Learning Continuous Treatment Policy and Bipartite Embeddings for Matching with Heterogeneous Causal Effects

Anonymous Authors[1]

Abstract

Causal Inference algorithms estimate treatment effects to help make action decisions. Most methodologies make decisions on a single dimension of outcome, on a single treatment instance, and with a binary treatment decision variable. For instance, estimate the improvement in one of patient's symptom given the treatment, compared with no treatment at all. These methods fail to capture continuous space action policy with a measure for intensity and complexity of treatment. Most algorithms also lacks consideration for cross-effects when we match treatment subjects with the selection of treatment types. Finally, previous methods do not focus on combined effectiveness of multiple outcomes in a holistic, population wide outcome.

We take a causal perspective and propose a ubiquitous treatment effect optimization framework with deep learning. This framework works with a continuous policy function that maps to intensity or complexity of treatment. This function can be directly optimized for best results. The algorithm deals with matching instances, for example, user with a product, medication to a patient, thus expands to a multitude of treatment complexities with those matches. Finally, we utilize novel deep learning methods to optimize the desired metric space rather than only predicting single dimensional treatment counter-factual, this approach employs a population-wide effectiveness measure and significantly improves overall holistic efficiency. We demonstrate effectiveness of our models which surpasses prior art.

We demonstrate the superior performance of our algorithms observe a X% improvement when using generic continuous space treatments. This captures subtle variations in treatment space, structures the most efficient optimizations, and opens up many application arenas.

1. Introduction

Given a number of patients and a number of medication treatments, how do we decide the medication to match each patient, and the extent of the treatment? The results of the treatment offer guidance to this optimization, to maximize the desired treatment effect. What are the complexities if we take into account the overall effectiveness with healing improvements, side-effects, and patient experience?

Past work in causal inference and machine learning provided algorithms to solve components of the problem. The main problem is to estimate treatment effect, or improvement observe from the subject, if we offer her treatment compared with no treatment. This is a 'counterfactual' (**?**) or 'what-if' argument. Given enough data, the approach can be used to estimate the multitude of counterfactual outcomes, then decisions can be based on weighted measures. In fact multiple works in the literature including highly effective models such as (**?**), based on the estimation of the counterfactual provided by the Average Treatment Effect functionτ. The above-mentioned approach is highly costly due to the number of models we need to train for all treatment matching options for medications, and also the number of outcomes, such as patient experience, or rare side-effect cases. The approach is also less effective since the matching can be correlated with sub-factors such as multiple medications dealing with infection, or correlation across complexities in matching and outcomes.

Our solution is based on constructing a learning objective using a bayesian decomposition. We decompose the posteriors in the ATE τ function's expected value computation, using bayesian decomposition into three components. The first being the instance effectiveness measure, such as user susceptibility to treatment, the second is the likelihood of matching with a treatment type, and the third term is a continuous space action policy for treatment intensity. With these three components and compose of the effectiveness measure which is posterior in the ATE function's weighting.

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

These three components are then parameterized using deep learning models to enable optimization on three separate segments of our study. Finally, we construct a combined objective function with multiple ATE tau functions to account for the multitude of outcomes, e.g. in the patient's treatment event.

Our proposal offers the application of continuous space treatment policy. Instead of evaluating the counterfactual of the outcome if a binary treatment, we assume the treatment is given at continuous levels like the dose of medication or price of a product. We apply continuous distributions that measure likelihood of treatment intensity, and re-weight with the prior distribution. The method, based on bayesian decomposition, offers a generic measure of how the intensity of the treatment affects its effectiveness. Using this methodology, we are able to answer questions such as 'what is the optimal level of treatment we can give for a specific subject?' and 'how is effective is the treatment if it were given 53%?', and apply to problems without constraint to binary treatment decisions.

To resolve formulation of matching the subject with multiple treatment types, we use a bipartite embedding approach combined with multi-layer neural networks. As with above, Bayesian decomposition also factors the effectiveness measure for matching of two items, and we use a deep learning method for parameterization. With layers of neural networks, the features are projected into embedding space and applied a distance measure for relevance matching. This is analogous to applications in learning-to-rank.

At the heart of our proposal is the Bayesian decomposition and its combination with deep learning methods. We derive from first principles the causal treatment effect functions, and represent the treatment effect with effectiveness representations, the representations are decomposed into a chain of probabilities that translate to continuous policy likelihood, the matching likelihood and subject instance prior. The Bayesian decomposition is then parameterized with deep learning algorithms and construct eventual objective using multiple treatment effect functions. The objective error is then back-propagated into the parameterization, eventually optimized with quasi-newton methods, and numerical optimizers. The powerful combination of Bayesian decomposition and deep learning offer the methodology to represent continuous treatment policy and bipartite embeddings for matching, enabling the highly effective techniques for causal learning. Overall, we we provide an algorithmic framework to optimize for the overall effectiveness of treatments, including the Bayesian decomposition into a larger set of controllable decision factors. With the critical decision factors, we can perform learning on the eventual sophisticated objective function that can be arbitrarily defined. Novel contributions of this paper are:

Deep Learning with Causal Effectiveness Objective - We derive from first principles the causal inference objective function to optimizes for treatment effectiveness. Instead of estimating treatment effects of a single outcome, we The objective is generalized into a casual learning loss function that can be utilized in deep learning models. We demonstrate superior algorithmic flexibility of this modeling framework evidenced by experimentation.

Continuous Treatment Policy - We are the first to study continuous treatment intensity into the causal machine learning framework. In this framework, we are able to decompose out the treatment decision and formulate as continuous distributions. Using Bayesian decompositions we define arbitrary continuous and learnable distribution that generates the effectiveness measure and optimize for the eventual objective, while leveraging its training signal to determine optimal continuous treatment level.

Bipartite Embeddings for Relevance Matching - Compared with previous research, our proposal makes decisions on subject instance being matched with types of treatment, rather than only considering subject instance scoring for treatment. We consider the likelihood of match in the perspective of *relevance matching* of bipartite instances, and formulate embedding spaces for both subject instance and matched instance. For example, this can be patient with medication, customer with products.

Dynamic Pooling and Combined Optimization - We formulate a dynamic pooling algorithm for creating pooling modules with sort operations in the ordinal matching scenario. These modules form models to create sparse connections in deep networks, induce invariance while allows overall metric to be corrected through parameter changes in lower layers. This is an all-purpose model that can be used to model both population-wide efficiency, and treatment effects when combined with the deep learning effectiveness objective.

The structure of this paper is as follows: in Section 2, we will cover related work in optimization of treatment effect. In Section **??**, we make the problem statement and introduce effectiveness measures and our modeling approaches for treatment effect optimization. In Section **??**, we will cover experimentation, results, comparisons across models and real-world performance from the product we launched. Finally we briefly cover future research steps.

2. Prior Work

The space of causal inference has been studied in view of treatment effect estimation (Shalit et al., 2017) which uncovers a range of algorithms by balancing representation space, (Louizos et al., 2017) discovers hidden confounding factors in combination with algorithms such as variational auto-encoders, (Parbhoo et al., 2018) investigates from information bottleneck specific to causal algorithms. In terms of temporal effects, (Lim, 2018) adopted recurrent networks to study the long-term causal effects of sequential treatments. There have been many with application to medical problems and precision medicine (Shalit et al., 2017) (Lim, 2018). It's worth noting a classic work (Rubin, 1974) proposed a paradigm for treatment effect estimation. where the outcome of subject treatment is observed and then used to fit a model for counterfactual estimation. Statistical viewpoints have been taken (Künzel et al., 2017) to decomposes the learning algorithm into composite models with meta-learners. Notably, quasi-oracle estimation by (Nie & Wager, 2017) is effective when estimating the treatment effect in a single outcome. Recently, decision trees and random forests (Chen & Guestrin, 2016) have started to be applied (Rzepakowski & Jaroszewicz, 2012), causal tree and random forests (Wager & Athey, 2017) , boosting (Powers et al., 2017) are applied to build causal inference models.

Most previous methods consider the *Average Treatment Effect (ATE)* across a single outcome, with shortfalls of not able to deal with continuous treatments, matching with treatment types, and deal with multiple outcomes to trade-off the dimentions. In this work we propose a framework to work with continuous treatment variables, create a paradigm for matching, and aggregate outcome metric space, eventually optimized this framework directly using deep learning.

3. Algorithms

3.1. Problem Statement: Matching Algorithm with Continuous Space Policy

Consider when a patient, based on her medical history and symptoms, needs to be matched with a treatment. The treatment is characterized with suitable symptoms, and historical statistics. When prescribed, there is a continuous measure for how intense the treatment should be. The goal is match the best treatment, with the optimal treatment intensity. Another example is pricing at an internet company, when a user needs to be matched with any product in the on-line store, and continuous treatment is the price to allocate for the specific user-product match. We solve the matching algorithm with continuous policy with causal learning and optimization with deep learning methods.

The quasi-oracle estimation algorithm, among other meta-learner algorithms (Künzel et al., 2017), is efficient for estimating conditional treatment effects. However, the Using causal learning derivations, our eventual goal is to maximize an effectiveness measure. For instance, problems in causal learning and reinforcement learning domain trades off the cost of action taking, with the reward obtained by those actions (R.S. & A.G., 2018) (Silver et al., 2016) (Kool et al., 2017). Thus effectiveness is a key theme in applications of causal inference. In this paper, we propose causal inference paradigm to maximize cost effectiveness of heterogeneous treatments.

The approach described in the previous section contains two separate steps, treatment effect prediction and constraint optimization.

The objective for our algorithm is identify a *collection of treatment sessions* to achieve highest effectiveness, each *treatment session* is composed of *a pair of subject and treatment candidate*. This does not rely on individual prediction of treatment effect, but rather, achieves the overall holistic effectiveness.

In order to formulate the problem, we introduce a notation of *the overlined treatment effect* $\overline{\tau}^*(\mathbf{x}) = E(Y_1 - Y_0|\mathbf{X} = \{\mathbf{x}^{(i)}\})$, this represents a representation of the treatment effect function across the collection of subjects $\{\mathbf{x}^{(i)}\}$. Also we may refer to parameterization of models using parameters θ and effectiveness measures $p_\theta(\mathbf{x}^{(i)})$ per session i. The models outputs effectiveness measures to correlate with the expectation across the collection, e.g. $\overline{\tau}^*(\mathbf{x}) = \sum_i p(\mathbf{x}^{(i)})Y_1^{(i)} - \sum_j p(\mathbf{x}^{(j)})Y_0^{(j)}$ thus combined into the same expression using $\overline{\tau}^*(\mathbf{x})$.

Note this critical step to define objective in the casual deep learning paradigm to represent a Treatment Effect Function with a parameterization of multiple deep learning models. We then are able to represent the eventual objective function with a combination of multiple $\overline{\tau}^*(\mathbf{x})$. For instance, a model tries to solve unconstrained optimization problems:

$$\text{maximize} \quad \overline{\tau}^{*q}(\mathbf{x})(\overline{\tau}^{*r}(\mathbf{x}) - \overline{\tau}^{*c}(\mathbf{x})) \tag{1}$$

$$\text{minimize} \quad \frac{\overline{\tau}^{*c}(\mathbf{x})}{\overline{\tau}^{*r}(\mathbf{x})} + \overline{\tau}^{*m}(\mathbf{x}) \tag{2}$$

The first problem represents a scheme to maximize a combined treatment effect of gain minus cost weighed by success rate. We aim to optimize for this holistic effectiveness. The cost per unit gain objective allows us to take all users and outcomes into consideration. The unconstrained optimization allows us to utilize deep learning to build flexible models and efficiently train with large-scale data.

Bayesian Decomposition with Causal Inference. Without loss generality we include per sample treatment propensity from causal statistics (Lunceford & Davidian, 2004) and derive $\overline{\tau}$ without superscript as it could extend to both $\overline{\tau}^{*r}$ and $\overline{\tau}^{*c}$. Start from the fundamental definition of treatment effect:

$$\overline{\tau}^* = E(Y_1 - Y_0) = E(Y_1) - E(Y_0)$$

The above treatment effect term is influenced by treatment policy thus given a certain policy $\mathbb{P}$, we can write:

$$\bar{\tau}^* = E(Y_1 - Y_0|\mathbb{P}) = E(Y_1|\mathbb{P}) - E(Y_0|\mathbb{P})$$

Different from prior work (Lunceford & Davidian, 2004) in Causal Inference, we differentiate the Policy $\mathbb{P}$ with the treatment cohort indicator T, the latter T random variable indicates whether an instance is in the treatment cohort or in the control cohort. Only within the treatment cohort, the Policy $\mathbb{P}$ is applied. The Policy can be optimized to produce the best possible outcome in treatment cohort, when treatment cohort variable is correlated with whether assign instances to control or held-out cohort. The propensity in the context of a possible policy evaluation, is the expected value of treatment cohort indicator.

From (Lunceford & Davidian, 2004), the expected value of outcome of any instance can be written as following equations. This takes Treatment Cohort Indicator and propensities into account:

$$E(Y_1|\mathbb{P}) = E(\frac{Y_1 T}{e(\mathbf{x})}|\mathbb{P}), \quad E(Y_0|\mathbb{P}) = E(\frac{Y_0(1-T)}{1-e(\mathbf{x})}|\mathbb{P}) \tag{3}$$

$e(\mathbf{x})$ is a propensity function $E(T=1|\mathbf{X}=\mathbf{x})$. This quantity is estimated given features of the instance, thus can be discriminative per instance. For instance, a learnable propensity function fitted with the feature and treatment cohort indicator labels. Detailed proof of the above equation is given for Y_1 case: $E(Y_1) = E(\frac{Y_1}{e(\mathbf{x})}e(\mathbf{x})) = E(\frac{Y_1}{e(\mathbf{x})}E(T|\mathbf{X}=\mathbf{x})) = E(\frac{Y_1}{e(\mathbf{x})}E(T|\mathbf{X}=\mathbf{x}, Y_1)) = E(E(\frac{Y_1 T}{e(\mathbf{x})}|X, Y_1)) = E(\frac{Y_1 T}{e(\mathbf{x})})$[1]

Substitute into the treatment effect definition:

$$\begin{aligned}\bar{\tau}^* &= E(Y_1 - Y_0|\mathbb{P}) = E(Y_1|\mathbb{P}) - E(Y_0|\mathbb{P}) \\ &= E(\frac{Y_1 T}{e(\mathbf{x})}|\mathbb{P}) - E(\frac{Y_0(1-T)}{1-e(\mathbf{x})}|\mathbb{P}) \\ &= E(E(\frac{Y_1 T}{e(\mathbf{x})}|T,\mathbb{P})) - E(E(\frac{Y_0(1-T)}{1-e(\mathbf{x})}|T,\mathbb{P}))\end{aligned} \tag{4}$$

The last step follows from law of total expectation. We then expand the equation and eliminate zero terms:

$$\begin{aligned}\bar{\tau}^* &= P(T=1)E(\frac{Y_1 T}{e(\mathbf{x})}|T=1,\mathbb{P}) \\ &\quad - P(T=0)E(\frac{Y_0(1-T)}{1-e(\mathbf{x})}|T=0,\mathbb{P})\end{aligned} \tag{5}$$

We expand expectation in the following equations. Every match is decomposed into two instances of the items being matched, which denote with random variables I and J indexed by i and j. We define $\hat{e}$ to be the overall propensity, or likelihood of being in the treatment cohort across all instances. For each user, the $e(\mathbf{x})$ term is estimated from a pretrained propensity function. In the last equation, we introduce Matches of instances denoted by random variable M, and use the tuple notation $M = (I, J)$ and $m = (i, j)$ to indicate each match across two distinct items.

$$\bar{\tau}^* = \hat{e} \sum_{T_{i,j}=1} \frac{p(I=i, J=j|\mathbb{P}^{(i,j)})}{e(\mathbf{x})} Y_1^{(i,j)} \tag{6}$$

$$- (1-\hat{e}) \sum_{T_{i,j}=0} \frac{p(I=i, J=j|\mathbb{P}^{(i,j)})}{1-e(\mathbf{x})} Y_0^{(i,j)} \tag{7}$$

$$= \hat{e} \sum_{T_m=1} \frac{p(M=m)p(\mathbb{P}^{(m)}|M=m)}{e(\mathbf{x})\sum_l p(M=l)p(\mathbb{P}^{(l)}|M=l)} Y_1^{(m)} \tag{8}$$

$$- (1-\hat{e}) \sum_{T_m=0} \frac{p(M=m)p(\mathbb{P}^{(m)}|M=m)}{(1-e(\mathbf{x}))\sum_l p(M=l)p(\mathbb{P}^{(l)}|M=l)} Y_0^{(m)} \tag{9}$$

The last two lines are obtained using Bayes Rule to replace the posterior $p(I=i, J=j|\mathbb{P}^{(i,j)})$ with $\frac{p(M=m)p(\mathbb{P}^{(m)}|M=m)}{Z}$, prior multiplied by likelihood, divided by the normalization factor or partition function Z.

As stated before, we parameterize the above function, then combine the Treatment Effect Functions $\bar{\tau}^*$, we will be able to compose an objective function for the learning algorithm.

Parameterize continuous policy model. We next parameterize the action policy model. The key to defining an objective function composed of conditional average treatment effect (CATE) representations with τ^* is finding representations for the likelihood $p(\mathbb{P}|M=m)$, the probability of a policy given a match, and prior $p(M=m)$, the probability of a match in the previous equation. In this paper, we first solve the problem with a continuous policy problem statement. This is to say, we assume the random variable $\mathbb{P}$ to be a continuous random variable. For instance, we limit its range $\mathbb{P} \in [0,1]$ as intensity of treatment. Given any specific match, we formulate regression models

$$s(\mathbf{x}_i, \mathbf{y}_j)_{\mathbb{P}} = f(\mathbf{x}_i, \mathbf{y}_j) \tag{10}$$

Here the feature vectors $\mathbf{x_i}$ and $\mathbf{y_j}$ denote features to specify the match subject i and object j, e.g. user and product, patient and treatment. There is a prediction model for the best treatment intensity $s_{\mathbb{P}}$ for this specific match of subject and object. Note we can construct any number of these regression models. We can implement this regression function with any differentiable function, such as a logistic regression, or multi-layer neural network. We can define the likelihood $p(\mathbb{P}|M=m)$ to be the following:

$$p(\mathbb{P}|M=m) \propto \sigma(1-\sigma)(s(\mathbf{x}_i, \mathbf{y}_j)_{\mathbb{P}}) \tag{11}$$

[1] Third step follows from unconfoundedness assumption (Lunceford & Davidian, 2004) (Nie & Wager, 2017).

Where σ denotes the sigmoid function $\sigma(x) = \frac{1}{1+\exp^{(-x)}}$ and $\sigma(1-\sigma)$ is the derivative of the sigmoid function. This formulation gives the likelihood a probabilistic interpretation, such that the cumulative density of the random variable takes forms of the sigmoid, and the probabilistic density takes a bell-shaped form. This means whenever the intensity $\mathbb{P}$ deviates from the optimal, the likelihood function penalizes by lowering the value, positioning the highest value at top of bell curve at the regressed value in equation 10.

Alternatively, we can define the likelihood as:

$$p(\mathbb{P}|M = m) = Beta(\mathbb{P}, \alpha(\mathbf{x}_i, \mathbf{y}_j), \beta(\mathbf{x}_i, \mathbf{y}_j)) \qquad (12)$$

Where the random variable $\mathbb{P}$ assumes a density of a Beta distribution with parameters α, β which are derived with a regression function from feature sets $\mathbf{x}$ and $\mathbf{y}$. This second formulation has the interpretation for variable $\mathbb{P}$ to be exactly in range of $[0, 1]$, and if both parameters are limited to be above 1.0, the distribution also takes the hump shape with optimal at $\frac{\alpha}{\alpha+\beta}$.

Parameterize subject-candidate matching model. We next parameterize the matching model model. We can decompose the following equation:

$$\begin{aligned} p(M = m) =& p(I = i, J = j) && (13)\\ =& p(I = i)p(J = j|I = i) && (14)\\ \propto& g(\mathbf{x}_i)m(\mathbf{x}_i, \mathbf{y}_j) && (15) \end{aligned}$$

A simple form of g is a sigmoid compressed version of arbitrary differentiable regression function, and forms the base instance weighting model. With sigmoid as the nonlinearity, the outputs are positive and are well-controlled below 1.0.

For the functional form of m we adopt a bipartite embedding formulation. First the input features $\mathbf{x}$ are projected with a neural network to an embedding space for both subject instance, and matching candidate: $\mathbf{e}_i = f_{sp}(\mathbf{x}_i), \mathbf{e}_j = f_{cp}(\mathbf{y}_j)$. In this manner, the embedding space can be learned using eventual effectiveness measures, across the treatment subjects, e.g. patient, as well as the treatment candidates, e.g. medications. The eventual objective helps to learn the embedding sub-space each of the subject and candidate lies in. Also the algorithm for matching here is highly related to learning-to-rank models such as DSSM (Huang et al., 2013). Where the candidates are projected first into a vector space before the metric distance is used to define the relevance of terms. Note that we use the same projection function in the definition of g. Then, we can formulate the likelihood function:

$$m(\mathbf{x}_i, \mathbf{y}_j) = cos(\phi) + 1 = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i \cdot \mathbf{e}_j\|} + 1 \qquad (16)$$

With the previously defined likelihood functions, we can sum together to form the partition function in order to normalize the product of prior and likelihood over all possible matches in, each indexed by l, written as $\sum_l p(M = l)p(\mathbb{P}^{(l)}|M = l)$. When the partition function can be computed, the posterior can be computed and Equation 6 can be computed to define the overall objective function.

$$\begin{aligned} \bar{\tau}^* &= \hat{e} \sum_{T_i=1} \frac{Y_1^{(i)} p_i}{e(\mathbf{x}^{(i)})} - (1-\hat{e}) \sum_{T_i=0} \frac{Y_0^{(i)} p_i}{1-e(\mathbf{x}^{(i)})} \\ &= \hat{e} \sum_{i=1}^{n} \frac{1}{e(\mathbf{x}^{(i)})} Y_1^{(i)} p_i \mathbb{I}_{T_i=1} - (1-\hat{e}) \sum_{i=0}^{n} \frac{1}{1-e(\mathbf{x}^{(i)})} Y_0^{(i)} p_i \mathbb{I}_{T_i=0} \\ &= \hat{e} \sum_{i=1}^{n} \frac{1}{e(\mathbf{x}^{(i)})} Y^{(i)} p_i \mathbb{I}_{T_i=1} - (1-\hat{e}) \sum_{i=0}^{n} \frac{1}{1-e(\mathbf{x}^{(i)})} Y^{(i)} p_i \mathbb{I}_{T_i=0} \end{aligned} \qquad (17)$$

Eq. 17 is a generalized form of the $\bar{\tau}^*$ including part of the objective function with propensity weighting. When $\hat{e} = e(\mathbf{x}^{(i)})$ is constant in a fully randomized experiment, the term becomes:

$$\bar{\tau}^* = \sum_{i=1}^{n} Y^{(i)} p_i (\mathbb{I}_{T_i=1} - \mathbb{I}_{T_i=0}) \qquad (18)$$

4. Experiments

Benchmark Models ***Quasi-oracle estimation (R-Learner)***. We use Linear Regression[2] as the base estimator. Since the experiment cohorts are randomly selected, we use constant treatment percentage as propensity in the algorithm. Since we need to define one CATE function for R-learner, we use the R-learner to model the gain value incrementality with τ_r.

Causal Forest. We leverage the generalized random forest (*grf*) library in R (Wager & Athey, 2018) (grf). For details, we apply causal forest with 100 trees, 0.2 as alpha, 3 as the minimum node size, and 0.5 as the sample fraction. We apply the ratio of two average treatment effect functions in ranking by training two causal forests. To rank users or other cardinalities with respect to cost vs gain effectiveness, we estimate the conditional treatment effect function both for gain (τ_r) and cost (τ_c), i.e. train two Causal Forest models.

[2] Using SKLearn library's Ridge Regression with 0.0 as the regularization weight.

For evaluation, we compute the ranking score according to the ratio of the two $\frac{\tau_r}{\tau_c}$. For hyper-parameters, we perform search on deciles for parameters *num_trees*, *min.node.size*, and at 0.05 intervals for *alpha*, *sample.fraction* parameters. We also leverage the *tune.parameters* option for the *grf* package, eventually, we found best parameters through best performance on validation set[3].

Simpliflied Session Scoring Model (SSSM). We implement our deep learning based models with Tensorflow. To align with baseline and other methods in our experiments, we use a one layer with linear parameterization $\tanh(\mathbf{w}^T\mathbf{x} + b)$ as the scoring function, without weight regularization, the objective functions stated in the algorithm section are used. We use the Adam optimizer with learning rate 0.01 and default beta values. We compute gradients for entire batch of data, and run for 600 iterations.

Continuous Policy Bipartite Matching Model (CPBMM). We implement our deep learning based models with Tensorflow. To align with baseline and other methods in our experiments, we use a one layer with linear parameterization $\tanh(\mathbf{w}^T\mathbf{x} + b)$ as the scoring function, without weight regularization, the objective functions stated in the algorithm section are used. We use the Adam optimizer with learning rate 0.01 and default beta values. We compute gradients for entire batch of data, and run for 600 iterations.

Datasets In the next subsection we offer a detailed description of datasets we used to benchmark the models. ***US Census 1990*** The US Census (1990) Dataset (Asuncion & Newman, 2007 (usc) contains data for people in the census. Each sample contains a number of personal features (native language, education...). The features are pre-screened for confounding variables, we left out dimensions such as other types of income, marital status, age and ancestry. This reduces features to d = 46 dimensions. Before constructing experiment data, we first filter with several constraints. We select people with one or more children (*'iFertil'* ≤ 2), born in the U.S. (*'iCitizen'* = 0) and less than 50 years old (*'dAge'* < 5), resulting in a dataset with $225,814$ samples. We select 'treatment' label as whether the person works more hours than the median of everyone else, and select the income (*'dIncome1'*) as the gain dimension of outcome for τ_r, then the number of children ('iFertil') multiplied by -1.0 as the cost dimension for estimating τ_c. The hypothetical meaning of this experiment is to measure the cost effectiveness, and evaluate who in the dataset is effective to work more hours.

Covertype Data The Covertype Dataset (Asuncion & Newman, 2007) contains the cover type of northern Colorado forest areas with tree classes, distance to hydrology, distance to wild fire ignition points, elevation, slope, aspect, and soil type. We pre-filter and only consider two types of forests: *'Spruce-Fir'* and *'Lodgepole Pine '*, and use data for all forests above the median elevation. This results in a total of $247,424$ samples. After processing and screening for confounding variables, we use 51 features for model input. With the filtered data, we build experiment data by assuming we are able to re-direct and create water source in certain forests to fight wild fires, but also like to ensure the covertype trees are not imbalanced by changing the hydrology with preference to 'Spruce-Fir'. Thus, the treatment label is selected as whether the forest is close to hydrology, concretely, distance to hydrology is below median of the filtered data. The gain outcome is a binary variable for whether distance to wild fire points is smaller than median, and cost outcome is the indicator for 'Lodgepole Pine' (1.0, undesired) as opposed to 'Spruce-Fir' (0.0, desired).

Ponfare Data The Ponfare Dataset is a public dataset with a coupons website (**?**), the dataset contains session as row items where the instance contains customer, coupon browsed, a continuous discount percentage, whether or not purchase, and auxiliary information on customer and coupon such as geographical location. We pre-filter the sessions where customers are below the age of 45. Due to disproportion of positive and negative samples, we subsample 4.0% of sample of sessions that do not result in purchase. The eventual dataset is around 130,822 samples, we utilize discount level as the continuous treatment policy, and use the median of the level to segment out sessions into treatment and control groups, respectively.

Figure 1 shows results of causal models on US Census. The baseline R-learner on gain performs slightly better due to less cost impact. Duality R-learner still works reasonably well. Direct Ranking and Constrained Ranking outperforms Duality R-learner by *2.8%* and *21.2%*, respectively to AUCC *0.58* and *0.69*. Figure **??** shows results of causal models on Covtype datasets. The optimization problem on this dataset is easier as results on multiple models are better. Direct Ranking and Constrained Ranking out-performs Duality R-learner by *7.3%* and *17.9%*, with AUCC for Con-

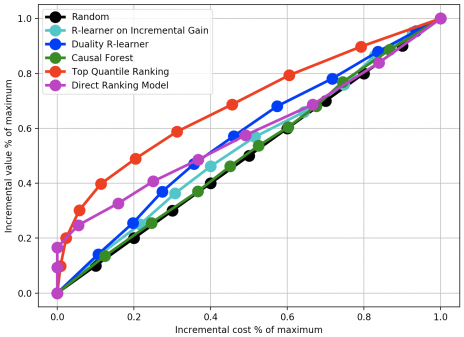

Figure 1. Cost-Curve results for public US Census data.

[3] Best parameters are the same for all three datasets we experimented: *num_trees*= 100 (50 trees for each of the two CATE function, τ_r, τ_c), *alpha*= 0.2, *min.node.size*= 3, *sample.fraction*= 0.5

Table 1. Summary of AUCC results across models and datasets.

Algorithm	USCensus	Covtype
Random	0.500	0.500
R-learner G	0.533	0.779
Duality R-learner	0.567	0.783
Causal Forest	0.510	0.832
Direct Ranking	**0.583**	**0.840**
Constrained Ranking	**0.687**	**0.915**

strained Ranking algorithm as high as *0.92*.

1 shows results algorithms are significantly better, more than *25%* in terms of AUCC, than R-learner on gain outcome, and out-performs Duality R-Learner by around *10%*. One example for cost effectiveness is to look at the vertical dash line at 20% of total incremental cost, we can achieve 2X more incremental retention than random selection by using our causal models. This can result in 50% reduction in cost.

References

Grf: Generalized random forests. https://grf-labs.github.io/grf/. Accessed: 2019-11-15.

Us census 1990 dataset on uci machine learning repository. https://archive.ics.uci.edu/ml/datasets/US+Census+Data+(1990). Accessed: 2019-11-15.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM, 2016.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. Learning deep structured semantic models for web search using clickthrough data. ACM International Conference on Information and Knowledge Management (CIKM), October 2013.

Kool, W., Gershman, S. J., and Cushman, F. A. Cost-benefit arbitration between multiple reinforcement-learning systems. *Psychological science*, 28(9):1321–1333, 2017.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. Metalearners for estimating heterogeneous treatment effects using machine learning. *arXiv preprint arXiv:1706.03461*, 2017.

Lim, B. Forecasting treatment responses over time using recurrent marginal structural networks. In *Advances in Neural Information Processing Systems*, pp. 7483–7493, 2018.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

Lunceford, J. and Davidian, M. Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. 2004.

Nie, X. and Wager, S. Quasi-oracle estimation of heterogeneous treatment effects, 2017.

Parbhoo, S., Wieser, M., and Roth, V. Causal deep information bottleneck. *arXiv preprint arXiv:1807.02326*, 2018.

Powers, S., Qian, J., Jung, K., Schuler, A., Shah, N. H., Hastie, T., and Tibshirani, R. Some methods for heterogeneous treatment effect estimation in high-dimensions. *arXiv preprint arXiv:1707.00102*, 2017.

R.S., S. and A.G., B. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Rubin, D. B. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Rzepakowski, P. and Jaroszewicz, S. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327, 2012.

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3076–3085. JMLR. org, 2017.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016. URL http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.

Wager, S. and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, (just-accepted), 2017.

Wager, S. and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.