vrije Universiteit Brussel, Brussels, Belgium, and the PhD degree. He is a researcher at the Data Analytics Laboratory, Solvay Business School, Vrije Universiteit Brussel, Brussels, Belgium. His research focuses on uplift modeling (or treatment effect estimation) and profit maximization techniques in domains such as churn prediction and marketing. He presented his work at several international academic conferences, and his work is published in international journals such as the ‘Big Data’ and the ‘Information Sciences’.
Jente Van Belle received the MS degree in business engineering from the Vrije Universiteit Brussel, Brussels, Belgium, in 2015. He is currently working toward the PhD degree in applied economics at the Data Analytics Laboratory, Solvay Business School, Vrije Universiteit Brussel, Brussels, Belgium. His research focuses on time series forecasting and prescriptive data analytics and is driven by business applications in supply chain and customer relationship management. His work has been published in established international scientific journals such as the *European Journal of Operational Research*.

Tias Guns received the MS degree in computer science from KU Leuven, Belgium, in 2006, and the PhD degree in engineering from the Faculty of Computer Science, in 2012. He is currently an associate professor at the Declarative Languages and Artificial Intelligence (DTAI) research group within the Computer Science Department, KU Leuven, Belgium. His research lies on the border between data mining and constraint programming, and integrating domain expertise and user constraints into data analytics tasks.

Wouter Verbeke received the PhD degree. He is associate professor of data science at the Faculty of Economics and Business, KU Leuven, Belgium. His research is situated in the field of cost-sensitive and causal machine learning for business decision-making. In 2014, he won the distinguished EURO award for best article published in the *European Journal of Operational Research* in the category 'Innovative Applications of O.R.' His work has been published in established international scientific journals such as the *IEEE Transactions on Knowledge and Data Engineering*, *Information Sciences* and, *European Journal of Operational Research*. He has authored two practitioner-oriented books, entitled 'Fraud Analytics Using Descriptive, Predictive & Social Network Techniques' and 'Profit-driven Business Analytics', published by Wiley.

For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdli](http://www.computer.org/csdli).