Dropout Training is Distributionally Robust Optimal

José Blanchet JOSE.BLANCHET@STANFORD.EDU
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, USA

Yang Kang YANGKANG@STAT.COLUMBIA.EDU
Department of Statistics
Columbia University
New York, NY 10027, USA

José Luis Montiel Olea MONTIEL.OLEA@GMAIL.COM
Department of Economics
Columbia University
New York, NY 10027, USA

Viet Anh Nguyen VIET-ANH.NGUYEN@STANFORD.EDU
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, USA

Xuhui Zhang XUHUI.ZHANG@STANFORD.EDU
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, USA

Editor:

Abstract

This paper shows that dropout training in Generalized Linear Models is the minimax solution of a two-player, zero-sum game where an adversarial nature corrupts a statistician's covariates using a multiplicative nonparametric errors-in-variables model. In this game, nature's *least favorable distribution* is *dropout noise*, where nature independently deletes entries of the covariate vector with some fixed probability δ. This result implies that dropout training indeed provides out-of-sample expected loss guarantees for distributions that arise from multiplicative perturbations of in-sample data. In addition to the decision-theoretic analysis, the paper makes two more contributions. First, there is a concrete recommendation on how to select the tuning parameter δ to guarantee that, as the sample size grows large, the in-sample loss after dropout training exceeds the true population loss with some pre-specified probability. Second, the paper provides a novel, parallelizable, Unbiased Multi-Level Monte Carlo algorithm to speed-up the implementation of dropout training. Our algorithm has a much smaller computational cost compared to the naive implementation of dropout, provided the number of data points is much smaller than the dimension of the covariate vector.

Keywords: Generalized Linear Models, Distributionally Robust Optimization, Machine Learning, Minimax Theorem, Multi-Level Monte Carlo.

1. Introduction

Dropout training is an increasingly popular estimation method in machine learning.[1] The general idea consists in ignoring some dimensions of the covariate vector at random while estimating the parameters of a statistical model. A common motivation for dropout training is that the random feature selection implicitly performs *model averaging*, potentially improving out-of-sample prediction error and thus mitigating overfitting. See Hinton et al. (2012) for a discussion about this point in the context of neural networks. See also Draper (1994) and Raftery et al. (1997) for classical results on the optimality of model averaging for prediction purposes.

Our main goal is to contribute to the growing literature explaining the success of dropout training in mitigating overfitting; e.g., Wager et al. (2013), Helmbold and Long (2015), Wei et al. (2020). Our main result (Theorem 2) shows that dropping out input features when training Generalized Linear Models can be viewed as the minimax solution to an adversarial game known in the stochastic optimization literature (Shapiro et al. (2014)) as a Distributionally Robust Optimization (DRO) problem.

Broadly speaking, a DRO problem is a two-player, zero-sum game between a decision maker (a statistician) and an adversary (nature). The statistician wishes to choose an action to minimize a given expected loss (e.g., squared loss in a typical linear regression setting or, more generally, the negative of the log-likelihood function), while nature intends this loss to be maximal. We consider a framework in which nature is allowed to harm the statistician by corrupting the available data using a multiplicative nonparametric errors-in-variables model; as in the classical work of Hwang (1986). The statistician is aware of the data corruption and knows the distribution used by nature, but does not have access to the realizations of the corruption noise. Under mild assumptions, nature's *least favorable distribution* in this game is shown to be *dropout noise*, where nature independently deletes entries of the covariate vector with some fixed probability δ. The Minimax Theorem (Morgenstern and von Neumann, 1953) is shown to also hold for this game: the minimax value coincides with the maximin solution, and these coincide with the payoffs in the game's Nash equilibrium. One direct consequence is that the statistician's selected procedure in the face of multiplicative nonparametric noise maintains optimal performance even if the adversary is allowed to corrupt after the statistician uses the training data.

Our main result (Theorem 2) shows that, by construction, dropout training indeed provides out-of-sample performance guarantees for distributions that arise from multiplicative perturbations of in-sample data. More precisely, given any fixed sample size, the *out-of-sample expected loss* is no larger than that obtained by dropout training *in-sample*, provided

[1] Section 7.12 of Goodfellow et al. (2016) provides a textbook treatment on dropout training. Bishop (1995) and Srivastava et al. (2014) are seminal references on this topic.

we consider out-of-sample distributions obtained as multiplicative perturbations of the in-sample distribution. Therefore, our result formally qualifies the ability of dropout training to enhance out-of-sample performance, which is one of the reasons often invoked to use the dropout method. Moreover, our results show that for any parameter value the loss used in dropout training is larger than the negative log-likelihood of Generalized Linear Models.

We make two additional contributions. First, we suggest a novel procedure to select the dropout probability δ. To this end, we study how often the in-sample loss of dropout training exceeds the true (and unknown) population expected loss. When $\delta = 0$, the Central Limit Theorem implies this event occurs with approximately .5 probability. When δ is fixed, Theorem 2 implies this event occurs with probability 1. We show that picking δ to be of the form $c/\sqrt{n}$ (where n denotes the number of training examples) makes the in-sample loss of dropout training exceed the population loss with probability that depends on c. Consequently, by choosing a target probability, say 95%, it is possible to provide a concrete recommendation for the selection of c, and therefore, of δ.

Second, we suggest a new stochastic optimization implementation of dropout training. A well-known drawback of dropout is its computational cost. As we will explain, a d-dimensional covariate vector requires 2^d evaluations of the loss in order to integrate out the dropout noise for a particular data point. The computational cost is alleviated by implementing dropout training by using either Stochastic Gradient Descent (Robbins and Monro (1951)) or naive Monte-Carlo approximations to the expected loss, both of which require draws from the joint distribution of the data and dropout noise. Unfortunately, both of these approximations introduce bias to the solution of dropout training. Also, none of these procedures can exploit the increasing availability of parallel computing in order to alleviate their computational burden. We borrow ideas from the Multi-level Monte Carlo literature—in particular from the work of Blanchet et al. (2019a)—to suggest an unbiased (in a sense we will make precise) dropout training routine that is easily parallelizable and that has a much smaller computational cost compared to naive dropout training methods when the number of features is large (Theorem 4). Our algorithm thus complements the recent literature suggesting approaches to speed-up dropout training by either using a parallelized implementation of Stochastic Gradient Descent (Zinkevich et al., 2010) or a fast dropout training based on Gaussian approximations (Wang and Manning, 2013).

The rest of the paper is organized as follows. Section 2 explains dropout training in the context of Generalized Linear Models. Section 3 presents a general description of the DRO framework used in this paper. Section 4 specializes the DRO problem by using the negative log-likelihood of Generalized Linear Models to define a loss function for the statistician, and by allowing nature to harm the statistician via a multiplicative errors-in-variables model for the covariates. This section also presents our main theorem. Section 5 presents our approach to select the dropout probability, δ. Section 6 discusses different computational methods available for implementing dropout training (full integration, Stochastic Gradient Descent, Naive Monte Carlo integration) and presents our suggested *Unbiased Multi-level Monte Carlo* algorithm. Section 7 presents some simulations comparing our recommended selection of δ to cross-validation, as well as our preferred implementation of dropout training to Stochastic Gradient Descent. Finally, Section 8 discusses extensions of our results to a particular class of feed-forward neural networks with a single hidden layer. We show that dropout training of the hidden units in the hidden layer is distributionally robust optimal. All the proofs are collected in the Appendix.

2. Dropout Training in Generalized Linear Models

This section describes dropout training in the context of Generalized Linear Models. As some other recent papers in the literature, we view Generalized Linear Models as a convenient, transparent, and relevant framework to better understand the theoretical and algorithmic properties of dropout training.

2.1 Generalized Linear Models (GLMs)

A Generalized Linear Model—with parameters β and ϕ—is defined by a conditional density for the response variable $Y \in \mathcal{Y} \subseteq \mathbb{R}$ given $X \in \mathbb{R}^d$

$$f(Y|X,\beta,\phi) \equiv h(Y,\phi) \exp\left(\left(Y\beta^\top X - \Psi(\beta^\top X) \right) / a(\phi) \right), \tag{1}$$

see McCullagh and Nelder (1989, Equation 2.4). In our notation $h(\cdot,\phi)$ is a real-valued function (integrable with respect to the true data distribution), parameterized by ϕ defined on the domain $\mathcal{Y}$; $a(\cdot)$ is a positive function of ϕ; and $\Psi(\cdot)$ is the log-partition function, which we assume to be defined on all the real line. It is well-known that in GLMs with a scalar response variable the log-partition function is infinitely differentiable and strictly convex on its domain; see Proposition 3.1 in Wainwright and Jordan (2008).

Normal, Logistic, and Poisson Regression have conditional densities of the form (1). For the sake of exposition, we provide details below for linear and logistic regression.

Example 1 (Linear regression with unknown variance) *Consider the linear model* $Y = \beta^\top X + \varepsilon$*, in which* $\varepsilon \sim \mathcal{N}(0,\sigma^2)$ *with unknown variance* $\sigma^2 \in \mathbb{R}_{++}$ *and* $\varepsilon \perp X$*. The conditional distribution of* Y *given* X *satisfies* (1) *with* $\phi = \sigma^2$, $a(\phi) = \phi$, $\Psi(\beta^\top X) = (\beta^\top X)^2/2$ *and* $h(Y,\phi) = (2\pi\phi)^{-\frac{1}{2}} \exp(-Y^2/(2\phi))$.

Example 2 (Logistic regression) *Consider $Y|X \sim$ Bernoulli$(1/(1+\exp(-\beta^\top X))$ with $\mathcal{Y} = \{0,1\}$. The conditional probability mass function of Y given X satisfies* (1) *with $a(\phi) = 1$, $\Psi(\beta^\top X) = \log(1+\exp(\beta^\top X))$ and $h(Y,\phi) = 1$.*

Generalized Linear Models are typically estimated via Maximum Likelihood using (1). Given n i.i.d. data realizations or *training examples* (x_i, y_i), the Maximum Likelihood estimator $(\widehat{\beta}_{\mathrm{ML}}, \widehat{\phi}_{\mathrm{ML}})$ is defined as any solution of the problem

$$\min_{\beta,\phi} \sum_{i=1}^{n} -\ln f(y_i|x_i,\beta,\phi). \tag{2}$$

Maximum Likelihood estimators in GLMs are known to be consistent and asymptotically normal under mild regularity conditions on the joint distribution of (X_i, Y_i) (Fahrmeir and Kaufmann, 1985). We denote this distribution as $P^\star$.

2.2 Dropout Training

An alternative to standard Maximum Likelihood estimation in GLMs is dropout training. The general idea consists in ignoring some randomly chosen dimensions of x_i while training a statistical model.

For a given covariate vector x_i—and an user-selected constant, $\delta \in [0,1)$—define the d-dimensional random vector

$$\xi_i = (\xi_{i,1}, \ldots, \xi_{i,d})^\top \in \{0, 1/(1-\delta)\}^d,$$

where each of the d entries of ξ_i is an independent draw from a scaled Bernoulli distribution with parameter $1-\delta$. This is, for $j = 1, \ldots, d$:

$$\xi_{i,j} = \begin{cases} 0 & \text{with probability } \delta, \\ (1-\delta)^{-1} & \text{with probability } (1-\delta). \end{cases} \tag{3}$$

Note that when $\delta = 0$, the distribution of $\xi_{i,j}$ collapses to $\xi_{i,j} = 1$ with probability 1. Let $\odot$ denote the binary operator defining element-wise multiplication between two vectors of the same dimension. Consider the covariate vector

$$x_i \odot \xi_i \equiv (x_{i,1}\xi_{i,1}, \ldots, x_{i,d}\xi_{i,d})^\top. \tag{4}$$

Some entries of the new covariate vector are 0 (those for which $\xi_{i,j} = 0$) and the rest are equal to $x_{i,j}/(1-\delta)$.

In a slight abuse of notation, let $\mathbb{E}_\delta$ denote the distribution of the random vector ξ_i, whose distribution is parameterized by δ. The estimators of (β, ϕ) obtained by *dropout training* correspond to any parameters $(\widehat{\beta}(\delta), \widehat{\phi}(\delta))$ that solve the problem

$$\min_{\beta,\phi} \ \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\delta \left[-\ln f(y_i | x_i \odot \xi_i, \beta, \phi) \right]. \tag{5}$$

One possibility to solve (5) is to use Stochastic Gradient Descent (Robbins and Monro (1951)). This is tantamount to i) taking a draw of (x_i, y_i) according to its empirical distribution, ii) independently taking a draw of ξ_i using the distribution in (3) and iii) computing the stochastic gradient descent update using

$$\nabla \ln f(y_i | x_i \odot \xi_i, \beta, \phi).$$

We provide further details about the Stochastic Gradient Descent implementation of dropout training in Section 6.

2.3 Question of Interest

Adding noise to the Maximum Likelihood objective in (2) seems, at first glance, arbitrary. Our first obvious observation is that dropout training estimators will generally not share the same probability limit as the Maximum Likelihood estimators whenever $\delta \neq 0$. This can be formalized under the following assumptions:

Assumption 1 *The log-partition function $\Psi(\cdot)$ has a bounded second derivative.*

Assumption 2 *The second moment matrix $\mathbb{E}_{P^\star}[XX^\top]$ is finite, positive definite.*

Proposition 1 (Consistency) *Suppose that Assumptions 1 and 2 hold. Then for any sequence $\delta_n \to \delta \in [0, 1)$ as $n \to \infty$, $\widehat{\beta}(\delta_n)$ converges in probability to*

$$\beta^\star(\delta) \equiv \arg\min_{\beta} \ E_{P^\star} \left[\mathbb{E}_\delta \left[-\ln f(Y | X \odot \xi, \beta, \phi) \right] \right], \tag{6}$$

where the minimizer in (6) is unique and does not depend on ϕ.

Proof See Appendix A.1. ■

The proof of this result consists of expressing dropout training as an extremum estimator and verifying standard conditions for consistency in Newey and McFadden (1994). The main message of the proposition above is that the parameter $\beta^\star(0)$—which gives the probability limit of the Maximum Likelihood estimator—generally differs from $\beta^\star(\delta)$ when $\delta \neq 0$, which is the probability limit of the dropout estimator.[2] To further illustrate this point, it is helpful to workout the details of the probability limit of the dropout estimator in the linear regression model. Algebra shows that in this model

$$\beta^\star(\delta) = \Big(\mathbb{E}_{P^\star}[XX^\top] + (\delta/1-\delta)\text{diag}(\mathbb{E}_{P^\star}[XX^\top])\Big)^{-1}\mathbb{E}_{P^\star}[YX],$$

which can be interpreted as a population version of the Ridge estimator. This estimator differs from the best linear predictor of y using x as long as $E_{P^\star}[YX] \neq 0$ and $\delta \neq 0$. This estimator differs from Ridge regression in that $\mathbb{E}_{P^\star}[XX^\top]$ replaces the identity matrix (and this simple adjustment makes the estimator scale equivariant).

Despite the lack of consistency, there is some literature that has provided empirical evidence that using intentionally corrupted features for training has the potential to improve the performance of machine learning algorithms; see Maaten et al. (2013). Even if one is willing to accept that corrupting features is desirable for estimation, the choice of dropout noise in (3) remains quite arbitrary.

The main contribution of this paper is to provide a novel decision-theoretic interpretation of dropout training (in the population and in the sample). We will argue there is a natural two-player, zero-sum game between a decision maker (statistician) and an adversary (nature) in which dropout training emerges naturally as a minimax solution. In this game, dropout noise turns out to be nature's least favorable distribution, and dropout training becomes the statistician's optimal action. The framework we use is known in the stochastic optimization literature as Distributionally Robust Optimization and we describe it very generally in the next section. We will therein also revisit the interpretation of $\beta^*(\delta)$ in the linear regression model.

3. Problem Setup

Consider a general problem where there is a multivariate predictor $X \in \mathbb{R}^d$ and a scalar outcome variable $Y \in \mathbb{R}$. A Distributionally Robust Optimization (DRO) problem is a simultaneous two-player zero sum game between a decision maker (statistician) and an

[2] Relatedly, Farrell et al. (2020)—who study deep neural networks and their use in semiparametric inference—report that their numerical exploration of dropout increased bias and interval length compared to nonregularized models.

adversary (nature).[3] In this section we describe the action space for each player, their strategies, and the payoff function.

Actions and Payoff: The statistician's action space consists of vectors $\theta \in \Theta$. The ranking of the statistician's actions is contingent on the realization of (X, Y), and this is captured by a real-valued loss function $\ell(X, Y, \theta)$. We assume that the statistician is called to choose an action before observing the realization of (X, Y). If the statistician knew the distribution of (X, Y)—which we denote by $\mathbb{Q}$—the statistician's preferred choice of θ would be the solution to

$$\inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}} \left[\ell(X, Y, \theta) \right]. \tag{7}$$

Instead of assuming that the distribution $\mathbb{Q}$ is exogenously determined, we think of the distribution $\mathbb{Q}$ as being chosen by nature. Thus, nature's action space consists of a set of probability distributions denoted as $\mathcal{U}$. We refer to this set as the *distributionally uncertainty set.* If nature knew the action selected by the statistician, nature's preferred action would be

$$\sup_{\mathbb{Q} \in \mathcal{U}} \mathbb{E}_{\mathbb{Q}}[\ell(X, Y, \theta)]. \tag{8}$$

Strategies and Solution: The choice of θ and $\mathbb{Q}$ are assumed to happen simultaneously. A statistician's strategy for this game consists of a choice of θ. Likewise, nature's strategy for this game consists of a choice of $\mathbb{Q}$.

A *Nash equilibrium* for this game is a pair $(\theta^\star, \mathbb{Q}^\star)$ such that: a) given $\mathbb{Q}^\star$, the parameter $\theta^\star$ solves (7) and b) given $\theta^\star$, the distribution $\mathbb{Q}^\star$ solves (8).

The *minimax solution* for this game is a pair $(\theta^*, \mathbb{Q}^*)$ that solves

$$\inf_{\theta \in \Theta} \sup_{\mathbb{Q} \in \mathcal{U}} \mathbb{E}_{\mathbb{Q}}[\ell(X, Y, \theta)], \tag{9a}$$

while the *maximin* solution is based on the program

$$\sup_{\mathbb{Q} \in \mathcal{U}} \inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}}[\ell(X, Y, \theta)]. \tag{9b}$$

If $\mathbb{Q}^\star$ solves (9b), we say that $\mathbb{Q}^\star$ is *nature's least favorable distribution.* The mathematical program in (9a) is typically referred to as a DRO problem.

[3] A seminal reference is the robust inventory control problem of Scarf (1958). Recent references describing the use of distributionally robust stochastic programs (as those considered in this paper) are Delage and Ye (2010) and Shapiro (2017). Christensen and Connault (2019) used distributionally robust optimization to characterize the sensitivity of counterfactual analysis with respect to distributional assumptions in a class of structural econometric models.

4. Dropout Training is Distributionally Robust Optimal

The previous section provided a general description of a Distributionally Robust Optimization problem. We specialize the general framework of Section 3 by imposing two restrictions. First, we use the negative log-likelihood of Generalized Linear Models (McCullagh and Nelder, 1989) as a loss function for the statistician. Second, we define nature's uncertainty set (i.e., the possible data distributions that nature can take) using the multiplicative errors-in-variables model of Hwang (1986).

4.1 Statistician's Payoff

We define the loss function for the statistician to be the negative of the logarithm of the likelihood in (1), that is,

$$\ell(X, Y, \theta) = -\ln h(Y, \phi) + (\Psi(\beta^\top X) - Y(\beta^\top X))/a(\phi), \tag{10}$$

where $\theta \equiv (\beta^\top, \phi^\top)^\top \in \Theta$. Equation (10) defines the statistician's objective and its set of actions.

4.2 Nature's Distributionally Uncertainty Set

We now define the possible distributions that nature can choose. We start out by letting $\mathbb{Q}_0$ denote some benchmark or *reference* distribution over (X, Y). This distribution need not correspond to that induced by a Generalized Linear Model. In other words, our framework allows for the statistician's model to be misspecified.

Next, we define nature's action space by considering perturbations of $\mathbb{Q}_0$. Although there are different ways of doing this—for example, by using either f-divergences (such as the Kullback-Leibler as in Nguyen et al. (2020)) or the optimal transport distance (such as the Wasserstein distance as in Blanchet et al. (2019b)) to define a neighborhood—we herein use a nonparametric multiplicative errors-in-variables model as in Hwang (1986).

The idea is to allow nature to independently introduce measurement error to the covariates using multiplicative noise. Multiplicative errors have found different applications in empirical work across different disciplines, from economics to epidemiology. For example, Alan et al. (2009) use it to account for measurement error in consumption data when estimating the elasticity of intertemporal substitution via Euler equations. Pierce et al. (1992) and Lyles and Kupper (1997) use it to relate health outcomes to the exposure of a chemical toxicant that is observed with error. Moreover, due to privacy considerations, statistical agencies such as the U.S. Census Bureau sometimes mask data using multiplicative noise; see the discussion in Kim and Winkler (2003) and Nayak et al. (2011). Examples of datasets that contain variables masked with multiplicative noise include the Commodity Flow Survey

Data (2017), the Survey of Business Owners (2012)—both from the U.S. Census Bureau—and the U.S. Energy Information Administration Residential Energy Consumption survey.

Let $\xi \equiv (\xi_1, \ldots, \xi_d)^\top$ be defined as a d-dimensional vector of random variables that are independent of (X, Y). We perturb the distribution $\mathbb{Q}_0$ by considering the transformation

$$(X, Y) \mapsto (X_1\xi_1, \ldots, X_d\xi_d, Y)^\top.$$

As a result, each covariate X_j is distorted in a multiplicative fashion by ξ_j. We often abbreviate $(X_1\xi_1, \ldots, X_d\xi_d)^\top$ by $X \odot \xi$, where $\odot$ is the element-wise multiplication.

We restrict the distribution of ξ in the following way. First, for a parameter $\delta \in [0, 1)$, we define $\mathcal{Q}_j(\delta)$ to be the set of distributions for ξ_j that are supported on the interval $[0, 1/(1-\delta)]$ and that have mean equal to 1. More specifically,

$$\mathcal{Q}_j(\delta) \equiv \left\{\mathbb{Q}_j : \mathbb{Q}_j \text{ is a probability distribution on } \mathbb{R},\, \mathbb{Q}_j([0, (1-\delta)^{-1}]) = 1,\; \mathbb{E}_{\mathbb{Q}_j}[\xi_j] = 1\right\}. \tag{11}$$

This set of distributions prescribed using support and first-order moment information is popular in the DRO literature thanks to its simplicity and tractability (Wiesemann et al., 2014). From the perspective of an errors-in-variables model, these distributions are also attractive because they preserve the expected value of the covariates, assuming that X_j and ξ_j are drawn independently.

Consider now the joint random vector $(X, Y, \xi) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d$. For a constant $\delta \in [0, 1)$ consider the joint distributions over (X, Y, ξ) defined by

$$\mathcal{U}(\mathbb{Q}_0, \delta) = \{\mathbb{Q}_0 \otimes \mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d : \mathbb{Q}_j \in \mathcal{Q}_j(\delta)\ \forall j = 1, \ldots, d\}, \tag{12}$$

where $\otimes$ is used to denote the product measure (meaning that the joint distribution is the product of the independent marginals $\mathbb{Q}_j$, $j = 0, \ldots, d$). Thus, in the game we consider $\mathcal{U}(\mathbb{Q}_0, \delta)$ is nature's action space or *nature's distributionally uncertainty set.*

We will make only one assumption about the reference distribution $\mathbb{Q}_0$:

Assumption 3 *The distribution $\mathbb{Q}_0$ satisfies $\mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)] < \infty$ for any $\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)$, any $\theta \in \Theta$, and any scalar $\delta \in [0, 1)$.*

This assumption implies a minimal regularity condition to guarantee that the expected loss is well-defined for both the statistician and nature. Assumption 3 is trivially satisfied when $\mathbb{Q}_0$ is the empirical distribution of the data, which is one of the main cases of interest in the paper.

4.3 Dropout Training is DRO

We now present the main result of this section.

Theorem 2 *Consider the two-player zero sum game where the statistician has the loss function in* (10) *and nature has the action space in* (12) *for some reference distribution* $\mathbb{Q}_0$ *and a scalar* $\delta \in [0,1)$. *If Assumption 3 is satisfied, then the minimax solution of the two-player zero sum game defined by* (10) *and* (12)

$$\inf_{\theta \in \Theta} \sup_{\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)} \mathbb{E}_{\mathbb{Q}} \left[\ell(X \odot \xi, Y, \theta) \right] \tag{13}$$

is equivalent to

$$\inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}^\star} [\ell(X \odot \xi, Y, \theta)], \tag{14}$$

where $\mathbb{Q}^\star = \mathbb{Q}_0 \otimes \mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$, *and* $\mathbb{Q}_j^\star = (1-\delta)^{-1} \times \text{Bernoulli}(1-\delta)$ *is a scaled Bernoulli distribution for any* $j = 1, \ldots, d$, *i.e., under* $\mathbb{Q}_j^\star$

$$\xi_j = \begin{cases} 0 & \text{with probability } \delta, \\ (1-\delta)^{-1} & \text{with probability } (1-\delta). \end{cases} \tag{15}$$

In addition, let $\theta^\star \in \Theta$ *be a solution to* (14). *Then* $(\theta^\star, \mathbb{Q}^\star)$ *constitutes a Nash equilibrium of the two-player zero sum game defined by* (10) *and* (12) *and* $\mathbb{Q}^\star$ *is nature's least favorable distribution.*

Proof See Appendix A.2. ■

The first part of theorem characterizes the statistician's best response to an adversarial nature that is allowed to corrupt the covariates using a multiplicative errors-in-variables model. From the statistician's perspective, nature's worst-case perturbation of $\mathbb{Q}_0$ is given by $\mathbb{Q}^\star$ in (15). Under this *worst-case* distribution, nature independently corrupts each of the entries of $X = (X_1, \ldots, X_d)^\top$, by either dropping the j-th component (if $\xi_j = 0$) or replacing it by $X_j/(1-\delta)$. Dropout training—which here refers to estimating the parameter θ after adding dropout noise to X—thus becomes the statistician's preferred way of estimating the parameter θ when facing an adversarial nature. This gives a decision-theoretic foundation for the use of dropout training.

Note that in order to recover the objective function introduced in (5) (the sample average of the contaminated log-likelihood) it suffices to set the reference measure—$\mathbb{Q}_0$—as the empirical distribution $\widehat{\mathbb{P}}_n$ of $\{(x_i, y_i)\}_{i=1}^n$, which satisfies Assumption 3. Likewise, Theorem 2 allows to interpret the probability limit $\beta^\star(\delta)$ of the dropout estimator derived in

Proposition 1 as a solution to a DRO problem. In particular

$$\beta^\star(\delta) = \arg\min_{\beta} \sup_{\mathbb{Q}\in\mathcal{U}(P^*,\delta)} \mathbb{E}_{\mathbb{Q}}\left[-\ln f(Y|X\odot\xi,\beta,\phi)\right],$$

provided the data-generating distribution P^* satisfies Assumption 3. In the specific case of the linear regression model, this means that we can interpret $\beta^\star(\delta)$ as giving us the population's best linear predictor for y in terms of x, provided nature is allowed to perturb the distribution of covariates using a multiplicative error-in-variables model.

More generally, because dropout noise is nature's best response for every β, the dropout loss is an upper bound for the true loss:

$$\mathbb{E}_{\mathbb{Q}^\star}\left[-\ln f(Y|X\odot\xi,\beta^\star(\delta),\phi)\right] \geq \mathbb{E}_{P^*}\left[-\ln f(Y|X\odot\xi,\beta^\star(0),\phi)\right].$$

We provide now some intuition about how dropout noise becomes nature's worst-case distribution. Algebra shows that, in light of Assumption 3, the expected loss under an arbitrary distribution $\mathbb{Q}$ is finite and can be written as

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi,Y,\theta)] = &-\mathbb{E}_{\mathbb{Q}_0}\left[\ln h(Y,\phi)\right]\\ &+\mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[(\Psi((\beta\odot X)^\top\xi)-Y((\beta\odot X)^\top\xi))/a(\phi)]\right],\end{aligned}$$

where the first expectation is taken with respect to the reference distribution, and the second one with respect to ξ. For fixed values of (X,Y,θ) we can define

$$A_{(X,Y,\theta)}((\beta\odot X)^\top\xi) \equiv (\Psi((\beta\odot X)^\top\xi)-Y((\beta\odot X)^\top\xi))/a(\phi).$$

Because $\Psi(\cdot)$ is a convex function defined on all of the real line, the function $A_{(X,Y,\theta)}(\cdot)$ inherits these properties. We show in the appendix that for these type of functions

$$\sup\left\{\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A_{(X,Y,\theta)}((\beta\odot X)^\top\xi)] : \mathbb{Q}_j\in\mathcal{Q}_j(\delta)\right\} = \mathbb{E}_{\mathbb{Q}_1^\star\otimes\ldots\otimes\mathbb{Q}_d^\star}[A_{(X,Y,\theta)}((\beta\odot X)^\top\xi)], \tag{16}$$

for any θ, and this establishes the equivalence between (13) and (14). The proof of the equality above exploits convexity. In fact, to derive our result we first characterize the worst-case distribution for the expectation of a real-valued convex function (Lemma 5) and then we generalize this result to functions that depend on ξ only through linear combinations, as $A_{(X,Y,\theta)}(\cdot)$ (Proposition 6).

How about the Nash Equilibrium of the two-player zero sum game defined by (10) and (12)? The equality in (16) clearly shows that $\mathbb{Q}^\star$ is nature's best response for any $\theta\in\Theta$. If there is a vector $\theta^\star$ that solves the dropout training problem in (14), then this vector is the statistician best's response to nature's choice of $\mathbb{Q}^\star$. Consequently, $(\theta^\star, \mathbb{Q}^\star)$ is a Nash equilibrium.

Finally, we discuss the extent to which $\mathbb{Q}^\star$ can be referred to as nature's least favorable distribution, which has been defined as nature's solution to the maximin problem. It is well known that the maximin value of a game is always smaller than its minimax value:[4]

$$\sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)] \le \inf_{\theta\in\Theta} \sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)].$$

We have shown that the right-hand side of the display above equals (14). Therefore, if there is a $\theta^\star \in \Theta$ that solves such program, then $\mathbb{Q}^\star$ achieves the upper bound to the maximin value of the game. This makes dropout noise nature's least favorable distribution.

Now that we have established that dropout training gives the minimax solution of the DRO game, we discuss the implications of this result regarding the out-of-sample performance of dropout training. Suppose $\mathbb{Q}_0$ is the empirical measure $\widehat{\mathbb{P}}_n$ supported on n training samples $\{(x_i, y_i)\}_{i=1}^n$. The *in-sample* loss of dropout training is

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbb{Q}^\star}[\ell(X\odot\xi, Y, \theta^\star)|X = x_i, Y = y_i]. \tag{17}$$

A typical concern with estimation procedures is whether their performance in a specific sample translates to good performance out of sample. In our context, the out-of-sample performance of dropout training can be thought of as the expected loss that would arise for some other data distribution $\tilde{\mathbb{Q}}_0$ over (X, Y) at the parameter estimated via dropout training:

$$\mathbb{E}_{\tilde{\mathbb{Q}}_0}[\ell(X, Y, \theta^\star)].$$

The minimaxity of dropout training shows that for any distribution $\tilde{\mathbb{Q}}_0$ over (X, Y) that can be obtained from $\widehat{\mathbb{P}}_n$ by perturbing covariates with mean-one independent multiplicative error $\xi_j \in [0, (1-\delta)^{-1}]$ we have

$$\mathbb{E}_{\tilde{\mathbb{Q}}_0}[\ell(X, Y, \theta^\star)] \le \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbb{Q}^\star}[\ell(X\odot\xi, Y, \theta^\star)|X = x_i, Y = y_i].$$

[4] This follows from the fact that for any $\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)$:

$$\inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)] \le \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)] \le \sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)].$$

See also the discussion of the minimax theorem in Ferguson (1967) p. 81.

This means that the out-of-sample loss will be upper-bounded by the in-sample loss. Thus, our results give a concrete result about the class of distributions for which dropout training estimation "generalizes" well.

Finally, we note that Theorem 2 was stated for a scalar δ that is homogeneous across the multiplicative noise ξ_j. To model non-identical dropout noise, we can substitute the sets in (11) and (12) by $\mathcal{Q}_j(\delta_j)$ for a collection of parameters $(\delta_1, \dots, \delta_d) \in [0,1)^d$. In this case, the results of Theorem 2 hold with $\mathbb{Q}_j^\star = (1-\delta_j)^{-1} \times \text{Bernoulli}(1-\delta_j)$ for $j = 1, \dots, d$.

5. Statistical Guidance on Choosing δ

Theorem 2 in the previous section showed that dropout training is distributionally robust optimal and that nature's least favorable distribution is dropout noise with probability $\delta \in [0,1)$. This section suggests a strategy to pick this parameter. Broadly speaking, our approach relies on a simple idea: we study how often the in-sample loss obtained from dropout training exceeds the population loss. If overfitting is successfully mitigated, this probability ought to be large. We show that by appropriately tuning the parameter δ of the dropout noise, it is possible to control the probability of such event as the sample size grows large.

5.1 Additional Notation

Throughout this section, we use $\widehat{\phi}$ to denote an arbitrary $\sqrt{n}$-consistent, asymptotically normal estimator for the scale parameter ϕ. Such estimator can be obtained, for example, by using $\widehat{\phi}_{\text{ML}}$ in (2). We use $\phi^\star$ to denote the true, unknown scale parameter. Just as before, we let $\widehat{\beta}(\delta)$ denote the dropout estimator of the true $\beta^\star$ under dropout probability δ.

The in-sample loss of dropout training—given a dropout probability of δ—evaluated at parameters β and ϕ is given by

$$\mathcal{L}_n(\beta, \phi, \delta) \equiv \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\delta \left[-\ln f(y_i | x_i \odot \xi_i, \beta, \phi) \right]. \tag{18}$$

The goal of this section is to understand how likely is that the in-sample loss in (18)—when evaluated at the dropout estimator $\widehat{\beta}(\delta)$ and some estimator $\widehat{\phi}$—exceeds the true population loss. Thus, if we define the population loss as

$$\mathcal{L}(\beta^\star, \phi^\star) \equiv \mathbb{E}_{P^\star} \left[-\ln f(Y | X, \beta^\star, \phi^\star) \right], \tag{19}$$

we are interested in understanding how often

$$\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}, \delta_n) \geq \mathcal{L}(\beta^\star, \phi^\star), \tag{20}$$

where the sequence δ_n is allowed to change with the sample size.

5.2 Additional Assumptions

It is well-known that under mild regularity conditions on the true data generating process—and without the need of dropout training—the usual in-sample loss evaluated at the Maximum Likelihood estimators, which we can denote as $\mathcal{L}_n(\widehat{\beta}_{\text{ML}}, \widehat{\phi}_{\text{ML}}, 0)$, provides a consistent estimator for the population loss. This remains true if the Maximum Likelihood estimator for ϕ is replaced by another $\sqrt{n}$-consistent estimator $\widehat{\phi}$. One sufficient condition that guarantees such behavior is the following high-level assumption:

Assumption 4 *The following central limit theorem result holds for some σ^2*

$$\sqrt{n}\left(\mathcal{L}_n(\beta^\star, \widehat{\phi}, 0) - \mathcal{L}(\beta^\star, \phi^\star)\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2).$$

In particular, the identity

$$\begin{aligned}\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}_{\text{ML}}, \widehat{\phi}, 0) - \mathcal{L}(\beta^\star, \phi^\star)\right) =& \sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}_{\text{ML}}, \widehat{\phi}, 0) - \mathcal{L}_n(\beta^\star, \widehat{\phi}, 0)\right) \\ &+ \sqrt{n}\left(\mathcal{L}_n(\beta^\star, \widehat{\phi}, 0) - \mathcal{L}(\beta^\star, \phi^\star)\right),\end{aligned}$$

and Assumptions 1, 2, 4 imply

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}_{\text{ML}}, \widehat{\phi}, 0) - \mathcal{L}(\beta^\star, \phi^\star)\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2). \tag{21}$$

We view the result in (21) concerning the in-sample loss of Maximum Likelihood estimators, which is quite standard, as unsatisfactory. Our main complaint is that the probability of the event

$$\mathcal{L}_n(\widehat{\beta}_{\text{ML}}, \widehat{\phi}, 0) \leq \mathcal{L}(\beta^\star, \phi^\star)$$

approaches 1/2 as the sample size grows large. Our interpretation is that the in-sample loss at the Maximum Likelihood estimator is deceivingly small, as the true population loss will be above it 50% of the time if the sample size is large enough. We argue that this probability can be made smaller by appropriately tuning dropout noise.

5.3 In-sample loss of dropout training

Theorem 2 showed that dropout noise is nature's choice to inflict the highest loss for the statistician at any parameter values. Therefore, invoking Theorem 2 using the empirical distribution as the reference distribution, the random variable

$$\mu_n(\beta,\phi,\delta) \equiv \mathcal{L}_n(\beta,\phi,\delta) - \mathcal{L}_n(\beta,\phi,0) \tag{22}$$

is nonnegative for any $\delta \in [0,1)$. Since Proposition 1 has shown that $\widehat{\beta}(\delta_n) \xrightarrow{p} \beta^\star$ for any sequence $\delta_n \to 0$, intuition suggests that the in-sample loss of dropout training, $\mathcal{L}_n(\widehat{\beta}(\delta_n),\widehat{\phi},\delta_n)$, may provide a consistent estimate of the population loss that does not underestimate this limit frequently.

Our result is the following proposition:

Proposition 3 *Suppose that Assumptions 1, 2, 4 hold. Then for any sequence* $\delta_n = c/\sqrt{n}$*,*

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n),\widehat{\phi},\delta_n) - \mathcal{L}(\beta^\star,\phi^\star)\right) \xrightarrow{d} \mathcal{N}(\mu_\infty(\beta^\star,\phi^\star,c),\sigma^2),$$

where $\mu_\infty(\beta^\star,\phi^\star,c) \geq 0$ *is the probability limit of*

$$\sqrt{n}\mu_n\left(\beta^\star,\widehat{\phi},\delta_n\right),$$

and $\mu_n(\cdot)$ *is defined as in (22).*

Proof See Appendix A.3. ■

The main message of Proposition 3 is that the probability of the event (20) can be approximated, as the sample size goes large, by the probability—under a normal random variable with positive mean—of the positive half of the real line. Some elementary algebra can be used to illustrate the main argument behind the proof. Note that

$$\begin{aligned}\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n),\widehat{\phi},\delta_n) - \mathcal{L}(\beta^\star,\phi^\star)\right) &= \sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n),\widehat{\phi},\delta_n) - \mathcal{L}_n(\beta^\star,\widehat{\phi},\delta_n)\right)\\ &+ \sqrt{n}\mu_n(\beta^\star,\widehat{\phi},\delta_n)\\ &+ \sqrt{n}\left(\mathcal{L}_n(\beta^\star,\widehat{\phi},0) - \mathcal{L}(\beta^\star,\widehat{\phi})\right).\end{aligned}$$

We start by showing that the first term converges in probability to zero. To do this we show that

$$\sqrt{n}(\widehat{\beta}(c/\sqrt{n}) - \beta^\star)$$

is asymptotically normal and that the derivative of $\mathcal{L}_n(\cdot)$ with respect to β (evaluated at $\beta^\star$) converges in probability to zero.

The key step in the proof shows that the second term has a finite probability limit. In fact, we can characterize this limit explicitly and show that

$$\sqrt{n}\mu_n(\beta^\star, \widehat{\phi}, \delta_n) \xrightarrow{p} c \cdot \mu,$$

where

$$\mu \equiv \left(\left(\sum_{\xi \in \mathcal{A}} \mathbb{E}_{P^\star}[\Psi((X \odot \xi)^\top \beta^\star)] \right) - d\mathbb{E}_{P^\star}[\Psi(X^\top \beta^\star)] + \mathbb{E}_{P^\star}[YX^\top]\beta^\star \right) \Big/ a(\phi^\star),$$

and $\mathcal{A}$ is the collection of all vectors in $\{0,1\}^d$ for which there is only one zero. In Section 7 we provide an expression for this term in the linear regression model.

It is important to mention that the DRO interpretation of dropout training can be leveraged to select the dropout parameter δ. For example, a possible approach consists in choosing δ so that true data generating process belongs to nature's choice set with some prespecified probability. This approach, which is often advocated in the literature in machine learning and robustness (Hansen and Sargent, 2008), often leads to a very pessimistic selection of δ simply because this criterion is not informed at all by the loss function defining the decision problem. Further, in our problem, it is not possible to apply this approach given that the set of multiplicative perturbations of the empirical distribution will, in general, not cover the true data generating process.

Another approach involves using generalization bounds leading to finite sample guarantees; see, for instance a summary of this discussion in Section 6.2 of Rahimian and Mehrotra (2019). This method, while appealing, often requires either distributions with compact support or strong control on the tails of the underlying distributions. Also, often, the bounds depend on constants that may be too pessimistic or difficult to compute.

Finally, there is a recent method introduced in Blanchet et al. (2019b) for the case in which nature's choice set is defined in terms of the Wasserstein's distance around the empirical distribution. The idea therein is that—for a fixed δ—every distribution that belongs to nature's choice set corresponds to an optimal parameter choice for the statistician. Thus, one can collect each and every of the statistician's optimal choices associated to each distribution in nature's uncertainty set, and treat the resulting region as a confidence set for the true parameter. This confidence set grows bigger (in the sense of nested confidence regions) as δ increases. The goal is then to minimize δ subject to a desired level of coverage in the underlying parameter to estimate. This leads to a data-driven choice of δ that is explicitly linked to the statistician's decision problem. However, this approach is not feasible in our problem because, once again, regardless of the value of δ, the parameter choices for each of the multiplicative perturbations of the empirical distribution will, in general, fail to cover the true parameter.

Hence, we advocate the strategy of choosing the parameter δ to control how often the in-sample loss obtained from dropout training exceeds the population loss. The proof of Proposition 3 shows that $\mu_\infty(\beta^\star, \phi^\star, c)$ is of the form $c \cdot \mu$, where μ depends on $(\beta^\star, \phi^\star)$. Consequently, as long as $\mu > 0$, it is straightforward to pick c to guarantee a pre-specified "coverage" of the population loss: for any $\alpha \in (0,1)$, if we pick c to be

$$z_{1-\alpha} \cdot \sigma / \mu,$$

where $z_{1-\alpha}$ is the 1-α quantile of a standard normal, then the probability of the event (20) asymptotically approaches $1-\alpha$. Thus, overfitting can be successfully mitigated.

6. An Algorithm for Dropout Training

The goal of this section is to suggest an algorithm for solving the dropout training problem

$$\inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta)],$$

where $\mathbb{Q}^\star = \widehat{\mathbb{P}}_n \otimes \mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$ and $\mathbb{Q}_j^\star$, $j = 1, \ldots, d$ is the dropout noise distribution defined in (15). Notice that we here consider the specific case in which $\mathbb{Q}_0$ is set to the empirical measure $\widehat{\mathbb{P}}_n$ supported on n training samples $\{(x_i, y_i)\}_{i=1}^n$. We will use $\theta_n^\star$ to denote the solution of the dropout training problem above. It will sometimes be convenient to rewrite this dropout training problem as

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta) \mid X = x_i, Y = y_i], \tag{23}$$

which coincides with expression (5). Conditioning on the values of (x_i, y_i) makes it clear that the expectation is computed over the d-dimensional vector ξ. We now briefly describe three common approaches to implement dropout training and we discuss some of its limitations.

6.1 Naive Dropout Training

Because $\mathbb{Q}_j^\star$ places mass on only two points, namely 0 and $(1-\delta)^{-1}$, the support of the joint distribution $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$ has cardinality 2^d. Thus, a naive approach to solve the dropout training problem (23) is to expand the objective function as a sum with $n \cdot 2^d$ terms, then to apply a tailored gradient descent algorithm to the resulting optimization problem. Unfortunately, this approach is computationally demanding because the number of individual terms in the objective function grows exponentially with the dimension d of the features.

6.2 Dropout Training via Stochastic Gradient Descent

Another method to solve the dropout training problem in (14) is by stochastic gradient descent (henceforth, SGD). This gives us the commonly used dropout training algorithm. For the sake of comparison, we provide concrete details about this algorithm below.

Given a current estimate $\widehat{\theta}$, we compute an unbiased estimate of the gradient to the objective function of (14), and move in the direction of the negative gradient with a suitable step size. Since $\mathbb{Q}^\star$ is discrete, the expectation under $\mathbb{Q}^\star$ can be written as a finite sum and by differentiating under the expectation, we have

$$\nabla_\theta \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \widehat{\theta})] = \mathbb{E}_{\mathbb{Q}^\star}\left[\nabla_\theta \ell(X \odot \xi, Y, \widehat{\theta})\right]. \tag{24}$$

The standard SGD algorithm uses a naive Monte Carlo estimator as an estimate of the gradient (24), that is, at iterate $k \in \mathbb{N}$ with incumbent solution $\widehat{\theta}^k$,

$$\nabla_\theta \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \widehat{\theta}^k)] \approx \nabla_\theta \ell(x_k \odot \xi_k, y_k, \widehat{\theta}^k),$$

where (x_k, y_k, ξ_k) is an independent draw from $\mathbb{Q}^\star$.

One drawback of using SGD to solve (14) is that it is not easily parallelizable, and thus its implementation can be quite slow. Moreover, under strong convexity assumption of the loss function ℓ, SGD only exhibits linear convergence rate (Nemirovski et al., 2009, Section 2.1). By contrast, the gradient descent (GD) enjoys exponential convergence rate (Boyd and Vandenberghe, 2004, Section 9.3.1).

6.3 Naive Monte Carlo Approximation for Dropout Training

Consider solving the dropout training problem in (23) using a naive Monte Carlo approximation. Instead of using 2^d terms to compute

$$\mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta) \mid X = x_i, Y = y_i],$$

we approximate this expectation by taking a large number of K i.i.d. draws $\{\xi_i^k\}_{k=1}^K$, $\xi_i^k \in \mathbb{R}^d$, according to the distribution $\mathbb{Q}_1^* \otimes \ldots \otimes \mathbb{Q}_d^*$. When d is large this approximation is computationally cheaper than the naive dropout training procedure described above, provided that $K \ll 2^d$.

Thus, the naive Monte Carlo approximation of the dropout training problem is

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \left[\frac{1}{K} \sum_{k=1}^{K} \ell(x_i \odot \xi_i^k, y_i, \theta) \right], \tag{25}$$

where the random vectors ξ_i^k are sampled independently—over both k and i—using the distribution $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$.

Relative to the solution of the dropout training problem—which we denoted as $\theta_n^\star$—the minimizer of (25) is consistent and asymptotically normal as $K \to \infty$. This follows by standard arguments; for example, those in Shapiro et al. (2014, Section 5.1). There are, however, two problems that arise when using (25) as a surrogate for the dropout training problem. First, the solution to (25) is a biased estimator for $\theta_n^\star$. This means that if we average the solution of (25) over the $K \cdot n$ different values of ξ_i^k, the average solution need not equal $\theta_n^\star$. Second, implementing (25) requires a choice of K and, to the best of our knowledge, there is no off-the-shelf procedure for picking this number.

6.4 Unbiased Multi-level Monte Carlo Approximation for Dropout Training

To address these two issues, we apply the recent techniques suggested in Blanchet et al. (2019a) that we refer to as *Unbiased Multi-level Monte Carlo Approximations.* Multi-level Monte Carlo methods (Giles, 2008, 2015) refer to a set of techniques for approximating the expectation of random variables. The adjective "multi-level" emphasizes the fact that random samples of different *levels* of accuracy are used in the approximation. Before presenting the detailed algorithm, we provide a heuristic description. To this end, let $\widehat{\theta}_n^\star(K)$ denote the *level* K solution of the problem in (25); that is, the solution based on K draws. Define the random variable

$$\Delta_K \equiv \widehat{\theta}_n^\star(K) - \widehat{\theta}_n^\star(K-1).$$

and, for simplicity, assume $\widehat{\theta}_n^\star(0)$ is defined to equal a vector of zeros. Under suitable regularity conditions, there holds

$$\sum_{K=1}^{\infty} \mathbb{E}[\Delta_K] = \lim_{K \to \infty} \mathbb{E}[\widehat{\theta}_n^\star(K)] = \theta_n^\star.$$

Consider now picking K^* at random from some discrete distribution supported on the natural numbers. Let $p(\cdot)$ denote the probability mass function of such distribution and consider a Monte Carlo approximation scheme in which—after drawing K^*—we sample

$K^* \cdot n$ different random vectors $\xi_i^k \in \mathbb{R}^d$ according to $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$. The estimator

$$Z(K^*) \equiv \frac{\Delta_{K^*}}{p(K^*)}$$

has two sources of randomness. Firstly, the random choice of K^* and, secondly, the random draws ξ_i^k. Averaging over both yields

$$\mathbb{E}[Z(K^*)] = \sum_{K=1}^{\infty} \mathbb{E}[Z(K^*)|K^* = K] \cdot p(K) = \sum_{K=1}^{\infty} (\mathbb{E}[\Delta_K]/p(K)) \cdot p(K) = \theta_n^\star.$$

Thus, by taking into account the randomness in the selection of K, we have managed to provide a rule for deciding the number of draws (specifically, our recommendation is to pick K^* at random) and at the same time we have removed the bias of naive Monte Carlo approximations.

One possible concern with our suggested implementation is that the expected computational cost of $Z(K^*)$ could be infinitely large. Fortunately, this issue can be easily resolved by an appropriate choice of the distribution $p(\cdot)$. To see this, define the computational cost simply as the number of random draws that are required to obtain $Z(K^*)$. In the construction we have described above, we need $K^* \cdot n$ draws for the construction of the estimator. Thus, the average cost is

$$\mathbb{E}[K^* \cdot n] = n \sum_{K=1}^{\infty} K \cdot p(K)$$

which, under mild integrability conditions on $p(\cdot)$, will be finite.[5]

We now present the algorithm that will be used to solve the dropout training problem. To ensure that the estimator $Z(K^*)$ has a finite variance, instead of defining Δ_K as the difference between the level K and $K-1$ solutions to problem (25) in the above heuristic arguments, we use solutions to problem (25) with a sample of size 2^{K+1} and with its *odd* and *even* sub-samples of size 2^K.

Algorithm for the Unbiased Multilevel Monte Carlo: We present a parallelized version of it using L processors, but the suggested algorithm works even when $L = 1$. Parallel computing reduces the variance of the estimator, and our suggestion is to use as many processors as available in one run.

Fix an integer $m_0 \in \mathbb{N}$ such that $2^{m_0+1} \ll 2^d$. For each processor $l = 1, \ldots, L$ we consider the following steps.

[5] For example, if $p(\cdot)$ is selected as a geometric distribution with parameter r, the expected computational cost will be $n(1-r)/r$.

i) Take a random (integer) draw, m_l^*, from a geometric distribution with parameter $r > 1/2$.[6]

ii) Given m_l^*, take $2^{K_l^*+1}$ i.i.d. draws from the d-dimensional vector $\xi_i \sim \mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$, where

$$K_l^* \equiv m_0 + m_l^*.$$

Repeat this step independently for each $i = 1, \ldots, n$.

iii) Solve problem (25) using the first 2^{m_0} i.i.d. draws of ξ_i for each i. Let θ_{l,m_0} denote a minimizer.

iv) Denote by $\widehat{\theta}_n^\star(2^{K_l^*+1})$, $\widehat{\theta}_n^O(2^{K_l^*})$, and $\widehat{\theta}_n^E(2^{K_l^*})$ any solution to the following optimization problems (all of which are based on sample average approximations as (25)):

$$\widehat{\theta}_n^\star(2^{K_l^*+1}) \in \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2^{K_l^*+1}}\sum_{k=1}^{2^{K_l^*+1}} \ell(x_i \odot \xi_i^k, y_i, \theta)\right),$$

$$\widehat{\theta}_n^O(2^{K_l^*}) \in \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2^{K_l^*}}\sum_{k=1}^{2^{K_l^*}} \ell(x_i \odot \xi_i^{2k-1}, y_i, \theta)\right),$$

$$\widehat{\theta}_n^E(2^{K_l^*}) \in \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2^{K_l^*}}\sum_{k=1}^{2^{K_l^*}} \ell(x_i \odot \xi_i^{2k}, y_i, \theta)\right).$$

Intuitively, $\widehat{\theta}_n^O$ and $\widehat{\theta}_n^E$ denote the solutions to problem (25) but using a sample of size 2^{K_l} with only *odd* and *even* indices, respectively.

v) Define

$$\bar{\Delta}_{K_l^*} \equiv \widehat{\theta}_n^\star(2^{K_l^*+1}) - \frac{1}{2}(\widehat{\theta}_n^O(2^{K_l^*}) + \widehat{\theta}_n^E(2^{K_l^*}))$$

and let

$$Z(K_l^*) = \frac{\bar{\Delta}_{K_l^*}}{r(1-r)^{K_l^*-m_0}} + \theta_{l,m_0}.$$

[6] To see why we require that $r > 1/2$, notice if the computational cost of evaluating $Z(K^*)$ (as in the heuristic description above) increases exponentially in K and takes the form $C \cdot 2^K$, the expected computational cost will be

$$\sum_{K=1}^{\infty} Cr(2(1-r))^K = Cr(1/2(1-r)),$$

provided $2(1-r) < 1$, or equivalently, $r > 1/2$. As we show in the proof of Theorem 4, constraining the variance requires then imposing $r < 3/4$. Ultimately, optimizing the product of computational cost and variance leads to the optimal selection $r = 1 - 2^{-3/2}$.

Our recommended estimator is

$$\frac{1}{L}\sum_{l=1}^{L} Z(K_l^*).$$

We now show that the suggested algorithm gives an estimator with desirable properties. We do so under the following regularity assumptions.

Assumption 5 *Suppose that the parameter space Θ is compact. Suppose in addition that the optimal solution $\theta_n^\star$ to the dropout training problem in* (23) *is (globally) unique.*

Assumption 6 *Let $\widehat{\theta}_n^\star(K)$ denote the solution of the problem in* (25) *based on K draws. Suppose that as $K \to \infty$,*

$$\mathbb{E}[\|K^{\frac{1}{2}}(\widehat{\theta}_n^\star(K) - \theta_n^\star)\|_2^4] = O(1),$$

where the expectation is taken over the i.i.d dropout noise distribution used to generate ξ_i^k.

Assumption 7 *Assume that for each (X, Y, ξ), $\ell(X \odot \xi, Y, \cdot)$ is thrice continuously differentiable over Θ and that*

$$\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta_n^\star)]$$

is non-singular.

Theorem 4 *Under Assumption 5, $\mathbb{E}[Z(K_l^*)] = \theta_n^\star$. The number of random draws required to compute $Z(K_l^*)$ is $n \cdot 2^{K_l^*+1}$ and thus the expected computational complexity for producing $Z(K_l^*)$ equals*

$$\frac{n(2^{m_0+1})r}{2r-1} < n(2^{m_0+1}) \ll n2^d.$$

Suppose, in addition, that $\widehat{\theta}_n^\star(K)$ is almost surely in the interior of Θ for K large enough. If Assumptions 6 and 7 hold, and $r < 3/4$. Then $\mathrm{Var}(Z(K_l^*)) < \infty$.

Proof See Appendix A.4. ■

Our suggested algorithm has finite expected computational complexity that does not grow exponentially with the dimension d, thus every time we need to obtain $\widehat{\theta}_n^\star(2^{K_l^\star+1})$, we can do so by applying a gradient descent algorithm. Combined with parallelization, the Unbiased Multi-level Monte Carlo approach produces an unbiased estimator with a variance that can be made arbitrarily small if L is large enough, provided that the regularity assumptions that give $\mathrm{Var}(Z(K_l^*)) < \infty$ are satisfied.

7. Numerical Experiment

We conduct numerical experiments in this section to compare our preferred implementation of dropout training to Stochastic Gradient Descent, as well as our recommended selection of δ to cross-validation. The benefits of our suggested Unbiased Multi-Level Monte Carlo algorithm are analyzed using a high-dimensional regression, whereas our selection of δ is analyzed using a low-dimensional regression model.

7.1 Advantage of the Unbiased Multi-level Monte Carlo Estimator

We present a simple numerical experiment to illustrate the advantage of using the Unbiased Multi-level Monte Carlo estimator suggested in Section 6.4. We consider the linear regression problem with known variance and we focus on solving the dropout training problem with our recommended δ chosen according to Proposition 3.

Our simulation setting considers a linear regression model with covariate vector having dimension $d = 100$ and sample size $n = 50$. We pick a known regression coefficient $\beta_0 \in \mathbb{R}^d$ being a vector with all entries equal to 1. With fixed coefficients, we assume the covariate vector follows independent Gaussian, as well as for the regression noise. More specifically, we can get our $n = 50$ observations (x_i, y_i) via

- sampling $x_i \sim \mathcal{N}(0, I_d)$, $i = 1, \ldots, n$,
- sampling $y_i \in \mathbb{R}$ conditional on x_i, where y_i is given by the linear assumption and ε_i are i.i.d. random noise following $\mathcal{N}(0, 10^2)$, for $i = 1, \ldots, n$.

Our simulation setting considers first a high-dimension setting (relative low ratio between sample size per dimension $n/d = 0.5$) with high noise to signal ratio (variability on residual noise is high compared to the variability on x_i).

If we set $\mathbb{Q}_0$ to be the empirical distribution of $\{(x_i, y_i)_{i=1}^n\}$, the dropout training problem in the linear regression model is

$$\min_{\beta \in \mathbb{R}^d} \mathbb{E}_{\mathbb{Q}^\star}\left[\left(\beta^\top (X \odot \xi) - Y\right)^2\right].$$

Corollary 7 in Appendix A.5 shows that in the linear regression model the dropout training problem can be written as

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n}\left[(\mathbf{Y} - \mathbf{X}\beta)^\top (\mathbf{Y} - \mathbf{X}\beta) + \frac{\delta}{1-\delta}\beta^\top \mathbf{\Lambda} \beta\right],$$

where $\mathbf{Y} = [y_1, y_2, \ldots, y_n]^\top$, $\mathbf{X} = [x_1, x_2, \ldots, x_n]^\top$ and $\mathbf{\Lambda}$ is the diagonal matrix with its diagonal elements given by the diagonals of $\mathbf{X}^\top \mathbf{X}$. Moreover, there is a closed-form solution for the dropout training problem and it is given by the ridge regression formula:

$$\beta_n^\star = \left(\mathbf{X}^\top \mathbf{X} + \frac{\delta}{1-\delta} \mathbf{\Lambda} \right)^{-1} \mathbf{X}^\top \mathbf{Y}.$$

We choose the dropout probability δ following Proposition 3. More specifically, Proposition 3 suggests the choice $\delta = c/\sqrt{n}$ where $c = z_{1-\alpha} \cdot \sigma/\mu$. For linear regression with known variance, it is straightforward to compute

$$\mu = \frac{1}{2\phi^\star} \sum_{j=1}^{d} \mathbb{E}_{P^\star}[X_j^2](\beta_j^\star)^2,$$

and

$$\sigma^2 = \mathbb{V}\mathrm{ar}_{P^\star} \left[\frac{1}{2} \log(2\pi\phi^\star) + \frac{(Y - (\beta^\star)^\top X)^2}{2\phi^\star} \right].$$

Choosing $\alpha = 0.1$ and note that $\beta^\star = \beta_0, \phi^\star = 10^2$, we have $\delta \approx 0.26$.

Since neither our suggested Multi-level Monte Carlo algorithm nor standard SGD (as defined in Section 6.2) uses closed-form formulae for their implementation, we analyze the extent to which these procedures can approximate the parameter $\beta_n^\star$. We provide more details of the algorithms as follows. The two algorithms we compare are:

- Standard SGD algorithm with a learning rate 0.0001, and initialization at the origin. Note that however we take batched SGD instead of single-sample SGD introduced in Section 6.2.

- Multi-level Monte Carlo algorithm with the geometric rate $r = 0.6$ and the burn-in period $m_0 = 5$. Note that in each parallel running, we use gradient descent (GD) with 0.01 learning rate and initialization at origin for steps iii) and iv) in Section 6.4.

We run our simulation on a cluster with two Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz processors (with 10 cores each), and a total memory of 128 GB. We fix 60 seconds as a "wall-clock time", so that we terminate the two algorithms after 60 seconds.[7]

We run 1000 independent experiments. For each run, we calculate and report the average parameter estimation divergence to $\beta_n^\star$ and 1-standard deviation error bar for the divergence. We consider difference number of parallelizations (i.e., L in Section 6.4) from 400 to 2400. We cap the run at 2400 due to the saturation of divergence after $\sim$ 2000 parallelizations.

[7] The parameters for the SGD algorithm are appropriately tuned to achieve good convergence within 60s (see Appendix A.6 for the tuning procedure). However, we do not claim that this choice of parameters is optimal.

Figure 1 shows the l_2 divergence to the true $\beta_n^\star$ of the two algorithms for varying L, while Figure 2 and Figure 3 show l_∞ and l_1 divergence, respectively. We observed that our unbiased estimator outperforms standard SGD algorithm once the number of parallel iterations reaches above some moderate threshold ($\sim$ 1000 here). We provide supporting evidence in Appendix A.6 to argue our choice of learning rate, initialization, and wall-clock time, where our proposed algorithm is robust to any reasonable choices.

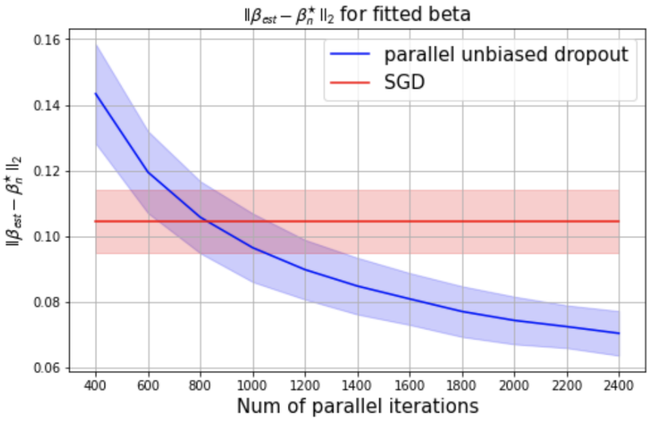

Figure 1: l_2 difference

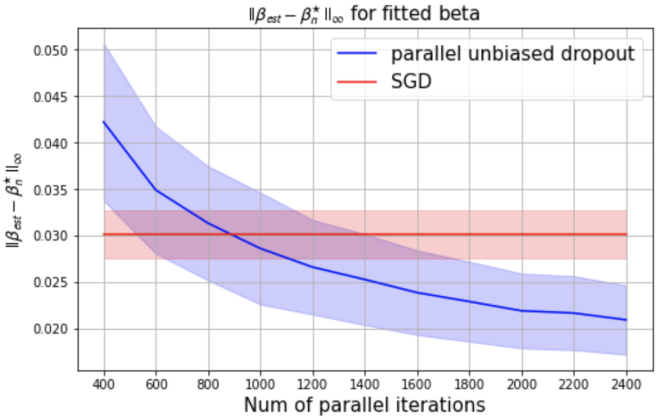

Figure 2: l_∞ difference

7.2 Coverage of the True Loss of Dropout Training

We validate that our recommended selection of δ guarantees that the in-sample loss of dropout training is covering the true loss with arbitrary high probability as prescribed by Proposition 3.

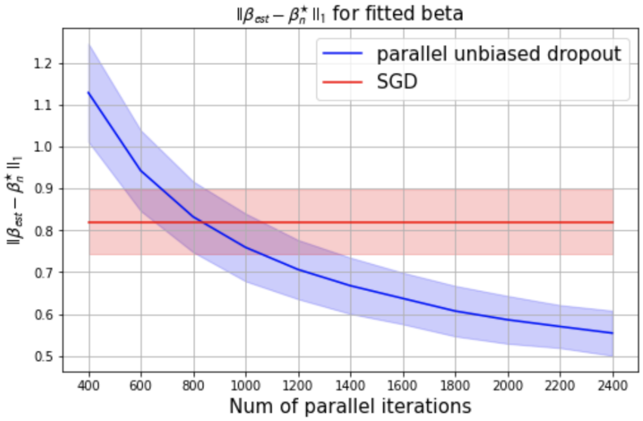

Figure 3: l_1 difference

We use the same linear regression model with dimension $d = 10$ and training samples $n \in \{10^3, 10^4\}$. We choose different quantiles of the normal as in Proposition 3. We also include 10-fold cross validation and ordinary least squares for comparison. See Table 1, where we estimate the frequency of coverage over 1000 independent runnings. The main message is that our suggested choice of δ guarantees that the in-sample loss of dropout training exceeds the true, unknown, population loss with probability $1 - \alpha$. Using standard OLS or choosing δ by cross-validation the in-sample loss is smaller than the population loss with probability close to 1/2, which implies that these methods are unsatisfactory in terms of frequency of coverage.

	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	10-fold CV	plain OLS
$n = 10^3$	0.77 ± 0.01	0.88 ± 0.01	0.94 ± 0.01	0.52 ± 0.02	0.40 ± 0.01
$n = 10^4$	0.79 ± 0.01	0.90 ± 0.01	0.94 ± 0.01	0.49 ± 0.02	0.47 ± 0.02

Table 1: Frequency of in-sample loss covering the true population loss. Our recommended selection of $\delta = c/\sqrt{n}$ with $c = z_{1-\alpha}\sigma/\mu$ has a theoretical $1-\alpha$ coverage probability.

8. Extensions

In this section we discuss the extent to which the decision-theoretic support for dropout training carries over to Neural Networks. The main idea is that we use a GLM model where the natural parameter is no longer a linear function of the covariates, but instead a neural network.

8.1 One-hidden-layer Feed-Forward Neural Networks

Suppose the scalar response variable Y is generated by the conditional density

$$f(Y|X,\theta,\phi) \equiv h(Y,\phi)\exp\left(\left(Y\Omega_\theta(X)\right) - \Psi(\Omega_\theta(X))\right)/a(\phi)\right), \tag{26}$$

where $\Omega_\theta(X)$ is a neural network with parameters θ and $X \in \mathbb{R}^d$. This is a simple extension of the regression model that has been used recently to study deep neural networks; see Schmidt-Hieber (2020) in which the conditional density is Gaussian.

In this section, we will assume that $\Omega_\theta(X)$ is a neural network with a *single hidden layer*, a *differentiable activation (squashing) function*, and *linear ouput function.* A function $h : \mathbb{R} \to [0,1]$ is a squashing function if it is non-decreasing and if

$$\lim_{r\to\infty} h(r) = 1, \quad \lim_{r\to-\infty} h(r) = 0.$$

See Definition 2.3 in Hornik et al. (1989).

Although these types of networks—which will be formally described below—are restrictive compared to the modern deep learning architectures, they can approximate any Borel measurable function from a finite-dimensional space to another, provided the hidden units in the hidden layer are large; see Hornik et al. (1989).

Consider a neural network with K units in the hidden layer, each using input weights $w_k \in \mathbb{R}^d$, $k = 1, \ldots, K$. Denote the activation function in the hidden layer as $h(\cdot)$. Assume the output function is linear with vector of weights $\beta \in \mathbb{R}^K$. Thus, the network under consideration is defined by the function:

$$\Omega_\theta(X) \equiv \beta_1 h(w_1^\top X) + \ldots + \beta_K h(w_K^\top X) = \beta^\top H(X),$$

where $H(X) = (h(w_1^\top X), \ldots, h(w_K^\top X))^\top$. The neural network is parameterized by $\theta \equiv (\beta^\top, w_1^\top, \ldots, w_k^\top)^\top$. Under this model, the distribution of $Y|X$ is a GLM model with covariates $H(X)$.

8.1.1 Statistician's Objective Function

We will endow the statistician with the loss function given by the negative of the conditional log-likelihood for the model in (26).

8.1.2 Nature's Uncertainty Set

We allow nature to introduce additional noise to the statistician's model. We do this in two steps. First, we allow nature to distort the distribution of X using a multiplicative noise denoted as $\xi(1) \in \mathbb{R}^d$. This is exactly analogous to what we did in the GLM model, where nature was allowed to pick a distribution for the covariates of the form $(X \odot \xi(1))$. Using the jargon of neural networks, we allow nature to contaminate the *input layer* with independent and multiplicative noise. Second, we also allow nature to contaminate *each of the hidden units* with multiplicative noise $\xi(2) \in \mathbb{R}^K$. That is, nature is also allowed to pick a vector $\xi(2) = (\xi(2)_1, \ldots, \xi(2)_K)^\top$, independently of $\xi(1) \in \mathbb{R}^d$, to distort the each of the K units in the hidden layer as

$$H(X) \odot \xi(2) \equiv (h(w_1^\top X)\xi(2)_1, \ldots, h(w_K^\top X)\xi(2)_K)^\top.$$

Our choice of a one-hidden neural network was simply for expositional simplicity, but the analysis would be the same with a feed-forward neural network with L hidden layers.

8.1.3 Minimax Solution

The minimax solution of the DRO game is given by

$$\inf_{\theta} \sup_{\mathbb{Q}} \mathbb{E}_{\mathbb{Q}} \left[-\ln f(Y | H(X \odot \xi_1) \odot \xi_2, \beta, \phi) \right], \tag{27}$$

where $\mathbb{Q}$ now refers to the joint distribution of $(X, Y, \xi(1), \xi(2))$ and $f(Y|X, \beta, \phi)$ is the GLM density defined in (1). We continue working with the assumption that $\xi \equiv (\xi(1)^\top, \xi(2)^\top)^\top$ has independent marginals and that it is independent of (X, Y).

We would like to solve for the worst-case distributions of the random vectors $\xi(1)$ and $\xi(2)$, assuming that both of these satisfy the restrictions analogous to (12). The solution for the distribution of $\xi(2)$ can be obtained as a corollary to Theorem 2, as it suffices to define

$$\tilde{X} \equiv H(X \odot \xi(1)),$$

and view (27) as the DRO problem in a linear regression model, in which the data is $(\tilde{X}, Y)$ and $\xi(2) \in \mathbb{R}^K$ is simply the multiplicative noise that transforms the covariates into $(\tilde{X} \odot \xi(2))$.

The worst-case choice of $\xi(1)$, the multiplicative error for the inputs, is more difficult to characterize and we were not able to find general results for it. Below, we provide a heuristic argument suggesting that dropout noise might approximate the worst-case choice when the output layer is a Gaussian linear model. Let $\xi(1)_j$ denote the j-th coordinate of $\xi(1)$. Suppose that the distribution of this random variable places most of its mass on the interval $[1 - \epsilon, 1 + \epsilon]$.[8] This allows us to 'linearize' the output of each of the hidden units

[8] This is compatible with dropout noise for which δ is very close to zero.

around the output corresponding to unperturbed inputs as

$$\begin{aligned} h(w_k^\top(X \odot \xi(1))) &= h(w_k^\top(X \odot (\xi(1) - \mathbf{1})) + w_k^\top X) \\ &\approx h(w_k^\top X) + \Big(\dot{h}(w_k^\top X) \cdot (w_k \odot X)^\top (\xi(1) - \mathbf{1})\Big) . \end{aligned}$$

In the notation above, $\mathbf{1}$ denotes the d-dimensional vector of ones. For the sake of exposition, ignore the approximation error in the linearization above. If we fix $(X, Y, \xi(2))$, then the worst-case choice for the distribution of $\xi(1)$, denoted by $\mathbb{Q}(1)$, maximizes

$$\mathbb{E}_{\mathbb{Q}(1)}\left[\left(\sum_{k=1}^{K} \beta_k \cdot \xi(2)_k \cdot \left[h(w_k^\top X) + \left(\dot{h}(w_k^\top X) \cdot \sum_{j=1}^{d} w_{k,j} \cdot X_j \cdot (\xi(1)_j - 1)\right)\right]\right)^2\right]$$

among all distributions with independent marginals for which $\mathbb{E}_{\mathbb{Q}(1)}[\xi(1)_j] = 1$ for all $j = 1, \ldots, d$. Algebra shows that such maximization problem is equivalent to maximizing

$$\mathbb{E}_{\mathbb{Q}(1)}\left[\left(\sum_{k=1}^{K} \beta_k \cdot \xi(2)_k \cdot \dot{h}(w_k^\top X) \cdot \left[\sum_{j=1}^{d} w_{k,j} \cdot X_j \cdot (\xi(1)_j - 1)\right]\right)^2\right], \tag{28}$$

which in turn can be written as

$$\mathbb{E}_{\mathbb{Q}(1)}\left[\left(a^\top(\xi(1) - \mathbf{1})\right)^2\right]$$

for an appropriate choice of a vector $a \in \mathbb{R}^d$ that depends only on $(\beta, \xi(2), h, \dot{h}, w, X)$. Proposition 6 in Appendix A.2 shows that the solution to this problem is dropout noise.

9. Concluding Remarks

In this paper we studied *dropout training*, an increasingly popular estimation method in machine learning. Dropout training is a fundamental part of the modern machine learning techniques for training very deep networks (Goodfellow et al., 2016).

Our main result (Theorem 2) established a novel decision-theoretic foundation for the use of dropout training. We showed that this method, when applied to Generalized Linear Models, can be viewed as the minimax solution to an adversarial two-player, zero-sum game between a statistician and nature. The framework used in this paper is known in the stochastic optimization literature (Shapiro et al., 2014) as a Distributionally Robust Optimization (DRO) problem.

Our minimaxity result showed, by construction, that dropout training indeed provides out-of-sample performance guarantees for distributions that arise from multiplicative perturbations of the in-sample data. Our result thus justified explicitly the ability of dropout training to enhance the out-of-sample performance, which is one of the reasons often invoked to promote the dropout method.

In addition to our theoretical result, we also suggested a new strategy to select the dropout probability and a new stochastic optimization implementation of dropout training. For the latter, we borrowed ideas from the Multi-level Monte Carlo literature—in particular from the work of (Blanchet et al., 2019a)—to suggest an unbiased dropout training routine that is easily parallelizable and that has a smaller computational cost compared to naive dropout training methods when the number of features is large (Theorem 4). Crucially, we showed that under some regularity conditions our estimator has finite variance (which means there are also theoretical, and not just practical, gains from parallelization).

We also discussed the extent to which our theoretical results extended to Neural Networks (in particular, to the universal approximators in (Hornik et al., 1989) consisting of a single-hidden layer and a squashing activation function). Our results showed that Theorem 2 can be used to establish the optimality of dropout training to estimate the parameters of the last hidden layer in general feed-forward neural networks, where the output layer takes the form of a Generalized Linear Model. We hope that our analysis serves as a foundation to understand the benefits of dropout training in Neural Networks.

Acknowledgments

We would like to thank Matias Cattaneo, Max Farrell, Michael Leung, Ulrich Müller, Mark Peletier, Hashem Pesaran, Ashesh Rambachan, Roger Moon, Frank Schorfheide, Stefan Wager, and participants at the Statistics Seminar series at Columbia University for helpful comments and suggestions. José Blanchet acknowledges support from NSF grants 1915967, 1820942, 1838576 and the Chinese Merchant Bank. Material in this paper is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0397.

Appendix A.

A.1 Proof of Proposition 1

Algebra shows that the dropout estimator of β maximizes

$$Q_n(\beta) \equiv \frac{1}{n}\sum_{i=1}^{n} y_i(\beta^\top x_i) - \mathbb{E}_{\delta_n}[\Psi(\beta^\top(x_i \odot \xi))].$$

In a slight abuse of notation, let ξ_δ denote a realization of dropout noise parameterized by δ. Then it is possible to re-write the objective function as a weighted average of the functions

$$Q_{n,\xi_{\delta_n}}(\beta) \equiv \frac{1}{n}\sum_{i=1}^{n} y_i(\beta^\top x_i) - \Psi(\beta^\top(x_i \odot \xi_{\delta_n})).$$

It will be convenient then to define the limiting objective function to be

$$Q(\beta) \equiv \mathbb{E}_{P^\star}[Y(\beta^\top X)] - \mathbb{E}_{P^\star}[\mathbb{E}_\delta[\Psi(\beta^\top(X \odot \xi))]],$$

which, by Assumptions 1 and 2, is finite and strictly concave. The population objective function is then the average (over dropout noise) of

$$Q_{\xi_\delta}(\beta) \equiv \mathbb{E}_{P^\star}[Y(\beta^\top X)] - \mathbb{E}_{P^\star}[\Psi(\beta^\top(X \odot \xi_\delta))].$$

It is straightforward to show that $\beta^*(\delta)$ in (6) denote the unique maximizer of $Q(\beta)$.

Proof The proof follows from standard arguments in the theory of extremum estimators. In particular, it suffices to verify the conditions of Theorem 2.7 in Newey and McFadden (1994).

Condition i) in Newey and McFadden (1994) requires $Q(\beta)$ to be uniquely maximized at $\beta^*(\delta)$. This holds because Assumptions 1 and 2 imply that $Q(\beta)$ is strictly concave.

Condition ii) in Newey and McFadden (1994) requires $\beta^*(\delta)$ to be an element in the interior of a strictly convex set, which holds because in the GLM models under consideration the parameter space is $\mathbb{R}^d$. Furthermore, $Q_n(\beta)$ is trivially concave by Assumption 1.

Condition iii) requires $Q_n(\beta)$ to converge in probability to $Q(\beta)$ for every β. For this purpose, it suffices to show that $Q_{n,\xi_{\delta_n}}(\beta)$ converges in probability to $Q_{\xi_\delta}(\beta)$ for each fixed β, and for a sequence ξ_{δ_n} and ξ_δ that have zeros and non-zeros in exactly the same entries. Assumptions 1 and 2 imply $\mathbb{E}_{P^*}[Y(\beta^\top X)] < \infty$ for all β. Thus, using the Law of Large

Numbers for i.i.d sequences

$$\frac{1}{n}\sum_{i=1}^{n} Y_i(\beta^\top X_i) \xrightarrow{p} \mathbb{E}_{P^*}[Y(\beta^\top X)].$$

Finally, Assumptions 1 and 2 imply that the triangular array

$$Z_{n,i} = \Psi(\beta^\top (X_i \odot \xi_{\delta_n})), \quad 1 \leq i \leq n,$$

satisfies the conditions for the Law of Large Numbers for triangular arrays (Theorem 2.2.11 in Durrett (2019)), and consequently

$$\frac{1}{n}\sum_{i=1}^{n} \Psi(\beta^\top (X_i \odot \xi_{\delta_n})) \xrightarrow{p} \mathbb{E}_{P^*}\left[\Psi(\beta^\top (X \odot \xi_{\delta}))\right].$$

This completes the proof. ■

A.2 Proof of Theorem 2

The proof of Theorem 2 relies on the following two preparatory results.

Lemma 5 (Extremal expectation of a univariate convex function) *For any $-\infty < a < b < +\infty$, let ζ be a random variable in $[a, b]$ with mean $\mu \in [a, b]$. For any function $f : [a, b] \to \mathbb{R}$ convex and continuous, the distribution of ζ that maximizes $\mathbb{E}[f(\zeta)]$ among all distributions over $[a, b]$ with a given mean $\mu \in [a, b]$ is a scaled and shifted Bernoulli distribution, i.e.,*

$$\zeta = \begin{cases} a & \textit{with probability } (b-\mu)/(b-a), \\ b & \textit{with probability } (\mu-a)/(b-a). \end{cases} \tag{29}$$

Proof Let Q^* denote the probability measure induced by the random variable in (29). By definition

$$\mathbb{E}_{Q^*}[f(\zeta)] = \frac{b-\mu}{b-a} f(a) + \frac{\mu-a}{b-a} f(b).$$

Suppose first that $\mu = a$. In this case, Jensen's inequality implies that for any other probability measure Q over $[a, b]$ with mean $\mu = a$,

$$\mathbb{E}_Q[f(\zeta)] \leq f(\mathbb{E}_Q[\zeta]) = f(a) = \mathbb{E}_{Q^*}[f(\zeta)].$$

An analogous result holds if $\mu = b$.

Consider then the case in which $\mu \in (a,b)$. For an arbitrary probability measure Q over $[a,b]$ with mean $\mu \in (a,b)$, we have

$$\int_{[a,b]} f(\zeta)\mathrm{d}Q = \int_{[a,b]} f\big(a\frac{b-\zeta}{b-a} + b\frac{\zeta-a}{b-a}\big)\mathrm{d}Q \le \int_{[a,b]} \big(\frac{b-\zeta}{b-a}f(a) + \frac{\zeta-a}{b-a}f(b)\big)\mathrm{d}Q,$$

where the inequality follows from the convexity of f. By the linearity of the integral operator and the fact that $\int_{[a,b]} \zeta \mathrm{d}Q = \mu$, we find

$$\int_{[a,b]} f(\zeta)\mathrm{d}Q \le \frac{b-\mu}{b-a}f(a) + \frac{\mu-a}{b-a}f(b).$$

Because the probability measure Q was chosen arbitrarily, this implies that the distribution of ζ in (29) maximizes the expectation of $f(\zeta)$. ■

Proposition 6 *Fix a vector of tuning parameters $\delta \in (0,1)^d$. Let $\mathcal{Q}_j(\delta_j)$ be defined as in* (11)*. Suppose that A is a convex and continuous function on $\mathbb{R}$. For any $\theta \in \mathbb{R}^d$, we have*

$$\sup\Big\{\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A(\theta^\top\xi)] : \mathbb{Q}_j \in \mathcal{Q}_j(\delta_j)\Big\} = \mathbb{E}_{\mathbb{Q}_1^\star\otimes\ldots\otimes\mathbb{Q}_d^\star}[A(\theta^\top\xi)],$$

where $\mathbb{Q}_j^\star$ is a scaled Bernoulli distribution of the form $\mathbb{Q}_j^\star = (1-\delta_j)^{-1} \times \mathit{Bernoulli}((1-\delta_j))$ for each $j = 1,\ldots,d$.

Proof First note that $\mathbb{Q}_j^\star \in \mathcal{Q}_j(\delta_j)$ for each j, and thus $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$ is a feasible solution to the maximization problem. It suffices to show that for any set of feasible measures $\mathbb{Q}_j \in \mathcal{Q}_j(\delta_j), j = 1,\ldots,d$, we have

$$\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A(\theta^\top\xi)] \le \mathbb{E}_{\mathbb{Q}_1^\star\otimes\ldots\otimes\mathbb{Q}_d^\star}[A(\theta^\top\xi)].$$

Towards this end, pick any $k \in \{1,\ldots,d\}$. By Fubini's theorem, we can write

$$\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A(\theta^\top\xi)] = \mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_{k-1}\otimes\mathbb{Q}_{k+1}\otimes\ldots\otimes\mathbb{Q}_d}\mathbb{E}_{\mathbb{Q}_k}[A(\theta^\top\xi)].$$

For any fixed value $(\xi_1,\ldots,\xi_{k-1},\xi_{k+1},\ldots,\xi_d)$ the function $\xi_k \mapsto A(\sum_{j\neq k}\theta_j\xi_j + \theta_k\xi_k)$ is convex in the variable ξ_k over the interval $[0,(1-\delta_k)^{-1}]$. Thus by Lemma 5,

$$\mathbb{E}_{\mathbb{Q}_k}[A(\sum_{j\neq k}\theta_j\xi_j + \theta_k\xi_k)] \le \mathbb{E}_{\mathbb{Q}_k^\star}[A(\sum_{j\neq k}\theta_j\xi_j + \theta_k\xi_k)] \quad \text{for any fixed } (\xi_1,\ldots,\xi_{k-1},\xi_{k+1},\ldots,\xi_d).$$

Thus by the monotonicity of the expectation operator,

$$\mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)] \leq \mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_{k-1} \otimes \mathbb{Q}_{k+1} \otimes \ldots \otimes \mathbb{Q}_d} \mathbb{E}_{\mathbb{Q}_k^\star}[A(\theta^\top \xi)] = \mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_{k-1} \otimes \mathbb{Q}_k^\star \otimes \mathbb{Q}_{k+1} \otimes \ldots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)].$$

By cycling through all possible values of $k \in \{1, \ldots, d\}$ we conclude that

$$\mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)] \leq \mathbb{E}_{\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star}[A(\theta^\top \xi)].$$

Therefore, the postulated claim holds. ■

We are now ready to prove Theorem 2.

Proof Note that for $\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)$, Assumption 3 implies $\mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)]$ is finite for any $\theta \in \Theta$ and any scalar $\delta \in [0, 1)$. Therefore, from Fubini's theorem and the definition of loss function:

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)] &= \mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d}[\ell(X \odot \xi, Y, \theta)]\right] \\
&= \mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d}[-\ln h(Y, \phi) + (\Psi(\beta^\top (X \odot \xi)) - Y(\beta^\top (X \odot \xi)))/a(\phi)]\right] \\
&= -\mathbb{E}_{\mathbb{Q}_0}\left[\ln h(Y, \phi)\right] \\
&+ \mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d}[(\Psi(\beta^\top (X \odot \xi)) - Y(\beta^\top (X \odot \xi)))/a(\phi)]\right].
\end{aligned}$$

Algebra shows that for any β, X and ξ:

$$\beta^\top (X \odot \xi) = (\beta \odot X)^\top \xi.$$

Thus, we can fix the values of (X, Y, θ) and define the function

$$A_{(X,Y,\theta)}((\beta \odot X)^\top \xi) \equiv (\Psi(\beta^\top (X \odot \xi)) - Y\beta^\top (X \odot \xi))/a(\phi).$$

Note that $A_{(X,Y,\theta)}$ satisfies the condition of Proposition 6. Therefore

$$\sup\left\{\mathbb{E}_{\mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d}[A_{(X,Y,\theta)}((\beta \odot X)^\top \xi)] : \mathbb{Q}_j \in \mathcal{Q}_j(\delta_j)\right\} = \mathbb{E}_{\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star}[A_{(X,Y,\theta)}((\beta \odot X)^\top \xi)],$$

for any (X, Y, θ), which completes the proof. ■

A.3 Proof of Proposition 3

Proof We write $\sqrt{n}(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}, \delta_n) - \mathcal{L}(\beta^*, \phi^*))$ as the sum of the following three terms

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}, \delta_n) - \mathcal{L}_n(\beta^*, \widehat{\phi}, \delta_n)\right), \tag{30a}$$

$$\sqrt{n}\left(\mathcal{L}_n(\beta^*, \widehat{\phi}, \delta_n) - \mathcal{L}_n(\beta^*, \widehat{\phi}, 0)\right), \tag{30b}$$

$$\sqrt{n}\left(\mathcal{L}_n(\beta^*, \widehat{\phi}, 0) - \mathcal{L}(\beta^*, \phi^*, 0)\right). \tag{30c}$$

The last term converges in distribution to a normal random variable, so we only need to analyze (30a) and (30b).

By Assumptions 1 and 2 the term in (30a) admits an exact second-order Taylor expansion around β^* for every ϕ and δ. We argue that because of this, the term in question if $o_p(1)$. First, using the same arguments as in Theorem 3.1 in Newey and McFadden (1994) we can show that for any sequence $\delta_n = c/\sqrt{n}$

$$\sqrt{n}(\widehat{\beta}(\delta_n) - \beta^\star) \xrightarrow{d} \Sigma(\beta^*)^{-1}\mathcal{N}_d(-c\tilde{\mu}, a(\phi^*)\Sigma(\beta^*)),$$

where

$$\Sigma(\beta) \equiv \mathbb{E}_{P^*}[\ddot{\Psi}(X^\top \beta)XX^\top],$$

and

$$\tilde{\mu} \equiv \left(\sum_{\xi \in \mathcal{A}} \mathbb{E}_{P^\star}[\dot{\Psi}((X \odot \xi)^\top \beta^\star)(X \odot \xi)]\right) - (d-1)\mathbb{E}_{P^\star}[YX] + \Sigma(\beta^*)\beta^*.$$

The set $\mathcal{A}$ above is defined as $\{\xi \in \{0,1\}^d : \text{exactly one entry of } \xi \text{ is zero}\}$. The argument is essentially the same as in every proof of asymptotic normality for extremum (or M-estimators), with the only difference being that, because of the dropout noise, the score term is asymptotically normal with a nonzero mean. In fact,

$$\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}, \delta_n) \equiv \nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}, 0) + \frac{1}{a(\widehat{\phi})}\left(\frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\delta_n}[(X_i \odot \xi)\dot{\Psi}(\beta^\top (X_i \odot \xi))] - X_i \dot{\Psi}(X_i^\top \beta)\right),$$

where

$$\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}, 0) \equiv -\frac{1}{a(\widehat{\phi})}\frac{1}{n}\sum_{i=1}^n X_i(Y_i - \dot{\Psi}(X_i^\top \beta)).$$

Recognizing the term $\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}, 0)$ as the negative of the score function in the GLM model and doing some algebra, it is possible to show that $\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}, \delta_n)$ is $o_p(1)$.

For the term in (30b), note first that it is nonnegative. Also: $\mathcal{L}_n(\beta^*, \widehat{\phi}, \delta_n) - \mathcal{L}_n(\beta^*, \widehat{\phi}, 0)$ equals

$$\frac{1}{a(\widehat{\phi})}\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbb{E}_{\delta_n}[\Psi((X_i \odot \xi)^\top \beta^*)] - \Psi(X_i^\top \beta^*)\right).$$

The term in parenthesis has finite mean equal to

$$\Delta_n \equiv \mathbb{E}_{P^*}\mathbb{E}_{\delta_n}[\Psi((X \odot \xi)^\top \beta^*)] - \mathbb{E}_{P^*}[\Psi(X^\top \beta^*)]. \tag{31}$$

It can be shown—by verifying the conditions for the Law of Large Numbers for triangular arrays (Theorem 2.2.11 in Durrett (2019))—that

$$\sqrt{n}(\mathcal{L}_n(\beta^*, \delta_n) - \mathcal{L}_n(\beta^*, 0) - a(\widehat{\phi})^{-1}\Delta_n) \xrightarrow{p} 0.$$

Moreover, Assumptions 1 and 2 imply

$$\sqrt{n}\Delta_n \xrightarrow{p} \Delta,$$

where

$$\Delta \equiv c\left(\sum_{\xi \in \mathcal{A}} \mathbb{E}_{P^\star}[\Psi((X \odot \xi)^\top \beta^\star)] - d\mathbb{E}_{P^\star}[\Psi(X^\top \beta^\star)] + \mathbb{E}_{P^\star}[\dot{\Psi}(X^\top \beta^*) X^\top \beta^\star]\right).$$

This gives the desired result.

■

A.4 Proof of Theorem 4

Proof By definition

$$Z(K_l^*) = \frac{\bar{\Delta}_{K_l^*}}{r(1-r)^{m_l^*}} + \theta_{l,m_0},$$

where K_l^* is a discrete random variable with probability mass function:

$$p(K_l^*) = r(1-r)^{K_l^* - m_0},$$

and supported on the integers larger than m_0.

We first show that the estimator $Z(K_l^*)$ is unbiased (as we average over both K_l^* and ξ_i^k). Algebra shows that

$$\begin{aligned}
\mathbb{E}[Z(K_l^*)] &= \sum_{K=m_0}^{\infty} \mathbb{E}[Z(K_l^*)|K_l^* = K]p(K) \\
&= \sum_{K=m_0}^{\infty} \mathbb{E}\left[\frac{\bar{\Delta}_{K_l^*}}{p(K_l^*)} + \theta_{l,m_0}\middle| K_l^* = K\right] p(K) \\
&= \sum_{K=m_0}^{\infty} \mathbb{E}\left[\frac{\bar{\Delta}_{K}}{p(K)} + \theta_{l,m_0}\middle| K_l^* = K\right] p(K) \\
&= \left(\sum_{K=m_0}^{\infty} \mathbb{E}\left[\widehat{\theta}_n^\star(2^{K+1}) - \frac{1}{2}(\widehat{\theta}_n^O(2^K) + \widehat{\theta}_n^E(2^K))\right]\right) + \mathbb{E}[\theta_{l,m_0}] \\
&= -\frac{1}{2}\left(\mathbb{E}[\widehat{\theta}_n^O(2^{m_0})] + \mathbb{E}[\widehat{\theta}_n^E(2^{m_0})]\right) + \mathbb{E}[\theta_{l,m_0}] + \lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^\star(2^{K+1})].
\end{aligned}$$

The expectations in the last line are all finite because Θ is compact. In addition, since the draws are i.i.d. and θ_{l,m_0} is the solution to the problem (25) when 2^{m_0} draws are used we have

$$-\frac{1}{2}\left(\mathbb{E}[\widehat{\theta}_n^O(2_0^m)] + \mathbb{E}[\widehat{\theta}_n^E(2_0^m)]\right) + \mathbb{E}[\theta_{l,m_0}] = 0.$$

Moreover, the sequence of random variables

$$\{\widehat{\theta}_n^\star(2^{K+1})\}$$

is uniformly integrable, because Θ is a compact subset of a finite-dimensional Euclidean space. Finally, we know that

$$\widehat{\theta}_n^\star(2^{K+1}) \xrightarrow{p} \theta_n^\star$$

as $K \to \infty$. The uniform integrability of the sequence of estimators then implies

$$\lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^\star(2^{K+1})] = \mathbb{E}\left[\lim_{K\to\infty} \widehat{\theta}_n^\star(2^{K+1})\right] = \theta_n^\star,$$

see Theorem 6.2 in DasGupta (2008). We conclude that

$$\mathbb{E}[Z(K_l^*)] = \lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^\star(2^{K+1})] = \theta_n^\star.$$

Now we show that the expected computational cost of $Z(K_l^*)$ is finite. In order to compute $Z(K)$ for a given K we need $n \cdot 2^{K+1}$ random draws. Thus, the expected computational cost of $Z(K_l^*)$ is

$$\begin{aligned}\sum_{K=m_0}^{\infty} n2^{K+1}r(1-r)^{K-m_0} &= n\cdot(2^{m_0+1})\cdot r\sum_{K=m_0}^{\infty} 2^{K-m_0}(1-r)^{K-m_0} \\ &= n\cdot(2^{m_0+1})\cdot r\sum_{K=m_0}^{\infty}(2(1-r))^{K-m_0}.\end{aligned}$$

The term above converges to

$$\frac{n\cdot(2^{m_0+1})\cdot r}{1-2(1-r)} = \frac{n\cdot(2^{m_0+1})\cdot r}{2r-1}$$

provided that $2(1-r)<1$, which holds because we have chosen $r>1/2$.

For the proof on finite variance, we intend to show that

$$\mathbb{E}\left[\bar{\Delta}_K^{\top}\bar{\Delta}_K\right] = O(2^{-2K}) \tag{32}$$

as $K\to\infty$. Equation (32) guarantees that every processor generates an estimator $Z(K_l^*)$ with finite variance. Since K_l^* is a discrete random variable with probability mass function

$$p(K_l^*) = r(1-r)^{K^*-m_0},$$

and

$$\begin{aligned}\mathbb{E}[Z(K_l^*)^{\top}Z(K_l^*)] &= \sum_{K=m_0}^{\infty}\mathbb{E}\left[Z(K_l^*)^{\top}Z(K_l^*)|K_l^*=K\right]p(K) \\ &= \sum_{K=m_0}^{\infty}\mathbb{E}\left[\left(\frac{\bar{\Delta}_K}{p(K)}+\theta_{l,m_0}\right)^{\top}\left(\frac{\bar{\Delta}_K}{p(K)}+\theta_{l,m_0}\right)\right]p(K) \\ &\le 2\left(\sum_{K=m_0}^{\infty}\mathbb{E}\left[\frac{\bar{\Delta}_K^{\top}\bar{\Delta}_K}{p(K)^2}\right]p(K)+\sum_{K=m_0}^{\infty}\mathbb{E}\left[\theta_{l,m_0}^{\top}\theta_{l,m_0}\right]p(K)\right) \\ &\le C\left(\sum_{K=m_0}^{\infty}\frac{2^{-2K}}{p(K)}+\sup_{\theta\in\Theta}\|\theta\|_2^2 p(K)\right) \\ &\le C\left(\sum_{K=m_0}^{\infty}\frac{1}{2^{2m_0}2^{2(K-m_0)}p(K)}+\sup_{\theta\in\Theta}\|\theta\|_2^2 p(K)\right) \\ &\le C_1\left(\sum_{K=m_0}^{\infty}\frac{1}{r4^{m_0}}\frac{1}{(4(1-r))^{K-m_0}}\right)+C_2.\end{aligned}$$

The geometric sum in the last expression is finite because we have assumed that $r < \frac{3}{4}$.

To show (32), we do a Taylor expansion of the first-order conditions of the problem (25) around $\theta_n^\star$. The Karush-Kuhn-Tucker optimality condition for the level 2^K solution $\widehat{\theta}_n^\star(2^K)$ of the problem in (25) implies

$$0 = \sum_{i=1}^n \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_\theta \ell(x_i \odot \xi_i^k, y_i, \widehat{\theta}_n^\star(2^K)) \right].$$

It follows by the Taylor expansion and Assumption 4 that

$$\begin{aligned} 0 &= \sum_{i=1}^n \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_\theta \ell(x_i \odot \xi_i^k, y_i, \theta_n^\star) \right] + \sum_{i=1}^n \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_{\theta\theta} \ell(x_i \odot \xi_i^k, y_i, \theta_n^\star) \right] \left(\widehat{\theta}_n^\star(2^K) - \theta_n^\star \right) \\ &\quad + R_{K,\theta} \\ &= \sum_{i=1}^n \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_\theta \ell(x_i \odot \xi_i^k, y_i, \theta_n^\star) \right] + \sum_{i=1}^n \nabla_{\theta\theta} \mathbb{E}_{\mathbb{Q}^\star} \left[\ell(X \odot \xi, Y, \theta_n^\star) | X = x_i, Y = y_i \right] \left(\widehat{\theta}_n^\star(2^K) - \theta_n^\star \right) \\ &\quad + R_K + R_{K,\theta}, \end{aligned} \tag{33}$$

where

$$R_K \equiv \left(\sum_{i=1}^n \left(\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_{\theta\theta} \ell(x_i \odot \xi_i^k, y_i, \theta_n^\star) - \nabla_{\theta\theta} \mathbb{E}_{\mathbb{Q}^\star} \left[\ell(X \odot \xi, Y, \theta_n^\star) | X = x_i, Y = y_i \right] \right) \right) \left(\widehat{\theta}_n^\star(2^K) - \theta_n^\star \right)$$

and

$$\|R_{K,\theta}\|_2 \le \sum_{i=1}^n \sup_{\theta \in \Theta, \xi} \|\nabla_{\theta\theta\theta} \ell(x_i \odot \xi, y_i, \theta)\|_2 \left\| \widehat{\theta}_n^\star(2^K) - \theta_n^\star \right\|_2^2 \le C_3 \left\| \widehat{\theta}_n^\star(2^K) - \theta_n^\star \right\|_2^2$$

by Assumption 4. Thus by Assumption 3, we have

$$\mathbb{E}[R_{K,\theta}^\top R_{K,\theta}] = O(2^{-2K})$$

as $K \to \infty$. Moreover, by the multivariate version of Theorem 2 in Bahr (1965) which follows from the Cramér-Wold theorem, we have that

$$\mathbb{E} \left[\left\| \sum_{i=1}^n \left(\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_{\theta\theta} \ell(x_i \odot \xi_i^k, y_i, \theta_n^\star) - \nabla_{\theta\theta} \mathbb{E}_{\mathbb{Q}^\star} \left[\ell(X \odot \xi, Y, \theta_n^\star) | X = x_i, Y = y_i \right] \right) \right\|_2^4 \right]$$

is $O(2^{-2K})$.

We can express $R_K^\top R_K$ as $\|R_K\|^2$. The Cauchy-Schwarz inequality implies

$$\begin{aligned}&\mathbb{E}[R_K^\top R_K]\\ \leq&\mathbb{E}\left[\left\|\sum_{i=1}^n\left(\frac{1}{2^K}\sum_{k=1}^{2^K}\nabla_{\theta\theta}\ell(x_i\odot\xi_i^k,y_i,\theta_n^\star)-\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}\left[\ell(X\odot\xi,Y,\theta_n^\star)|X=x_i,Y=y_i\right]\right)\right\|_2^2\cdot\right.\\ &\left.\left\|\widehat{\theta}_n^\star(2^K)-\theta_n^\star\right\|_2^2\right].\end{aligned}$$

By Hölder's inequality we have

$$\begin{aligned}&\mathbb{E}[R_K^\top R_K]\\ \leq&\mathbb{E}\left[\left\|\sum_{i=1}^n\left(\frac{1}{2^K}\sum_{k=1}^{2^K}\nabla_{\theta\theta}\ell(x_i\odot\xi_i^k,y_i,\theta_n^\star)-\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}\left[\ell(X\odot\xi,Y,\theta_n^\star)|X=x_i,Y=y_i\right]\right)\right\|_2^4\right]^{\frac{1}{2}}\times\\ &\qquad\qquad\mathbb{E}\left[\left\|\widehat{\theta}_n^\star(2^K)-\theta_n^\star\right\|_2^4\right]^{\frac{1}{2}}\\ \leq&O(2^{-2K}).\end{aligned}$$

Finally, consider the solutions $\widehat{\theta}_n^\star(2^{K_l^\star+1}),\widehat{\theta}_n^O(2^{K_l^\star}),\widehat{\theta}_n^E(2^{K_l^\star})$ conditional on $K_l^\star=K$. Denote the remainder terms in (33) corresponding to the level 2^{K+1} solution $\widehat{\theta}_n^\star(2^{K+1})$ as $R_{K+1}^\star,R_{K+1,\theta}^\star$. Similarly, denote the remainder terms in (33) corresponding to the level 2^K solution $\widehat{\theta}_n^O(2^K)$ (and, respectively, $\widehat{\theta}_n^E(2^K)$) as $R_K^O,R_{K,\theta}^O$ ($R_K^E,R_{K,\theta}^E$). By the construction of $\widehat{\theta}_n^O(2^K),\widehat{\theta}_n^E(2^K)$ using odd and even indices, we have, from (33)

$$\begin{aligned}&-\sum_{i=1}^n\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}[\ell(X\odot\xi,Y,\theta_n^\star)|X=x_i,Y=y_i]\left(\widehat{\theta}_n^\star(2^{K+1})-\frac{1}{2}(\widehat{\theta}_n^O(2^K)+\widehat{\theta}_n^E(2^K))\right)\\ =&R_{K+1}^\star-\frac{1}{2}(R_K^O+R_K^E)+R_{K+1,\theta}^\star-\frac{1}{2}(R_{K,\theta}^O+R_{K,\theta}^E).\end{aligned}$$

By Assumption 4,

$$\sum_{i=1}^n\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}[\ell(X\odot\xi,Y,\theta_n^\star)|X=x_i,Y=y_i]=n\cdot\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}[\ell(X\odot\xi,Y,\theta_n^\star)]$$

is invertible. Thus, we have shown that

$$\begin{aligned}\bar{\Delta}_K &\equiv \widehat{\theta}_n^\star(2^{K+1}) - \frac{1}{2}(\widehat{\theta}_n^O(2^K) + \widehat{\theta}_n^E(2^K)) \\ &= (n \cdot \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta_n^\star)])^{-1} \left(R_{K+1}^\star - \frac{1}{2}(R_K^O + R_K^E) + R_{K+1,\theta}^\star - \frac{1}{2}(R_{K,\theta}^O + R_{K,\theta}^E) \right).\end{aligned}$$

Since each of the terms on the right-hand side have been shown to be $O(2^{-2K})$, we conclude that $\mathbb{E}[\bar{\Delta}_K^\top \bar{\Delta}_K] = O(2^{-2K})$. ■

A.5 Dropout Training in Linear Regression

Corollary 7 (Linear regression with $\phi = 1$) *For linear regression with $\ell(x, y, \beta) = (\beta^\top x - y)^2$, we have*

$$\min_{\beta \in \mathbb{R}^d} \max_{\mathbb{Q} \in \mathcal{U}(\widehat{\mathbb{P}}_n, \delta)} \mathbb{E}_{\mathbb{Q}}\Big[\big(\beta^\top (X \odot \xi) - Y\big)^2\Big] = \min_{\beta \in \mathbb{R}^d} \mathbb{E}_{\mathbb{Q}^\star}\Big[\big(\beta^\top (X \odot \xi) - Y\big)^2\Big],$$

where $\mathbb{Q}^\star = \widehat{\mathbb{P}}_n \otimes \mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$ and $\mathbb{Q}_j^\star = (1-\delta)^{-1} \times$ Bernoulli$(1-\delta)$ for each $j = 1, \ldots, d$. Moreover,

$$\min_{\beta \in \mathbb{R}^d} \mathbb{E}_{\mathbb{Q}^\star}\Big[\big(\beta^\top (X \odot \xi) - Y\big)^2\Big] = \min_{\beta \in \mathbb{R}^d} \frac{1}{n}\left[(\mathbf{Y} - \mathbf{X}\beta)^\top (\mathbf{Y} - \mathbf{X}\beta) + \frac{\delta}{1-\delta}\beta^\top \mathbf{\Lambda}\beta\right], \tag{34}$$

which implies that the dropout training estimator equals

$$\widehat{\beta}(\delta) = \left(\mathbf{X}^\top \mathbf{X} + \frac{\delta}{1-\delta} diag(\mathbf{X}^\top \mathbf{X})\right)^{-1} \mathbf{X}^\top \mathbf{Y}.$$

Finally, if $\mathbb{E}_{P^\star}[XX^\top]$ is a diagonal matrix with strictly positive entries then

$$\widehat{\beta}(\delta) \xrightarrow{p} (1-\delta) E_{P^*}[XX^\top]^{-1}\mathbb{E}[XY].$$

Proof The first part of the corollary follows directly from (13) and (14) in our main theorem. The second part of the corollary follows from Proposition 1. According to this proposition, the limit of $\widehat{\beta}(\delta)$ is

$$\beta^*(\delta) = \Big(\mathbb{E}_{P^*}[XX^\top] + (\delta/1-\delta)\text{diag}(\mathbb{E}_{P^*}[XX^\top])\Big)^{-1} \mathbb{E}_{P^*}[YX].$$

Thus, if $\mathbb{E}_{P^*}[XX^\top]$ is a diagonal matrix, we obtained the desired limit.

■

A.6 Additional Numerical Results

Here we try to provide some justifications for our choice of parameter.

Learning Rate: We first fix an all zeros initialization scheme, and vary the learning rate. We summarize the average parameter divergence and 1-standard deviation error for 20 repetitions of the SGD algorithm in Table 2. We can observe the learning rate 0.0001 shows a clear advantage.

Initialization: Next we fix the learning rate to be 0.0001, and consider different initialization schemes. We note that the mean value (resp., absolute value) of elements in $\beta^\star$ is 0.3947 (resp., 0.6977). Table 3 shows the average parameter divergence and the 1-standard deviation from 20 repetitions of the SGD algorithm. We see that the initialization at origin is a fair choice.

Learning rate	0.001	**0.0001**	0.00001
$\|\widehat{\beta}_{SGD} - \beta^\star\|_\infty$	0.0827 ± 0.0133	$\mathbf{0.0301 \pm 0.0025}$	0.6702 ± 0.1082

Table 2: Comparison for different learning rates, with fixed zero initializations.

Initializations	**all zeros**	all 0.2's	all 1's
$\|\widehat{\beta}_{SGD} - \beta^\star\|_\infty$	$\mathbf{0.0301 \pm 0.0025}$	0.0317 ± 0.0047	0.0614 ± 0.0196
Initializations	i.i.d $\mathcal{N}(0,1)$	i.i.d $\mathcal{N}(0,10)$	i.i.d $\mathcal{N}(0,10^2)$
$\|\widehat{\beta}_{SGD} - \beta^\star\|_\infty$	0.0376 ± 0.0067	0.1006 ± 0.0469	0.3208 ± 0.1432

Table 3: Comparison for different initialization schemes with fixed learning rate 0.0001.

Wall-Clock Time: We then document the numerical results for 120s/180s wall-clock time, see Figures 4 - 6 for the case of 120s and Figures 7 - 9 for the case of 180s. We see that the proposed unbiased approach outperforms the standard SGD when the number of parallel iterations reaches above some threshold.

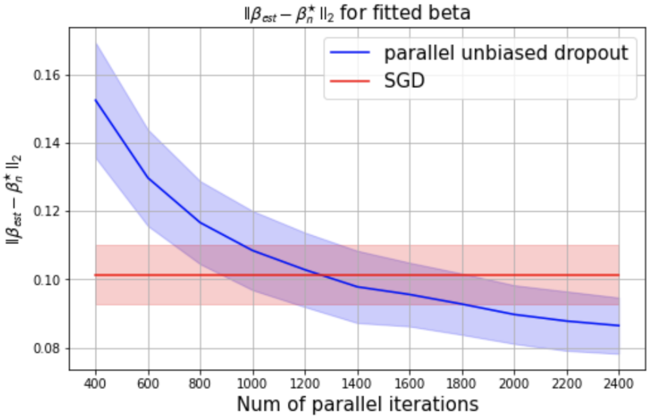

Figure 4: l_2 difference for 120s wall-clock time

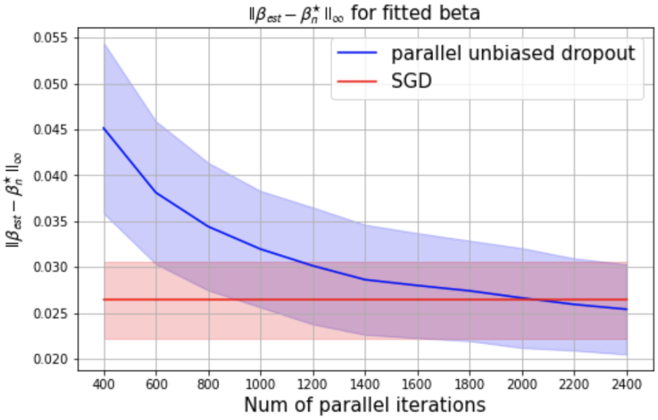

Figure 5: l_∞ difference for 120s wall-clock time

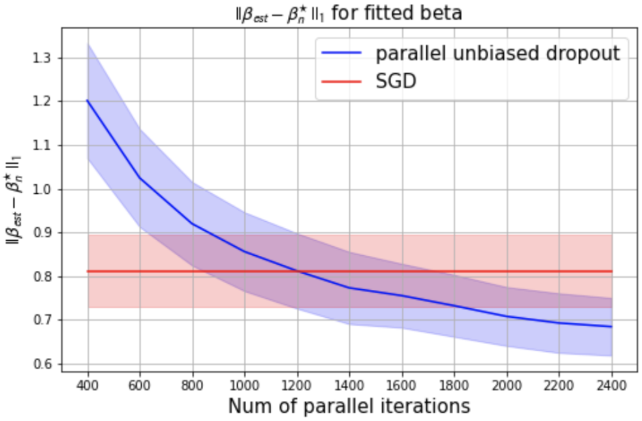

Figure 6: l_1 difference for 120s wall-clock time

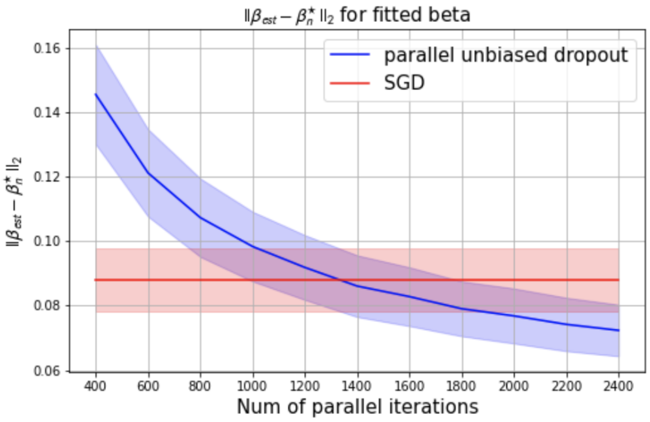

Figure 7: l_2 difference for 180s wall-clock time

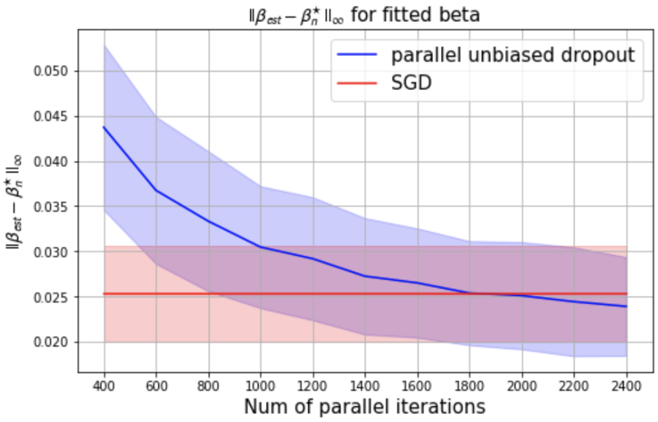

Figure 8: l_∞ difference for 180s wall-clock time

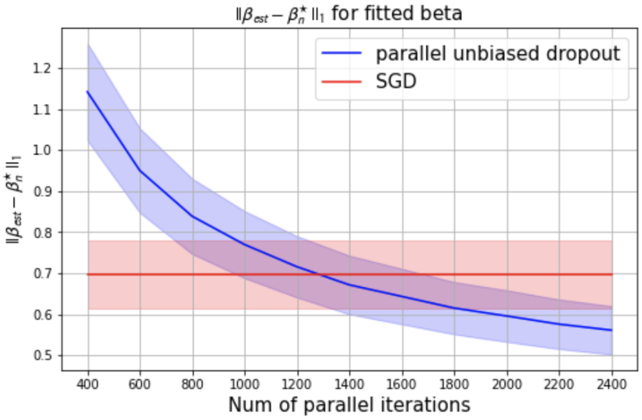

Figure 9: l_1 difference for 180s wall-clock time

References

Sule Alan, Orazio Attanasio, and Martin Browning. Estimating Euler equations with noisy data: two exact GMM estimators. *Journal of Applied Econometrics*, 24(2):309–324, 2009.

Bengt Von Bahr. On the convergence of moments in the central limit theorem. *Annals of Mathematical Statistics*, 36(3):808–818, 06 1965.

Chris M Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

Jose Blanchet, Peter Glynn, and Yanan Pei. Unbiased multilevel Monte Carlo: Stochastic optimization, steady-state simulation, quantiles, and other applications. *arXiv preprint arXiv:1904.09929*, 2019a.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56:830–857, 2019b.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

Timothy Christensen and Benjamin Connault. Counterfactual sensitivity and robustness. *arXiv preprint arXiv:1904.00989*, 2019.

A. DasGupta. *Asymptotic Theory of Statistics and Probability.* Springer Verlag, 2008.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

David Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society, Series B*, 56, 1994.

Rick Durrett. *Probability: Theory and Examples.* Cambridge University Press, 2019.

Ludwig Fahrmeir and Heinz Kaufmann. Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, pages 342–368, 1985.

Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Forthcoming at Econometrica*, 2020.

T.S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*, volume 7. Academic Press New York, 1967.

Michael B Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3): 607–617, 2008.

Michael B Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259, 2015.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

Lars Peter Hansen and Thomas J. Sargent. *Robustness.* Princeton University Press, 2008.

David P Helmbold and Philip M Long. On the inductive bias of dropout. *The Journal of Machine Learning Research*, 16(1):3403–3454, 2015.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Jiunn T Hwang. Multiplicative errors-in-variables models with applications to recent data released by the us department of energy. *Journal of the American Statistical Association*, 81(395):680–688, 1986.

J Kim and W Winkler. Multiplicative noise for masking continuous data. *Statistics*, 1:9, 2003.

Robert H Lyles and Lawrence L Kupper. A detailed evaluation of adjustment methods for multiplicative measurement error in linear regression with applications in occupational epidemiology. *Biometrics*, pages 1008–1025, 1997.

Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pages 410–418, 2013.

Peter McCullagh and J.A. Nelder. *Generalized Linear Models.* Chapman & Hall, 1989.

Oskar Morgenstern and John von Neumann. *Theory of Games and Economic Behavior.* Princeton University Press, 1953.

Tapan K Nayak, Bimal Sinha, and Laura Zayatz. Statistical properties of multiplicative noise masking for confidentiality protection. *Journal of Official Statistics*, 27(3):527, 2011.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of Econometrics*, 4:2111–2245, 1994.

Viet Anh Nguyen, Xuhui Zhang, Jose Blanchet, and Angelos Georghiou. Distributionally robust parametric maximum likelihood estimation. In *Advances in Neural Information Processing Systems 33*, 2020.

Donald A Pierce, Daniel O Stram, Michael Vaeth, and Daniel W Schafer. The errors-in-variables problem: considerations provided by radiation dose-response analyses of the a-bomb survivor data. *Journal of the American Statistical Association*, 87(418):351–359, 1992.

Adrian E Raftery, David Madigan, and Jennifer A Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437): 179–191, 1997.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

H. Scarf. A min-max solution of an inventory problem. *Studies in the Mathematical Theory of Inventory and Production*, 10:201–209, 1958.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020.

Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, 2014.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359. 2013.

Martin J Wainwright and Michael Irwin Jordan. *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers Inc, 2008.

Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning*, pages 118–126, 2013.

Colin Wei, Sham Kakade, and Tengyu Ma. The implicit and explicit regularization effects of dropout. In *Proceedings of the International Conference of Machine Learning*, 2020.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.