Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

A DISTRIBUTIONALLY ROBUST BOOSTING ALGORITHM

Jose Blanchet
Fan Zhang

Department of Management Science and Engineering
Stanford University
475 Via Ortega, Suite 310
Stanford, CA. 94305, USA

Yang Kang

Department of Statistics
Columbia University
1255th Amsterdam Ave. RM1005
New York, NY, 10027, USA

Zhangyi Hu

Department of Earth and Environmental Sciences
University of Michigan
428 Church St
Ann Arbor, MI 48109, USA

ABSTRACT

Distributionally Robust Optimization (DRO) has been shown to provide a flexible framework for decision making under uncertainty and statistical estimation. For example, recent works in DRO have shown that popular statistical estimators can be interpreted as the solutions of suitable formulated data-driven DRO problems. In turn, this connection is used to optimally select tuning parameters in terms of a principled approach informed by robustness considerations. This paper contributes to this growing literature, connecting DRO and statistics, by showing how boosting algorithms can be studied via DRO. We propose a boosting type algorithm, named DRO-Boosting, as a procedure to solve our DRO formulation. Our DRO-Boosting algorithm recovers Adaptive Boosting (AdaBoost) in particular, thus showing that AdaBoost is effectively solving a DRO problem. We apply our algorithm to a financial dataset on credit card default payment prediction. Our approach compares favorably to alternative boosting methods which are widely used in practice.

1 INTRODUCTION

Distributional robustness in decision making under uncertainty is an important topic with a long and successful history in the Operations Research literature (Ben-Tal and Nemirovski 2000; Ghaoui et al. 2003; Bertsimas and Sim 2004; Erdoğan and Iyengar 2006). In recent years, this topic has been further fueled by distributionally robust optimization (DRO) formulations applied to machine learning and statistical analysis. These formulations have been shown to produce powerful insights in terms of interpretability, parameter tuning, implementation of computational procedures and the ability to enforce performance guarantees.

For instance, the work of (Lam and Zhou 2017) studied connections between the statistical theory of Empirical Likelihood and distributionally robust decision making formulations. These connections have been helpful in order to obtain statistical guarantees in a large and important class of data-driven DRO problems.

The work of (Esfahani and Kuhn 2018) showed how DRO can be used to establish a connection to regularized logistic regression, therefore explaining the role of regularization in the context of improving out-of-sample performance and data-driven decision-making.

The works of (Blanchet et al. 2016; Blanchet and Kang 2017a) further show that square-root Lasso, support vector machines, and other estimators such as group-Lasso can be recovered exactly from DRO formulations. In turn, in (Blanchet et al. 2016), it is shown that exploiting the DRO formulation leads to an optimality criterion for choosing the associated regularization parameter in regularized estimators which is both by robustness and statistical principles. Importantly, such criterion is closely aligned with a well-developed theory of high-dimensional statistics, but the criterion can be used even in the context of non-linear decision making problems.

The work of (Duchi et al. 2016) shows that variance regularization estimation can be cast in terms of a DRO formulation which renders the implementation of such estimator both practical and amenable to rigorous statistical and computational guarantees.

The list of recent papers that exploit DRO for the design of statistical estimators, revisiting classical ideas to improve parameter tuning, interpretation or implementation task is rapidly growing (Blanchet et al. 2016; Blanchet and Kang 2017a; Blanchet and Kang 2017b; Blanchet et al. 2017; Blanchet et al. 2017; Esfahani and Kuhn 2018; Shafieezadeh-Abadeh et al. 2015; Gao and Kleywegt 2016; Duchi et al. 2016).

This paper contributes to these rapidly-expanding research activities by showing that DRO can naturally be used in the context of boosting statistical procedures. In particular, we are able to show that Adaptive Boosting (AdaBoost) can be viewed as a particular example of DRO-Boosting with a suitable loss function and size of uncertainty, see Corollary 5.

These connections, as indicated earlier, are useful to enhance interpretability. Further, as we explain, DRO provides a systematic and disciplined approach for designing boosting methods. Also, the DRO formulation can be naturally used to fit tuning parameters (such as the strength used to weight one model versus another in the face of statistical evidence) in a statistically principled way.

Finally, we provide easy-to-implement algorithms which can be used to apply our boosting methods, which we refer to as DRO-Boosting procedures.

As an application of our proposed family of procedures, we consider a classification problem in the context of a credit card default payment prediction problem. We find that our approach compares favorably to alternative boosting methods which are widely used in practice.

The rest of this paper is organized as follows. In Section 2 we provide a general discussion about boosting algorithms. This section also introduces various notations which are useful to formulate our problem. Section 3 contains a precise description of the algorithm and the results validating the correctness of the algorithms. We discuss in this section also a connection to AdaBoost. Section 4 contains statistical analysis for the optimal selection of tuning parameters (such as the weight assigned to different models in the presence of available evidence). In Section 5, we show the result of our algorithms in the context of an application to credit card default payments. In Section 6, we summarize the proofs of technical results.

2 GENERAL BACKGROUND AND PROBLEM FORMULATION

We first start by describing a wide class of boosting algorithms. For concreteness, the reader may focus on a standard supervised classification framework. The ultimate goal is to predict the label Y associated with a predictive variable X. We then discuss our DRO problem formulation.

2.1 General Notions for Boosting

Suppose that at our disposal we have a set $\mathscr{F}$ of different classifiers. Typically, $\mathscr{F}$ will contain finitely many elements, but the DRO-Boosting algorithm can be implemented, in principle, in the case of an infinite dimensional $\mathscr{F}$, assuming a suitable Hilbert-space structure, as we shall discuss once we present our algorithms. We will keep this in mind, but to simplify the exposition, let us assume that $\mathscr{F}$ contains

finitely many elements, say $\mathscr{F} = \{f_1, ..., f_T\}$. However, we will point out the ways in which the infinite dimensional case can be considered.

The different elements in $\mathscr{F}$ are called classifiers, also known as predicting functions or "learners". (Throughout our discussion we shall use learner to refer to a given predicting function or a classifier.) For simplicity, let us assume that the class of learners $\mathscr{F}$ has been trained with independent data sets.

So, based on the learner $f_r \in \mathscr{F}$, we may have a decision rule to decide on the predicted label given the observation or predictor $X = x$. For example, such rule may by simply take the form $\widehat{y} = \text{sgn}(f_r(x))$ (where $\text{sgn}(b) = I(b \geq 0) - I(b < 0)$) or, instead, the classifier may estimate the probability that the label is 1 based on a probabilistic model which takes as input $f_r(x)$, in the context of logistic regression, which attaches probability proportional to $\exp(f_r(x))/(1+\exp(f_r(x))$ to the label being 1 and $1/(1+\exp(f_r(x))$ to the label being -1.

The idea of boosting is to combine the power of the learners in $\mathscr{F}$ to create a stronger learner. (Schapire 1990) showed that the idea of combining relatively weak predictors or classifiers into a strong one having desirable probably approximately correct (PAC) guarantees (Valiant 1984) is feasible. Due to these appealing theoretical guarantees, boosting algorithms have attracted substantial attention in the community, first by considering static learning algorithms (Schapire 1990; Freund 1995), and, inspired by Littlestone and Warmuth's reweighting algorithm, by considering adaptive algorithms, such as AdaBoost, which will be reviewed in the sequel. AdaBoost was proposed by (Freund and Schapire 1997) as an ensemble learning approach to classification problems, which adaptive combine multiple weak learners.

In the context of classification problems, many boosting algorithms can be viewed as a weighted majority vote of all the weak learners (where the weights are computed relative to the information carried by each of the weak learners). However, boosting type algorithms are not only applicable to the classification problems, they can also be used to combine the prediction of simple regressors to improve the prediction power in regression of non-categorical data, see for example (Drucker 1997).

We will now focus our discussion on finding boosting learners of the form $F \in \text{Lin}(\mathscr{F})$, where $\text{Lin}(\mathscr{F})$ is the linear span generated by the class $\mathscr{F}$. In particular, if $\mathscr{F} = \{f_1, ..., f_T\}$, then $\text{Lin}(\mathscr{F}) = \left\{F : F = \sum_{t=1}^{T} \alpha_t f_t \text{ for } \alpha_t \in (-\infty, \infty),\ 1 \leq t \leq T\right\}$. More generally, we may assume that $\mathscr{F}$ contains elements in a subspace of a Hilbert space and then $\text{Lin}(\mathscr{F})$ is the closure (under the Hilbert norm) of finitely many linear combinations of the elements in $\mathscr{F}$.

As a consequence of our results, we will show that a suitable DRO formulation applied to the family $\text{Lin}(\mathscr{F})$ of boosting learners can be used to systematically produce adaptive algorithms which can be connected to well-known procedures such as AdaBoost.

Now, let us assume that we are given a data set $\mathscr{D}_N = \{(X_i, Y_i)\}_{i=1,...,N}$. The variable X_i corresponds to the predictor and the outcome Y_i corresponds to the classification output or the label. As indicated earlier, let us assume that $Y_i \in \{-1, 1\}$ for concreteness. We use P_N to denote the empirical measure associated with the data set $\mathscr{D}_N$. In particular, $P_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{(X_i,Y_i)}(dx, dy)$, where $\delta_{(X_i,Y_i)}(dx, dy)$ is a point measure (or delta measure) centred at (X_i, Y_i). We use $\mathbb{E}_{P_N}[\cdot]$ to denote the expectation operator associated to the measure P_N. So, in particular, for example, for any continuous and bounded $g(\cdot)$, we have that $\mathbb{E}_{P_N}[g(X,Y)] = \frac{1}{N}\sum_{i=1}^{N} g(X_i, Y_i)$.

We introduce a loss function which can be used to measure the quality of our learner, namely, $L(F(X), Y)$. Let us now discuss examples of loss functions of interest.

Given a learner $F(\cdot)$, and an observation (X,Y), the margin corresponding to the observation is defined as $Y \cdot F(X)$. A loss function $L(F(X),Y)$ is said to be a *margin-type-cost-function* if there exist a function $\phi(\cdot)$ such that $L(F(X),Y) = \phi(Y \cdot F(X))$. In the context of regression problems, a popular loss function is the squared loss, namely, $L(F(X),Y) = (Y - F(X))^2$.

A natural approach to search for a boosting learner is to solve the optimization problem

$$\min_{F \in \text{Lin}(\mathscr{F})} \mathbb{E}_{P_N}[L(F(X), Y)]. \tag{1}$$

The choice of the function ϕ to define the loss function is relevant to endow the boosting procedure with robustness characteristics (see, for example, (Freund 2009; Rosset 2005)). While we focus on robustness as well, our approach is different. We take the loss function as a given modeling input and, as we shall explain, robustify the boosting training procedure by quantifying the impact of distributional perturbations in the distribution P_N. Our procedure could be applied in combination of the choice of a given function ϕ which is chosen to mitigate outliers, for example, as it is the case typically in standard robustness studies in statistics (Huber 2011). This combination may introduce a double layer of robustification and it may be an interesting topic of future research, but we do not pursue it here.

Viewing the search of a best linear combination of functions from a family of weaker learners to minimize the training loss, as in (1), can be understood in the vein of functional gradient descent (Friedman 2001). For example, the AdaBoost algorithm is an instance of the functional gradient descent algorithm with margin cost function $\phi(Y \cdot F(X)) := \exp(-Y \cdot F(X))$.

The connection between functional gradient descent and AdaBoost will be discussed in Section 3.

2.2 DRO-Boosting Formulation

The DRO formulation associated to (1) takes the form

$$\min_{F \in \mathrm{Lin}(\mathscr{F})} \max_{P \in \mathscr{U}_\delta(P_N)} \mathbb{E}_P \left[L\left(F\left(X\right), Y\right) \right]. \tag{2}$$

where $\mathscr{U}_\delta\left(P_N\right)$ is the distributional uncertainty set, centred around P_N and the size of uncertainty is $\delta > 0$.

The motivation for the introduction of the inner maximization relative to (1) is to mitigate potential overfitting problems inherent to the training problems using directly empirical risk minimization (i.e. formulation (1)). The data-driven DRO approach (2) tries to address overfitting by alleviating the focus solely on the observed evidence. Rather than empirical risk minimization, our DRO procedure is finding the optimal decision F that performs uniformly well around the empirical measure, where the uniform performance is evaluated within the distributional uncertainty set.

The data-driven DRO framework has been shown to be a valid procedure to improve the generalization performance for many machine learning algorithms. In the context of linear models, some of the popular machine learning algorithms with good generalization performance, for instance, regularized logistic regression, support vector machine (SVM), square-root LASSO, group LASSO, among others, can be recovered exactly in terms of a DRO formulation such as (1) (Blanchet and Kang 2017b; Esfahani and Kuhn 2018; Blanchet and Kang 2017a; Blanchet et al. 2017). In these applications, the set $\mathscr{U}_\delta\left(P_N\right)$ is defined in terms of the Wasserstein distance (Blanchet and Murthy 2019; Esfahani and Kuhn 2018; Gao and Kleywegt 2016; Zhao and Guan 2018).

Other application areas of data-driven DRO which have shown promise because of good empirical performance and theoretical guarantees include reinforcement learning, graphic modeling, deep learning, etc, have been shown to perform well empirically (Sinha et al. 2017; Fathony et al. 2018; Smirnova et al. 2019).

Finally, we also mention formulations of the distributional uncertainty set based on moment constraints, for instance (Delage and Ye 2010; Goh and Sim 2010).

Our focus on this paper is on the use of the Kullback-Leibler (KL) divergence in order to describe the set $\mathscr{U}_\delta\left(P_N\right)$. We choose to use the KL divergence because we want to establish the connection to well-know boosting algorithms, and we wish to further add their interpretability in terms of robustness. But we emphasize, as mentioned earlier, that a wide range of DRO formulations can be applied. Formulations of problems such as (1) based on the KL and related divergence notions have been studied in the recent years (Namkoong and Duchi 2016; Shapiro 2017; Bayraksan and Love 2015; Ghosh and Lam 2019; Lam 2016; Lam and Zhou 2017).

Let $\mathscr{P}(\mathscr{D}_N)$ denote the set of all probability distributions with support on $\mathscr{D}_N$. (Any element $P \in \mathscr{P}(\mathscr{D}_N)$ is a random measure because $\mathscr{D}_N$ is a random set, but this issue is not relevant to implement our DRO-

Boosting algorithm, we just consider $\mathscr{D}_N$ as given. For statistical guarantees, this issue is relevant, and it will be dealt with in the sequel.)

For any distribution $P \in \mathscr{P}(\mathscr{D}_N)$, we define the weight vector $\boldsymbol{\omega}_P = (\omega_{P,1}, \ldots, \omega_{P,N})$ such that $\omega_{P,i} = P((X,Y) = (X_i, Y_i))$ for $i = \overline{1,N}$. In particular, for the empirical distribution P_N we have $\boldsymbol{\omega}_{P_N} = (1/N, \ldots, 1/N)$. Note that the set $\mathscr{P}(\mathscr{D}_N)$ is isomorphic to the standard simplex $C_N = \left\{ \boldsymbol{\omega} = (\omega_1, \ldots, \omega_N) \mid \omega_i \geq 0, \sum_{i=1}^N \omega_i = 1 \right\}$.

For any distribution $P \in \mathscr{P}(\mathscr{D}_N)$, the KL divergence between P and P_N, denoted by $D(P\|P_N)$ (also known as the Relative Entropy or the Empirical Likelihood Ratio (ELR)) is defined as

$$D(P\|P_N) = -\frac{1}{N}\sum_{i=1}^{N} \log(N\omega_{P,i}).$$

It is a well-known consequence of Jensen's inequality that $D(P\|P_N) \geq 0$, and $D(P\|P_N) = 0$ if and only if $\boldsymbol{\omega}_P = \boldsymbol{\omega}_{P_N}$. The distributional uncertainty set that we consider is defined via

$$\mathscr{U}_\delta(P_N) = \{P \in \mathscr{P}(\mathscr{D}_N) | D(P\|P_N) \leq \delta\}.$$

By substituting the definition of the uncertainty set $\mathscr{U}_\delta(P_N)$ into (2), the DRO-Boosting model based on KL uncertainty sets is well defined given the loss function, the class $\mathscr{F}$, and the choice of δ, which will be discussed momentarily.

3 MAIN RESULTS

In this section, we propose an algorithm based on functional gradient descent to solve the DRO-Boosting (2). As we shall see, similar to the AdaBoost algorithm as in (Friedman 2001), fitting the DRO-Boosting algorithm exhibits a procedure that alternates between reweighting the worst case probability distribution and updating the predicting function $F(\cdot)$. Therefore, the connection between DRO-Boosting and AdaBoost will be naturally established. Moreover, since choosing different types of distributional uncertainty sets, $\mathscr{U}_\delta(P_N)$, potentially based on different notions of discrepancies between P_N and alternative distributions results in different reweighting regimes, it can be speculated that our proposed DRO-Boosting framework is more flexible than AdaBoost.

For ease of notation and introduction of our functional gradient descent method, in the rest of this paper, we define two functionals. For any index $i = \overline{1,N}$, the *empirical loss functional* $\mathscr{L}_i : \mathrm{Lin}(\mathscr{F}) \to \mathbb{R}$ is defined as $\mathscr{L}_i(F) = L(F(X_i), Y_i)$. The *robust loss functional* $\mathscr{L}_{rob} : \mathrm{Lin}(\mathscr{F}) \to \mathbb{R}$ is defined as $\mathscr{L}_{rob}(F) = \max_{P \in \mathscr{U}(P_N)} \mathbb{E}_P[L(F(X), Y)]$.

Due to the fact that any finitely supported distribution, P, is characterized by its associated weights, $\boldsymbol{\omega}_P$, we can introduce the *weight uncertainty set* $\Omega_\delta(P_N)$ defined as

$$\Omega_\delta(P_N) = \{\boldsymbol{\omega}_P | P \in \mathscr{U}_\delta(P_N)\} = \left\{ \boldsymbol{\omega} \in C_N \middle| -\frac{1}{N}\sum_{i=1}^{N} \log(N\omega_i) \leq \delta \right\},$$

as an isomorphic counterpart of the distributional uncertainty set $\mathscr{U}_\delta(P_N)$.

Thus, using the right hand side weight vector $\boldsymbol{\omega}_P$ and the weight uncertainty set $\Omega_\delta(P_N)$, the functional $\mathscr{L}_{rob}$ can be rewritten as

$$\mathscr{L}_{rob}(F) = \max_{\boldsymbol{\omega} \in \Omega_\delta(P_N)} \sum_{i=1}^{N} \omega_i \cdot \mathscr{L}_i(F). \tag{3}$$

We denote by $\Omega^\star_{rob}(F)$ the set of all maximizers to the above optimization problem, i.e.

$$\Omega^\star_{rob}(F) = \left\{ \boldsymbol{\omega} \in \Omega_\delta(P_N) \middle| \mathscr{L}_{rob}(F) = \frac{1}{N}\sum_{i=1}^{N} \omega_i \cdot \mathscr{L}_i(F) \right\}.$$

Using these definitions, the DRO-Boosting formulation (2) admits an alternative expression, namely,

$$\min_{F\in \mathrm{Lin}(\mathscr{F})} \mathscr{L}_{rob}(F), \tag{4}$$

consequently it suffices to find a best predicting function F that minimizes $\mathscr{L}_{rob}(F)$. In order to guarantee that the optimization problem (4) has optimal solution, we impose the following assumption.

Assumption 1 (Convex Loss Functional) The empirical loss functional $\mathscr{L}_i$ is assumed to be convex and continuous. In addition, there exist $\alpha \in \mathbb{R}$, such that the level set $\{F \in \mathrm{Lin}(\mathscr{F}) \mid \mathscr{L}_i(F) \leq \alpha\}$ is compact.

Next, to simplify the exposition, we impose an assumption which guarantees that the data is rich enough relative to the class $\mathrm{Lin}(\mathscr{F})$. We will discuss how can one proceed if this assumption is violated.

Assumption 2 (Separation) For any two different predicting functions $F, G \in \mathrm{Lin}(\mathscr{F})$, there exist at least one $(X_i, Y_i) \in \mathscr{D}_N$ such that $F(X_i) \neq G(X_i)$. In other words, the observed data $\mathscr{D}_N$ is rich enough to distinguish or separate two different learners in $\mathrm{Lin}(\mathscr{F})$.

Observation 1: There are natural situations in which the Separation Assumption may fail to hold. For example, if $f_r \in \mathscr{F}$ is a regression tree, namely, $f_t(x) = \sum_{j=1}^{m_t} I_{A_j}(x)$, where the A_j's are disjoint sets forming a partition on the domain of x, then when considering $\mathrm{Lin}(\mathscr{F})$ we may violate the Separation Assumption may not hold. Nevertheless, we can define an equivalence relationship "$\sim$" defined via $F \sim G$ if and only if $F(X) = G(X)$ for all $X \in \mathscr{D}_N$, and then work with equivalence classes instead. Moreover, since the dimension of the quotient space is at most N, we may assume that $\mathrm{Lin}(\mathscr{F})$ is a finite dimensional space without loss of generality. To ease the exposition, we will state the assumption $\mathrm{Lin}(\mathscr{F})$ is finite dimensional next.

Assumption 3 (Finite Representation) We assume that a finite dimensional basis exists for $\mathrm{Lin}(\mathscr{F})$ (or has been extracted for $(\mathrm{Lin}(\mathscr{F})/\sim)$). So, we can write $\mathrm{Lin}(\mathscr{F}) = \mathrm{Lin}\{f_1, ..., f_T\}$ (or $(\mathrm{Lin}(\mathscr{F})/\sim) = \mathrm{Lin}\{f_1, ..., f_T\}$) where $\{f_1, f_2, ..., f_T\}$ form linearly independent functions .

If the dimension of $\mathrm{Lin}(\mathscr{F})$ is infinite, then T will grow with N and this will have consequences for the rate of convergence of the algorithm and the statistical analysis of the uncertainty set selection. But this is not important to implement and run the algorithms that we will present below.

We now are ready to summarize some properties of $\mathscr{L}_{rob}$ in Lemma 1 and thereafter develop a subgradient descent algorithm to find the optimal robust decision rule by solving (2).

Lemma 1 If Assumption 1 is imposed, then the robust loss functional $\mathscr{L}_{rob} : \mathrm{Lin}(\mathscr{F}) \to \mathbb{R}$ is convex. In addition, the set of optimizer $\Omega^{\star}_{rob}(F)$ is guaranteed to be nonempty.

The predicting function controls the value of $\mathscr{L}_{rob}(F)$ solely via $F(X_i)$. Thus, to derive the functional gradient descent algorithm, one needs to construct a metric on $\mathscr{F}$ where the structure is determined only by $F(X_i)$ as well. To this end, we consider the inner product $\langle \cdot, \cdot \rangle_{\mathscr{D}_N}$, where

$$\langle F, G \rangle_{\mathscr{D}_N} = \frac{1}{N} \sum_{i=1}^{N} F(X_i) \cdot G(X_i), \qquad \forall F, G \in \mathrm{Lin}(\mathscr{F}).$$

Due to the Assumption 2, $\langle \cdot, \cdot \rangle_{\mathscr{D}_N}$ is a well defined inner product on space $\mathrm{Lin}(\mathscr{F})$, and we denote by $\| \cdot \|_{\mathscr{D}_N}$ the induced norm of $\langle \cdot, \cdot \rangle_{\mathscr{D}_N}$ on $\mathrm{Lin}(\mathscr{F})$. As the dimension of space $\mathrm{Lin}(\mathscr{F})$ can be assumed to be finite due to Observation 1, the space $\mathrm{Lin}(\mathscr{F})$ endowed with inner product $\langle \cdot, \cdot \rangle_{\mathscr{D}_N}$ is a Hilbert space. Note that the topology induced by $\langle \cdot, \cdot \rangle_{\mathscr{D}_N}$ is isomorphic to the Euclidean topology in $\mathbb{R}^N$.

In order to formalize the functional subgradient algorithm, we first introduce the definition of functional subgradient and *functional sub-differential*.

Definition 1 (Functional Sub-Gradient) For a convex functional $\mathscr{L} : \mathrm{Lin}(\mathscr{F}) \to \mathbb{R}$ and $F_0 \in \mathrm{Lin}(\mathscr{F})$, a linear functional $\mathscr{G} : \mathrm{Lin}(\mathscr{F}) \to \mathbb{R}$, (i.e. $\mathscr{G} \in \mathrm{Lin}(\mathscr{F})^*$) is called the *functional subgradient* of $\mathscr{L}$ at F_0 if

$$\mathscr{L}(F) - \mathscr{L}(F_0) \geq \mathscr{G}(F - F_0), \qquad \forall F \in \mathrm{Lin}(\mathscr{F}).$$

The *functional sub-differential* of $\mathscr{L}$ at F_0, denoted by $\partial\mathscr{L}(F_0)$, is defined as

$$\partial\mathscr{L}(F_0) = \{\mathscr{G} \mid \mathscr{G} \text{ is a functional subgradient of } \mathscr{L} \text{ at } F_0\}.$$

If the functional sub-differential set $\partial\mathscr{L}(F_0)$ is a singleton set, then the functional $\mathscr{L}$ is said to be differentiable at F_0, in which case the only element in $\partial\mathscr{L}(F_0)$ is denoted by $\nabla\mathscr{L}(F_0)$, called the *functional gradient* of $\mathscr{L}$ at F_0. Intuitively, one may regard the functional gradient $\nabla\mathscr{L}(F_0)$ as the response of an infinitesimal change in the predicting function F_0.

Proposition 2 Suppose that Assumptions 1-3 are imposed, then the functional sub-differential $\partial\mathscr{L}_{rob}(F)$ is given by

$$\partial\mathscr{L}_{rob}(F) = \mathbf{Co} \bigcup_{\boldsymbol{\omega}\in\Omega^\star_{rob}(F)} (\omega_1 \cdot \partial\mathscr{L}_1(F) + \cdots + \omega_N \cdot \partial\mathscr{L}_N(F)) \tag{5}$$

Here, for sets A and B, $\mathbf{Co}\,A$ denotes the convex hull of set A, and $A+B := \{a+b \mid a\in A, b\in B\}$ denotes the Minkowski sum of sets A and B.

Corollary 3 Suppose that Assumptions 1-2 are imposed, then if for some functional F we have $\Omega^\star_{rob}(F) = \{\boldsymbol{\omega}^\star(F)\}$ and $\partial\mathscr{L}_i(F) = \{\nabla\mathscr{L}_i(F)\}$ for each i, then $\mathscr{L}_{rob}$ is also differentiable at F and

$$\nabla\mathscr{L}_{rob}(F) = \omega_1^\star(F)\cdot\nabla\mathscr{L}_1(F) + \cdots + \omega_N^\star(F)\cdot\nabla\mathscr{L}_N(F).$$

Using Proposition 2 and Corollary 3, we can make sense of the functional sub-differential $\partial\mathscr{L}_{rob}(F)$ or even the functional gradient $\nabla\mathscr{L}_{rob}(F)$. However, in order to ensure the trajectory of functional gradient descent lies in the space $\mathrm{Lin}(\mathscr{F})$, one wants to find a weak linear in $\mathscr{F}$ to approximate the functional gradient $\mathscr{G}$. This simply means finding a best approximation in $\Pi_{\mathscr{F}}(\mathscr{G})$ such that $\Pi_{\mathscr{F}}(\mathscr{G}) = \arg\min_{F\in\mathscr{F}} \|F-\mathscr{G}\|_{\mathscr{D}_N}$.

Using these observation, the functional subgradient descent algorithm for solving the DRO-Boosting problem (2) is given in Algorithm 1.

Note that in Algorithm 1, we have to compute a worst case probability weight $\boldsymbol{\omega}^\star \in \Omega^\star_{rob}(F)$ for some functional F. If $\mathscr{L}_1(F) = \cdots = \mathscr{L}_N(F)$, we can simply pick $\boldsymbol{\omega}^\star = \boldsymbol{\omega}_{P_N} = (1/N,\ldots,1/N)$. Otherwise, the worst case probability can be computed using the Lemma 4.

Lemma 4 For $F\in\mathrm{Lin}(\mathscr{F})$ and $\beta\in(-\infty, -\max_i\{\mathscr{L}_i(F)\})$, define

$$\Psi(\beta;F) = \frac{1}{n}\sum_{i=1}^n \log\left(\frac{-1}{\mathscr{L}_i(F)+\beta}\right) - \log\left(\frac{1}{n}\sum_{i=1}^n \frac{-1}{\mathscr{L}_i(F)+\beta}\right) + \delta. \tag{6}$$

Suppose that there exist i, j such that $\mathscr{L}_i(F)\neq\mathscr{L}_j(F)$, then $\Psi(\beta;F)$ is strictly increasing in β. Furthermore, if we set $\omega_i^\star = \frac{(\mathscr{L}_i(F)+\beta^\star)^{-1}}{\sum_{k=1}^n(\mathscr{L}_k(F)+\beta^\star)^{-1}}$, where $\beta^\star$ is the unique root of $\Psi(\beta;F)=0$, then $\boldsymbol{\omega}^\star\in\Omega^\star_{rob}(F)$.

Corollary 5 shows that Algorithm 1 exactly recovers the AdaBoost algorithm proposed in (Freund and Schapire 1997). The proof of Corollary 5 is elementary.

Corollary 5 (Connection to AdaBoost) Suppose that $\mathscr{L}_i(F) = -\exp(Y_i\cdot F(X_i))$ and

$$\delta = \log\left(\frac{1}{n}\sum_{i=1}^n \exp\big(-Y_i\cdot F(X_i)\big)\right) - \frac{1}{n}\sum_{i=1}^n Y_i\cdot F(X_i),$$

then $\beta^\star = 0$ and the worst case probability in Lemma 4 is given by $\omega_i^\star = \frac{(\mathscr{L}_i(F)+\beta^\star)^{-1}}{\sum_{k=1}^n(\mathscr{L}_k(F)+\beta^\star)^{-1}} = \frac{\exp(-Y_i\cdot F(X_i))}{\sum_{k=1}^n\exp(-Y_k\cdot F(X_k))}$.

Remark 1 (Convergence Analysis) Under the technical assumption that $\mathscr{L}_{rob}$ has Lipschitz continuous gradient, the convergence of Algorithm 1 follows from Theorem 2 of (Mason et al. 1999). This assumption is typically violated in our setting. However, by introducing a soft maximum approximation to $\mathscr{L}_{rob}$, as in Theorem 2 of (Blanchet et al. 2017), one may apply the results of (Mason et al. 1999) directly, at the expense of a small (user-controlled) error. We tested empirically the convergence of the algorithm, successfully. The smoothing analysis will appear elsewhere.

Algorithm 1 DRO-Boosting

1: **Initialize** weight $\boldsymbol{\omega}^\star = (1/N, \ldots, 1/N)$ and $F_0(x) = 0$, maximum number of iterations T_{max}.
2: **for** $t := 0$ to T_{max} **do**
3: **Update** F_t: With fixed $\boldsymbol{\omega}^\star$, we compute a subgradient $\mathscr{G}_t \in \partial \mathscr{L}_{rob}(F_t)$ using Proposition 2.
4: Find a best approximation to $\mathscr{G}_t$ in class $\mathscr{F}$ and get the update direction $\Pi_{\mathscr{F}}(\mathscr{G}_t)$.
5: Apply line search for step size α_t; update the function as $F_{t+1} = F_t - \alpha_t \Pi_{\mathscr{F}}(\mathscr{G}_t)$.
6: **Update** $\boldsymbol{\omega}^\star$: With fixed F_{t+1}, we can evaluate the loss function for the observations as

$$\mathscr{L}_i(F_{t+1}) = L(F_{t+1}(X_i), Y_i).$$

7: Apply Newton-Raphson Method to compute the unique root of $\beta^\star$ of $\Psi(\beta; F_{t+1})$ and update the worst case probability $\boldsymbol{\omega}^\star$ according to Lemma 4.
8: We terminate the algorithm if there is no improvement could be make or we reach the maximal step size.
9: **Output** $F_{T_{max}}$.

Now, note that Algorithm 1 requires supplying the parameter δ. This parameter is typically chosen using cross-validation. However, when $\text{Lin}(\mathscr{F}) = \text{Lin}\{f_1, ..., f_T\}$, with T fixed, we can use the work of (Duchi et al. 2016; Lam and Zhou 2017), which establishes a connection to Empirical Likelihood to obtain δ. This will be discussed in Section 4.

4 OPTIMAL SELECTION OF THE DISTRIBUTIONAL UNCERTAINTY SIZE

We now explain how to choose δ using statistical principles and avoiding cross-validation. The strategy is to invoke the theory of Empirical Likelihood, following the approach in (Lam and Zhou 2017; Blanchet and Kang 2017b), as we shall explain. We assume that the data $\mathscr{D}_N = \{(X_i, Y_i)_{i=1}^N\}$ is i.i.d. from an underlying distribution P_*, which is unknown. We use P_*^∞ to denote the product measure $P_* \times P_* \times ...$ governing the distribution of the whole sequence $\{(X_n, Y_n) : n \geq 1\}$. Moreover, in this section we will assume, in Assumption 3, that $\text{Lin}(\mathscr{F}) = \text{Lin}\{f_1, ..., f_T\}$ where T is fixed and independent of N.

For each probability measure P, there is an associated optimal predicting function F as the minimizer to the risk minimization problem $F^P = \arg\min_{F \in \text{Lin}(\mathscr{F})} \mathbb{E}_P[L(F(X), Y)]$. We assume the convexity and smoothness for the loss function as discussed previously in Section 3. Following the criterion discussed in (Blanchet and Kang 2017b), we define $\Lambda_\delta(P_N) = \{F^P : P \in \mathscr{U}_\delta(P_N)\}$. The set $\Lambda_\delta(P_N)$ can be interpreted as a confidence region for the functional parameter F^{P_*} and we are interested in minimizing the size of the confidence region, δ, while guaranteeing a desired level of coverage for the confidence region $\Lambda_\delta(P_N)$. In other words, we want to choose δ as the solution to the problem

$$\min\{\delta : P_*^\infty\left(F^{P_*} \in \Lambda_\delta(P_N)\right) \geq 1 - \alpha\}, \tag{7}$$

where $1 - \alpha$ is a desired confidence level, say 95%, which implies choosing $\alpha = .05$. This problem is challenging to solve, but we will provide an asymptotic approximation to it as $N \to \infty$.

We make the following assumptions to proceed the discussion.

Assumption 4 (First Order Optimality Condition) The optimal choice F^P is characterized via the corresponding first-order optimality condition, which yields $\mathbb{E}_P\left[\nabla_F L\left(F^P(X), Y\right)\right] = 0$.

By the definition of the functional gradient in Definition 1, we know $\nabla_F L\left(F^P(X), Y\right) \in \mathbb{R}^T$.

The derivation of the asymptotic results relies on central limitation theorem, and we assume the existence of the second moment for the functional gradient.

Assumption 5 (Square Integrability) We assume that $\mathbb{E}_{P_*}\left[\left\|\nabla_F L\left(F^P(X), Y\right)\right\|_2^2\right] < \infty$.

In order to compute δ according to (7) we will provide a more convenient representation for the set $\{F^{P_*} \in \Lambda_\delta(P_N)\}$. For any F, we can define a set of probability measures $M(F)$ for which F is an optimal boosting learner, that is, $P \in M(F)$ if $\mathbb{E}_P[\nabla_F L(F(X),Y)] = 0$. Consequently, $M(F) = \{P \mid \mathbb{E}_P[\nabla_F L(F(X),Y)] = 0\}$. Define the smallest Kullback-Leibler discrepancy between P_N and any element of $M(F)$ via

$$R_N(F) = \min_P \{D(P\|P_N) \mid \mathbb{E}_P[\nabla_F L(F(X),Y)] = 0\}. \tag{8}$$

It is immediate to see that $F^{P_*} \in \Lambda_\delta(P_N)$ if and only if there exists an element in $P \in M\left(F^{P_*}\right)$ such that $D(P\|P_N) \leq \delta$. Consequently, we have

$$F^{P_*} \in \Lambda_\delta(P_N) \Longleftrightarrow R_N\left(F^{P_*}\right) \leq \delta. \tag{9}$$

The asymptotic analysis of the object $R_N\left(F^{P_*}\right)$, known as the Empirical Profile Likelihood (EPL) has been studied extensively. From (9) we have that the solution to (7) is precisely the $1-\alpha$ quantile of $R_N\left(F^{P_*}\right)$. Therefore, to approximate δ, it suffices to estimate the asymptotic distribution of $R_N\left(F^{P_*}\right)$.

We apply the techniques in (Lam and Zhou 2017) to derive the asymptotic distribution of $R_N(F)$ as $n \to \infty$, and pick the $1-\alpha$ quantile of the asymptotic distribution as our uncertainty size.

The asymptotic results for (8) is similar as introduced in the literature for the empirical likelihood theorem as in (Owen 1988; Qin and Lawless 1994; Owen 2001). There show that the asymptotic distribution is chi-square and the degree of freedom depends on the estimating equation. We state the results in Theorem 6 below.

Theorem 6 (Generalized Empirical Likelihood Theorem) We assume that our loss function L is smooth, the functional gradient is well defined, and Assumption 4 and Assumption 5 hold. Then we have the EPL function defined in (8) has asymptotic chi-square distribution, i.e.

$$2NR_N\left(F^{P_*}\right) \Rightarrow \chi^2_T, \tag{10}$$

where χ_T is the chi-square distribution with degree of freedom equal T and $\Rightarrow$ stands for weak convergence.

We can notice the rate of convergence is free of the data dimension and the dimension of the dimension of the functional space. Theorem 6 is a generalization of the empirical likelihood theorem for general estimating equation introduced in (Qin and Lawless 1994).

5 NUMERICAL EXPERIMENTS

In this section, we apply our DRO-Boosting algorithm to predict default in credit card payment. We take the credit-card default payment data of Taiwan from UCI machine learning database (Lichman, M. 2013). The data has 23 predictors and the response is binary stands for non-default and default. There are 30000 observations in the data set with 6636 defaults and 23364 non-defaults.

We take the binary classification tree as the weak learner as the basis function, and we consider exponential loss function for model fitting. We compare our model with the AdaBoost algorithm, which is the state-of-art boosting algorithm for practical consideration.

Every time, we randomly split the data into a training set, with 3000 observations, and a testing set, with the rest 27000 data points. We train the model on the training set as we illustrated in Algorithm 1. We consider the basis function as 5-layer classification tree, and we assume the dimension (or effective degree of freedom) of the functional space is roughly 30. To pick the uncertainty set, we apply the method introduced in Section 4, where we pick the level of uncertainty to be 90%. We consider a more complex basis model, a 5-layer classification tree, as the weak learner, which is mainly due to we try to explore a case where the AdaBoost model is more likely to overfit the data.

We report the accuracy, true positive rate, false negative rate, and the exponential loss in Table 1, where we can observe superior performance for the DRO-Boosting model on the testing set.

Table 1: Numerical results for applying DRO-Boosting to credit card default payment prediction.

Reporting Domain	Training Set		Testing Set	
Algorithm	AdaBoost	DRO-Boosting	AdaBoost	DRO-Boosting
Accuracy $P\left(Y_{true}=Y_{pred}\right)$	0.948 ± 0.001	0.840 ± 0.006	0.788 ± 0.004	0.819 ± 0.003
False Negative Rate $P\left(P_{pred}=1\middle\vert P_{true}=1\right)$	0.807 ± 0.039	0.406 ± 0.032	0.352 ± 0.021	0.354 ± 0.023
True Positive Rate $P\left(P_{pred}=-1\middle\vert P_{true}=-1\right)$	0.988 ± 0.005	0.963 ± 0.006	0.913 ± 0.009	0.946 ± 0.006
Average Exponential Loss	0.497 ± 0.041	0.738 ± 0.013	0.860 ± 0.009	0.801 ± 0.007

We can observe from our numerical experiment that the DRO formalization helps improve the performance on the testing set. The worst-case expected loss function tires to mimic the testing error and avoid overemphasis on the training set. Thus we observe higher training error using DRO, while the advantage is reflected in the testing error.

6 PROOFS OF THE TECHNICAL RESULTS

The proofs are presented in the order as they appear in the paper.

Proof of Lemma 1. For the first argument, each $\mathscr{L}_i$ is a convex functional due to Assumption 1, so is their convex combination $\sum_{i=1}^N \omega_i \cdot \mathscr{L}_i(F)$. After taking maximization over $\boldsymbol{\omega} \in \Omega_\delta(P_N)$, the functional $\mathscr{L}_{rob}(F)$ is still convex. For the second argument, the objective function on the right hand side of (3) is continuous in $\boldsymbol{\omega}$ and the feasible region $\Omega_\delta(P_N)$ is compact, so the optimizer set $\Omega^\star_{rob}(F)$ is guaranteed to be nonempty. □

Proof of Proposition 2. According to (Hiriart-Urruty and Lemaréchal 2012) Chapter D, Theorem 4.1.1, we have $\partial\left(\omega_1 \cdot \mathscr{L}_1(F)+\cdots+\omega_N \cdot \mathscr{L}_N(F)\right) = \omega_1 \cdot \partial\mathscr{L}_1(F)+\cdots+\omega_N \cdot \partial\mathscr{L}_N(F)$ for all $\boldsymbol{\omega} \in C_N$. In addition, note that each $\mathscr{L}_i(F)$ is convex and upper semi-continuous according to Assumption 1, so $\omega_1 \cdot \mathscr{L}_1(F)+\cdots+\omega_N \cdot \mathscr{L}_N(F)$, as a convex combination of $\mathscr{L}_i(F)$, is also convex and upper semi-continuous. Furthermore, the set $\Omega^\star_{rob}(F)$ is a compact set. Consequently, using (Hiriart-Urruty and Lemaréchal 2012), Chapter D, Theorem 4.4.2, the desired result (5) is proved. □

Proof of Corollary 3. The result follows from Theorem 2 and (Hiriart-Urruty and Lemaréchal 2012), Chapter D, Corollary 2.1.4. □

Proof of Lemma 4. The strict monotonicity of the fucntion $\Psi(\beta;F)$ can be shown by taking derivative and applying Cauchy Schwartz inequality. Using the fact that $\beta^\star$ is the root of $\Psi(\beta;F)$, it follows that $\boldsymbol{\omega}^\star \in \Omega_\delta(P_n)$. The optimality of $\boldsymbol{\omega}^\star$ is proved by verifying the Karush–Kuhn–Tucker conditions of the convex optimization problem $\max_{\boldsymbol{\omega}\in\Omega_\delta(P_N)} \sum_{i=1}^N \omega_i \cdot \mathscr{L}_i(F)$. □

Proof of Theorem 6. The EPL function (8) is a convex optimization with constraint. We can write the optimization problem in the Lagrange form as $\max_{\boldsymbol{\omega}\in C_N}\left\{\frac{1}{N}\sum_{i=1}^N N\omega_i + \lambda_N^T \nabla_F L(F(X_i),Y_i)\right\}$, where λ_N is the Lagrange multiplier. The optimization problem could be solved by its first order optimality condition, and it gives $\omega_i = \frac{1}{N}\frac{1}{1+\lambda_N^T\cdot\nabla_F L(F(X_i),Y_i)}$ and $\frac{1}{N}\sum_{i=1}^N \frac{\nabla_F L(F(X_i),Y_i)}{1+\lambda_N^T\nabla_F L(F(X_i),Y_i)} = 0$. We denote $M_i = \nabla_F L(F(X_i),Y_i)$, $W_i = \lambda_N^T M_i$, and $S_N = \frac{1}{N}\sum_{i=1}^N M_i \cdot M_i^T$. Then we apply Lemma 11.1 and Lemma 11.2 in (Owen 2001),

$$\lambda_N = S_N^{-1}\cdot\overline{M_N} + \zeta_N \text{ with } \overline{M_N} = \frac{\sum_{i=1}^N M_i}{N} \text{ and } \zeta_N = o_p\left(N^{-1/2}\right), \tag{11}$$

$$\log(1+W_i) = W_i - \frac{1}{2}W_i^2 + \eta_i \text{ with } \sum_i^N \eta_i = o_p(1). \tag{12}$$

Therefore, for the ELP function, we have

$$2NR_N\left(F^{P_*}\right) = -2\sum_{i=1}^{N}\log N\omega_i = 2\sum_{i=1}^{N}\log\left(1+W_i\right) = 2\sum_{i=1}^{N}W_i - \sum_{i=1}^{N}W_i^2 + 2\sum_{i=1}^{N}\eta_i$$

$$= N\overline{M_N}^T S_N^- 1\overline{M_N} - N\zeta_N S_N^- 1\zeta_N + 2\sum_{i=1}^{N}\eta_i = N\overline{M_N}^T S_N^- 1\overline{M_N} + o_p(1) \Rightarrow \chi_T^2$$

The first two equations are by definition, the equation four, five and six are applying (11) and (12), while the final equation is the central limitation theorem for the estimating equation. □

ACKNOWLEDGMENTS

We gratefully acknowledge support from the following NSF grants 1915967, 1820942, 1838676 as well as DARPA award N660011824028.

REFERENCES

Bayraksan, G., and D. K. Love. 2015. "Data-Driven Stochastic Programming Using Phi-Divergences". In *The Operations Research Revolution*, 1–19. Catonsville: Institute for Operations Research and the Management Sciences.

Ben-Tal, A., and A. Nemirovski. 2000. "Robust Solutions of Linear Programming Problems Contaminated with Uncertain Data". *Mathematical Programming* 88(3):411–424.

Bertsimas, D., and M. Sim. 2004. "The Price of Robustness". *Operations Research* 52(1):35–53.

Blanchet, J., and Y. Kang. 2017a. "Distributionally Robust Groupwise Regularization Estimator". *arXiv preprint arXiv:1705.04241*.

Blanchet, J., and Y. Kang. 2017b. "Distributionally Robust Semi-Supervised Learning". *arXiv preprint arXiv:1702.08848*.

Blanchet, J., Y. Kang, and K. Murthy. 2016. "Robust Wasserstein Profile Inference and Applications to Machine Learning". *arXiv preprint arXiv:1610.05627*.

Blanchet, J., Y. Kang, F. Zhang, F. He, and Z. Hu. 2017. "Doubly Robust Data-Driven Distributionally Robust Optimization". *arXiv preprint arXiv:1705.07168*.

Blanchet, J., Y. Kang, F. Zhang, and K. Murthy. 2017. "Data-Driven Optimal Transport Cost Selection for Distributionally Robust Optimization". *arXiv preprint arXiv:1705.07152*.

Blanchet, J., and K. Murthy. 2019. "Quantifying Distributional Model Risk via Optimal Transport". *Mathematics of Operations Research* 44(2):565–600.

Delage, E., and Y. Ye. 2010. "Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems". *Operations Research* 58(3):595–612.

Drucker, H. 1997. "Improving Regressors Using Boosting Techniques". In *Proceedings of the Fourteenth International Conference on Machine Learning*, 107–115. Burlington: Elsevier.

Duchi, J., P. Glynn, and H. Namkoong. 2016. "Statistics of Robust Optimization: A Generalized Empirical Likelihood Approach". *arXiv preprint arXiv:1610.03425*.

Erdoğan, E., and G. Iyengar. 2006. "Ambiguous Chance Constrained Problems and Robust Optimization". *Mathematical Programming* 107(1-2):37–61.

Esfahani, P. M., and D. Kuhn. 2018. "Data-Driven Distributionally Robust Optimization Using the Wasserstein Metric: Performance Guarantees and Tractable Reformulations". *Mathematical Programming* 171(1-2):115–166.

Fathony, R., A. Rezaei, M. A. Bashiri, X. Zhang, and B. D. Ziebart. 2018. "Distributionally Robust Graphical Models". In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 8354–8365. Red Hook: Curran Associates Inc.

Freund, Y. 1995. "Boosting a Weak Learning Algorithm by Majority". *Information and Computation* 121(2):256–285.

Freund, Y. 2009. "A More Robust Boosting Algorithm". *arXiv preprint arXiv:0905.2138*.

Freund, Y., and R. E. Schapire. 1997. "A Decision Theoretic Generalization of Online Learning and an Application to Boosting". *Journal of Computer and System Sciences* 55(1):119–139.

Friedman, J. H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine". *Annals of Statistics*:1189–1232.

Gao, R., and A. J. Kleywegt. 2016. "Distributionally Robust Stochastic Optimization with Wasserstein Distance". *arXiv preprint arXiv:1604.02199*.

Ghaoui, L. E., M. Oks, and F. Oustry. 2003. "Worst-Case Value-at-Risk and Robust Portfolio Optimization: A Conic Programming Approach". *Operations Research* 51(4):543–556.

Ghosh, S., and H. Lam. 2019. "Robust Analysis in Stochastic Simulation: Computation and Performance Guarantees". *Operations Research*.

Goh, J., and M. Sim. 2010. "Distributionally Robust Optimization and its Tractable Approximations". *Operations Research* 58(4-part-1):902–917.

Hiriart-Urruty, J.-B., and C. Lemaréchal. 2012. *Fundamentals of Convex Analysis*. Berlin: Springer.

Huber, P. J. 2011. *Robust Statistics*. Berlin: Springer.

Lam, H. 2016. "Recovering Best Statistical Guarantees via the Empirical Divergence-Based Distributionally Robust Optimization". *arXiv preprint arXiv:1605.09349*.

Lam, H., and E. Zhou. 2017. "The Empirical Likelihood Approach to Quantifying Uncertainty in Sample Average Approximation". *Operations Research Letters* 45(4):301–307.

Lichman, M. 2013. "UCI Machine Learning Repository". http://archive.ics.uci.edu/ml, accessed 3rd August.

Mason, L., J. Baxter, P. Bartlett, and M. Frean. 1999. "Boosting Algorithms as Gradient Descent". In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, 512–518. Cambridge: MIT Press.

Namkoong, H., and J. C. Duchi. 2016. "Stochastic Gradient Methods for Distributionally Robust Optimization with f-Divergences". In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2216–2224. Red Hook: Curran Associates Inc.

Owen, A. B. 1988. "Empirical Likelihood Ratio Confidence Intervals for a Single Functional". *Biometrika* 75(2):237–249.

Owen, A. B. 2001. *Empirical Likelihood*. New York: CRC Press.

Qin, J., and J. Lawless. 1994. "Empirical Likelihood and General Estimating Equations". *The Annals of Statistics*:300–325.

Rosset, S. 2005. "Robust Boosting and its Relation to Bagging". In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 249–255. New York: Association for Computing Machinery.

Schapire, R. E. 1990. "The Strength of Weak Learnability". *Machine Learning* 5(2):197–227.

Shafieezadeh-Abadeh, S., P. M. Esfahani, and D. Kuhn. 2015. "Distributionally Robust Logistic Regression". In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, 1576–1584. Cambridge: MIT Press.

Shapiro, A. 2017. "Distributionally Robust Stochastic Programming". *SIAM Journal on Optimization* 27(4):2258–2275.

Sinha, A., H. Namkoong, and J. Duchi. 2017. "Certifying Some Distributional Robustness with Principled Adversarial Training". *arXiv preprint arXiv:1710.10571*.

Smirnova, E., E. Dohmatob, and J. Mary. 2019. "Distributionally Robust Reinforcement Learning". *arXiv preprint arXiv:1902.08708*.

Valiant, L. G. 1984. "A Theory of the Learnable". In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, 436–445. New York: Association for Computing Machinery.

Zhao, C., and Y. Guan. 2018. "Data-Driven Risk-Averse Stochastic Optimization with Wasserstein Metric". *Operations Research Letters* 46(2):262–267.

AUTHOR BIOGRAPHIES

JOSE BLANCHET is a faculty member in the Department of Management Science and Engineering at Stanford University. Jose is a recipient of the 2009 Best Publication Award given by the INFORMS Applied Probability Society and of the 2010 Erlang Prize. He also received a PECASE award given by NSF in 2010. His email is jose.blanchet@stanford.edu.

YANG KANG got his Ph.D. in Statistics from Columbia University. He has research interests in applied probability and robust statistic inference. Now he is working in the financial industry. His email is yang.kang@columbia.edu. His website is https://sites.google.com/view/yangkang1026/.

FAN ZHANG is a Ph.D. candidate in the Department of Management Science and Engineering at Stanford University. His research interests include applied probability, stochastic simulation and robust optimization. His email is fzh@stanford.edu.

ZHANGYI HU is a software developer in a financial service company. He was a Ph.D. candidate in Geophysics at University of Michigan. He used to work as a quantitative developer at Bloomberg L.P. His email is hu.zhangyi@gmail.com.