AN ADVERSARIAL APPROACH TO STRUCTURAL ESTIMATION

Tetsuya Kaji[1], Elena Manresa[2], and Guillaume Pouliot[1]

[1]University of Chicago
[2]New York University

August 3, 2023

Abstract

We propose a new simulation-based estimation method, adversarial estimation, for structural models. The estimator is formulated as the solution to a minimax problem between a generator (which generates simulated observations using the structural model) and a discriminator (which classifies whether an observation is simulated). The discriminator maximizes the accuracy of its classification while the generator minimizes it. We show that, with a sufficiently rich discriminator, the adversarial estimator attains parametric efficiency under correct specification and the parametric rate under misspecification. We advocate the use of a neural network as a discriminator that can exploit adaptivity properties and attain fast rates of convergence.

JEL Codes: C13, C45.

Keywords: structural estimation, generative adversarial networks, neural networks, simulated method of moments, indirect inference, efficient estimation.

1 INTRODUCTION

Structural estimation is a useful approach to learn about the effects of policies that are yet to be implemented. Structural models are naturally articulated as parametric models and, as such, may be estimated using maximum likelihood (MLE). However,

We thank Isaiah Andrews, Manuel Arellano, Stephane Bonhomme, Aureo De Paula, Costas Meghir, Chris Hansen, Koen Jochmans, Whitney Newey, Luigi Pistaferri, Bernard Salanié, Dennis Kristensen, Anna Mikusheva, Zhenling Jiang, Xintong Han, and Daniel Waldinger, as well as numerous participants in conferences for helpful discussion. Elsie Hoffet, Yijun Liu, Ignacio Cigliutti, and Marcela Barrios provided superb research assistance. We gratefully acknowledge the support of the NSF by means of the Grant SES-1824304, the Alfred P. Sloan Foundation Fellowship, and the Richard N. Rosett Faculty Fellowship and the Liew Family Faculty Fellowship at the University of Chicago Booth School of Business.

likelihood functions are sometimes too complex to evaluate or may not exist in closed form. This has motivated large literature on simulation-based estimation methods.

A prominent example of such methods is the simulated method of moments (SMM) (McFadden, 1989; Pakes and Pollard, 1989). If we want identification and estimation of the parameters to rely on specific features, SMM is a natural tool as long as such features can be expressed as moments. At the same time, the naive strategy of stacking many moments is known to yield poor finite sample properties (Altonji and Segal, 1996). This tradeoff is especially pronounced in models with rich heterogeneity, where the number of moments may grow rapidly with the number of covariates. While this problem may be resolved if we can reduce the moments to a handful of informative ones, such a choice is often not obvious.

This paper proposes a new simulation-based estimation method, which we call *adversarial estimation*. It is inspired by the *generative adversarial networks (GAN)*, a machine learning algorithm developed by Goodfellow et al. (2014) to generate realistic images. We adopt their adversarial framework to estimate the structural parameters that generate realistic economic data. The proposed estimator achieves efficiency under correct specification and the parametric rate under misspecification. Thus, our method is useful in applications where the likelihood is not computable but simulation is feasible and it can be a more efficient alternative to SMM.

The generative adversarial estimation framework is a minimax game between two components—the *discriminator* and the *generator*—over classification accuracy:

$$\min_{\{generator\}} \max_{\{discriminator\}} \textit{classification accuracy}.$$

The generator is an algorithm that produces the simulated data; its objective is to find a data-generating process that confuses the discriminator. The discriminator is a classification algorithm that distinguishes the observed data from the simulated data; it takes an observation as input and classifies it as coming from either observed data or simulated data; its objective is to maximize the accuracy of its classification.

In the original GAN, both the discriminator and the generator are given as neural networks (hence the name). In this paper, we take the generator to be the structural model we intend to estimate and the discriminator to be an arbitrary classification algorithm (while our primary choice is a neural network). To quantify classification

accuracy, we employ the cross-entropy loss, following Goodfellow et al. (2014).[1]

Interestingly, our framework establishes a bridge between SMM and MLE. When we use a logistic discriminator with inputs equal to moments, the resulting estimator is asymptotically equivalent to optimally-weighted SMM (Appendix S.1). When we use the oracle discriminator, the resulting estimator is equivalent to MLE when the simulation sample size increases faster than the actual sample size, since our classification accuracy is a symmetrized Kullback–Leibler divergence. Of particular interest is the middle case, in which the oracle discriminator is not available but a sufficiently rich discriminator capable of approximating it is used. Under some conditions, the resulting estimator enjoys the desirable properties of both SMM and MLE: the user has the flexibility to choose moments if desired, a closed-form likelihood is not required, and the asymptotic efficiency is attained.

We illustrate the theoretical properties of our estimator in simulations using simple models. We show that the curvature of the classification accuracy is comparable to that of the log likelihood function for a suitable choice of discriminator. In addition, we show that the estimator can achieve the parametric rate under misspecification and has smaller bias compared to SMM. We also showcase the implementation of the method using the Roy Model with two occupations over two time periods.

Our method contributes to the vast literature of simulation-based estimation. The first application of GAN to economics is due to Athey et al. (2020), who apply GAN to produce a realistic sampler of economic data for Monte Carlo studies. In contrast, in our paper the generative model is identified and we aim to estimate its parameters. In computer science, the literature on GAN is rapidly growing; for a recent review, see, e.g., Cheng et al. (2020) or Asimopoulos et al. (2022).

The rest is organized as follows. Section 2 defines the setup. Section 3 illustrates the estimator with simple examples. Section 4 develops the asymptotic properties.

2 ADVERSARIAL ESTIMATION FRAMEWORK

The adversarial estimation has two main components: simulation and discrimination. The simulation component is the same as other simulation-based estimation meth-

[1]There are also other losses considered in the literature. In machine learning, they concern high-dimensional data such as images, sounds, and texts, and the Wasserstein distance has gained huge popularity for its ability to measure the distance of disjoint probability distributions. It is also used in economic applications (Athey et al., 2020).

ods, such as SMM or indirect inference, but the discriminator component is new. The essence of the adversarial framework is to find a parameter value for which the corresponding simulated data is indistinguishable from the real data according to the discriminator. We now describe each component in turn.

Suppose we have data $\{X_i\}_{i=1}^n$ drawn i.i.d. from an unknown distribution P_0. Suppose we have a fully parametric model $\{P_\theta : \theta \in \Theta\}$ for which the likelihood is not tractable but simulation is feasible.[2] Our target is the parameter θ that best describes the distribution of the data P_0 through the model P_θ.

We formalize the simulation process as follows: for a given θ and a given sample size m, we obtain a sample of simulated observations, $\{X_{i,\theta}\}_{i=1}^m$, according to model P_θ by taking draws $\{Z_i\}_{i=1}^m$ from a known distribution P_Z and applying a transformation T_θ to them, $X_{i,\theta} = T_\theta(Z_i)$. Typically, Z_i is a vector of independent uniform random variables and $X_{i,\theta}$ is generated by inverse transform sampling, that is, $X_{i,\theta} = F_\theta^{-1}(Z_i)$. For models with intractable likelihood, T_θ may further involve optimization or integration. For illustration, take the example of a normal location model with known variance 1 and unknown mean θ, i.e., $P_\theta = N(\theta, 1)$. Then, we can generate a simulated observation $X_{i,\theta}$ from P_θ through $X_{i,\theta} = \theta + \Phi^{-1}(Z_i)$, where Φ is the standard normal cdf and $Z_i \sim U[0, 1]$.

We now turn to the discriminator. The discriminator is the novelty in the estimation framework and is the key component in the construction of the objective function for the adversarial estimator. For some θ and x, consider the problem of assessing whether x is from P_θ or P_0. If P_θ is very different from P_0, it should be easy to distinguish realizations of P_θ from those of P_0. If they are close, it should be harder. The idea, therefore, is to pick a classification algorithm that takes a value x and predicts which distribution it came from, and to search for the value of θ for which the algorithm is least able to classify the data correctly.

If we had access to the probability density functions corresponding to P_0 and P_θ, it would be easy to assign the provenance of x according to the likelihood of x for each distribution. This suggests an estimation strategy based on the search of θ for which the probability that any draw $X_{i,\theta}$ is drawn from P_0 versus P_θ is 0.5. Since we do not have access to the probability distributions, this strategy is infeasible. However, we can take advantage of the availability of samples $\{X_i\}_{i=1}^n$ and $\{X_{i,\theta}\}_{i=1}^m$ to estimate the extent to which, for a given θ, these two distributions are different. In particular,

[2] This is the case for many structural models in economics involving dynamic decision making.

we use the predictions of a discrete choice model (called the discriminator), where the dependent variable is 1 if the data is real and 0 if it is simulated, and the explanatory variables are X_i if the data is real, and $X_{i,\theta}$ if it is simulated. When θ is a poor candidate to describe the observed data, the predictions will be either close to 1 or close to 0. However, as θ becomes a better candidate to describe the real data, the distribution of the prediction will concentrate around 1/2.

Formally, classification is defined as a function $D : \mathcal{X} \to [0,1]$ such that $D(x)$ represents the likelihood of x being an actual observation; $D(x) = 1$ means that x is classified as "actual" with certainty; $D(x) = 0$ that x is classified as "simulated" with certainty. Denote by $\mathcal{D}_n$ the class of classification functions we consider. The dependence on n allows us to use a richer classification algorithm as the sample size gets larger. The choice of $\mathcal{D}_n$ is an important one for the researcher as it impacts the properties of the estimator. While any class of binary choice models would work, certain choices will have attractive properties, as we discuss below.

The *adversarial estimator* is defined by the following minimax problem:[3]

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \max_{D\in\mathcal{D}_n} \frac{1}{n}\sum_{i=1}^{n} \log D(X_i) + \frac{1}{m}\sum_{i=1}^{m} \log(1 - D(X_{i,\theta})).$$

Since D is between 0 and 1, both $\log D$ and $\log(1-D)$ are nonpositive. If $\{X_i\}$ and $\{X_{i,\theta}\}$ are very different from each other, the discriminator may be able to find D that assigns 1 on the support of $\{X_i\}$ and 0 on the support of $\{X_{i,\theta}\}$, in which case the inner maximization attains the value of zero. Meanwhile, regardless of the values of $\{X_i\}$ and $\{X_{i,\theta}\}$, the discriminator can always attain the classification accuracy of $2\log(1/2)$ by setting $D \equiv 1/2$.[4] In general, therefore, the inner maximization will give a number between $2\log(1/2)$ and 0, and the closer it is to $2\log(1/2)$, the less able the discriminator is to classify the observations.

When we let n and m grow, we obtain the population counterpart of the problem

$$\min_{\theta\in\Theta} \max_{D\in\mathcal{D}_n} \mathbb{E}_{X_i\sim P_0}[\log D(X_i)] + \mathbb{E}_{X_{i,\theta}\sim P_\theta}[\log(1 - D(X_{i,\theta}))].$$

If there is no restriction on $\mathcal{D}_n$ (so any function $D : \mathcal{X} \to [0,1]$ is allowed), the

[3] Minimization and maximization need not be solved exactly (Assumptions 3 and S.2).

[4] This is of course provided that a constant function 1/2 is in $\mathcal{D}_n$, which is usually the case.

optimum classifier for the population inner maximization is known to be

$$D_\theta(x) := \frac{p_0(x)}{p_0(x) + p_\theta(x)},$$

where p_0 and p_θ are the densities of P_0 and P_θ with respect to some common dominating measure (Goodfellow et al., 2014, Proposition 1).[5] We call this D_θ the *oracle discriminator*. If the model is correctly specified, then θ_0 is the unique solution to the outer minimization (Goodfellow et al., 2014, Theorem 1). In the normal location model, if we assume $P_0 = N(0,1)$, the oracle discriminator is given by $D_\theta(x) = \Lambda(\frac{1}{2}\theta^2 - \theta x) = \Lambda(-\theta(x - \frac{1}{2}\theta))$. Since Λ is a standard logistic cdf, $\Lambda(0) = 1/2$, $\lim_{t\to\infty}\Lambda(t) \to 1$, and $\lim_{t\to-\infty}\Lambda(t) \to 0$. Therefore, if $\theta < 0$, positive deviation of x from $\theta/2$ is classified as more likely an actual observation, and negative deviation as less likely; if $\theta = 0$, whatever value of x has an equal chance of being actual.

Certain choices of $\mathcal{D}_n$ make for interesting special cases. First, if we use the oracle discriminator D_θ, the resulting estimator for θ becomes efficient under correct specification and $m \gg n$. In the normal location model, we see that as $m \to \infty$, the oracle estimator solves

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \log\Lambda\left(\frac{1}{2}\theta^2 - \theta X_i\right) + \mathbb{E}_\theta\left[\log\left(1 - \Lambda\left(\frac{1}{2}\theta^2 - \theta X_{i,\theta}\right)\right)\right].$$

The FOC combined with the first-order Taylor expansion of Λ around 0 yields

$$0 = \frac{1}{n}\sum_{i=1}^{n}(\theta - X_i)\left[1 - \Lambda\left(\frac{\theta^2}{2} - \theta X_i\right)\right] - \mathbb{E}_\theta\left[(\theta - X_{i,\theta})\Lambda\left(\frac{\theta^2}{2} - \theta X_{i,\theta}\right)\right] \approx \frac{1}{2n}\sum_{i=1}^{n}(\theta - X_i).$$

Therefore, $\hat{\theta}$ is approximately the sample average, which is the MLE.

Second, if we use the logistic discriminator and $n = m$, the cross-entropy loss can be interpreted as the (scaled) log likelihood of the logistic regression where the actual observations are labeled 1 and the simulated ones are labeled 0.[6] The resulting estimator for θ is then asymptotically equivalent to the optimally-weighted SMM with moments $\mathbb{E}[X_i]$ under $m \gtrsim n$ (Appendix S.1). In practice, we may use a sieve of discriminators that can represent oracle D_θ asymptotically, e.g., the sieve of neural networks or the sieve of logistic discriminators with an increasing number of polyno-

[5] Note that D_θ does not depend on the relative sizes of n and m, since we take the averages, not the sums, of classification accuracy.

[6] When $n \neq m$, the binary cross-entropy loss weights the two sets of observations differently.

mials of X. In fact, we can regard D_θ as the nuisance parameter estimated in the inner maximization. Section 4 presents conditions under which the estimation of D_θ via nonparametric estimation makes the adversarial estimator efficient.

The asymptotic distribution of the adversarial estimator depends on the choice of $\mathcal{D}_n$. If the discriminator is logistic, the asymptotic variance of the adversarial estimator coincides with SMM (Appendix S.1). If $\mathcal{D}_n$ is a nonparametric discriminator, under some conditions, the asymptotic variance will be a function of the score and Hessian of the likelihood (Theorem 3 and Assumption 6). When the likelihood is intractable, estimating this asymptotic variance formula is not an easy task; we recommend using bootstrap in which we resample both $\{X_i\}_{i=1}^n$ and $\{Z_i\}_{i=1}^m$ with replacement. We note, however, that, while well-corroborated by simulation exercises, the validity of bootstrap is not theoretically established in this paper.

The estimation algorithm when $\mathcal{D}_n$ is a class of neural networks and the transformation $T_\theta(\cdot)$ is differentiable in θ is given in Algorithm 1. The algorithm is based on an iterative strategy in which (1) given a particular value θ, we train the discriminator to completion using $\{X_i\}_{i=1}^n$ and $\{X_{i,\theta}\}_{i=1}^m$, and (2) we update the value of θ according to the direction of the gradient of the objective function taking as fixed the predictions of the discriminator obtained in the previous step. Training the discriminator till completion, while computationally costly, provides a much more reliable strategy to compute the estimator, as opposed to only training the discriminator for a few steps for each update in θ. The algorithm is initialized with a random value of θ and with m independent draws from the distribution P_Z. Essential is the fact that we use the same shocks $\{Z_i\}_{i=1}^m$ to generate $\{X_{i,\theta}\}_{i=1}^m$ across different θ.[7]

To train the discriminator, we use the Adam algorithm (Kingma and Ba, 2015), which combines the estimate of the current gradient with previous estimates of the gradient and information about the second moment.[8] We use a version in which the gradient is stochastic, evaluated only at a small subsample of the data (minibatch, mb), and we update the predictions of the discriminator by looping through all minibatches in the dataset, and through many iterations across the entire dataset.

[7]Importance of using the same shocks is widely known in the literature. For example, Nevo (2000, Appendix) states that "it is important to draw these only once at the beginning of the computation. If the draws are changed during the computation the non-linear search is unlikely to converge."

[8]There are four tuning parameters in Adam: the learning rate l_r^D, the exponential decay of the first moment β_1, and the exponential decay of the second moment β_2, and ε, a small number to avoid dividing by zero.

Algorithm 1 Algorithm for adversarial estimation with a neural network discriminator. The discriminator is trained to completion for every update of the generator. We use Adam stochastic gradient descent with minibatch for the discriminator, and gradient descent with adaptive learning for the generator.

Input: - $\{X_i\}_{i=1}^n$, actual data - P_Z, distribution of shocks
- m, simulation sample size - T_θ, structural transformation
- $\mathcal{D}_n$, discriminator, e.g., one hidden layer ten nodes NN

Tuning Adam: - mb, minibatch size, assumed wlog to be a factor of $n+m$
- l_r^D, β_1, β_2, ϵ, parameters for Adam hypergradient descent
- n_{epochs}, number of iterations across the entire dataset

Tuning GD: - $l_{r,(0)}^\theta$, initial learning rate vector for gradient descent
- l^f, l^b, parameters to increase or decrease the learning rate

Output: - Estimate $\hat{\theta}$

▷ Initialization

Sample $Z_i \sim P_Z$ for $i = 1, \ldots, m$ ▷ Shocks are drawn only once here

$k \leftarrow 0$ ▷ Initialize the gradient descent counter

$\theta_k \leftarrow$ initial value

▷ Main loop for gradient descent

while θ_k has not converged **do**

  $X_{i,\theta_k} \leftarrow T_{\theta_k}(Z_i)$ for $i = 1, \ldots, m$ ▷ Generate the simulated data for θ_k

  ▷ Train the discriminator till completion.

  $D_k \leftarrow$ initial network

  $d \leftarrow 1$ ▷ Initialize the epoch counter

  while out-of-sample classification improves **and** $d \leq n_{epochs}$ **do**

    for all b in $1 : \frac{n+m}{mb}$ **do** ▷ b indexes the minibatches

      $(\{X_i^b\}_{i=1}^{mb}, \{X_{i,\theta_k}^b\}_{i=1}^{mb}) \sim (\{X_i\}, \{X_{i,\theta_k}\})$ ▷ Sample minibatch from data

      ▷ Compute the gradient with respect to the parameters in D

      $\delta_k^D(b) \leftarrow \nabla_D \frac{1}{mb} \sum_{i=1}^{mb} \log D_k(X_i^b) + \frac{1}{mb} \sum_{i=1}^{mb} \log(1 - D_k(X_{i,\theta_k}^b))$

      $D_k \leftarrow \text{Adam}(\delta_k^D(b), D_k, l_r^D, \beta_1, \beta_2)$ ▷ Update the discriminator

    end for

    $d \leftarrow d + 1$

  end while

  ▷ Compute the gradient with respect to θ, keeping the discriminator fixed

  $\delta_k^\theta \leftarrow \nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \log(1 - D_k(X_{i,\theta_k}))$

  $l_{r,k+1}^\theta \leftarrow \text{update}(l_{r,k}^\theta, \delta_k^\theta, \delta_{k-1}^\theta, l^f, l^b)$ ▷ Update the learning rate

  $\theta_{k+1} \leftarrow \theta_k - l_{r,k+1}^\theta \delta_k^\theta$ ▷ Update θ by gradient descent

  $k \leftarrow k + 1$ ▷ Update the iteration counter

end while

return $\theta_{(k)}$

The number of iterations across the dataset is called the number of *epochs*, and we typically set it on the order of thousands.[9]

To train the generator, we use standard gradient descent with a simple adaptive learning rate rule that combines information on the sign of each of components of the current and the previous gradient. For each component, when the sign of the previous gradient and the current gradient coincide, the learning rate of such component increases by a constant factor of $l_f > 1$. On the other hand, if the signs are opposite, the learning rate decreases by a factor of $0 < l_b < 1$. This strategy, in our experience, significantly speeds up convergence and helps escape areas of the parameter space where the gradient is flat due to the distributions of X_i and $X_{i,\theta}$ being too far apart.

Finally, the algorithm converges when two critera are met: changes in θ are small, and the loss function is bounded away from zero.

3 ILLUSTRATION WITH SIMPLE EXAMPLES

We illustrate properties of our estimator using simple examples. The first example we consider is a logistic location model in which the mean is unknown and the variance is known. We illustrate three points using this example: (1) the adversarial estimator achieves parametric efficiency under correct specification; (2) the adversarial estimator is asymptotically normal under model misspecification; (3) the adversarial estimator is less sensitive to the curse of dimensionality than SMM. Next, we consider a Roy model with two occupations over two time periods. This example illustrates the whole procedure of estimation and inference in a case when the likelihood is intractable.

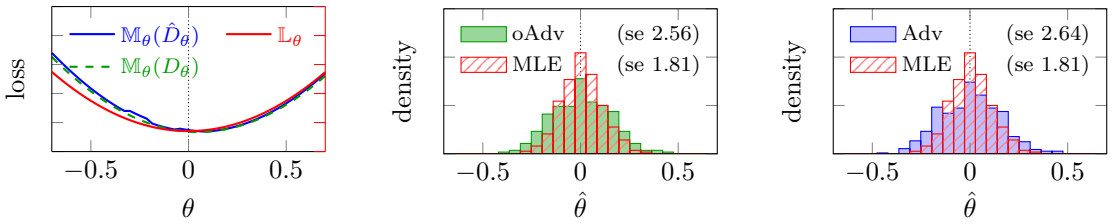
We write $\mathbb{L}_\theta \coloneqq -\frac{1}{2n}\sum_{i=1}^n \log p_\theta(X_i)$ for minus half the log likelihood and $\mathbb{M}_\theta(D) \coloneqq \frac{1}{n}\sum_{i=1}^n \log D(X_i) + \frac{1}{m}\sum_{i=1}^m \log(1 - D(X_{i,\theta}))$ for the sample objective function.

3.1 Logistic Location Model

3.1.1 Efficiency

Suppose we have n i.i.d. observations $X_1, \ldots, X_n$ from the standard logistic distribution with pdf $p_0(x) = \Lambda(x)(1-\Lambda(x))$. Our structural model is the logistic distribution

[9] Call back options, in which the optimization procedure stops after the loss function does not improve significantly for a few iterations, can be helpful to cut computation time. In addition, dropout is also useful, in our experience, for regularization.



(a) Curvature of cross-entropy loss and log likelihood. (b) Oracle adversarial estimator and MLE. (c) Adversarial estimator and MLE.

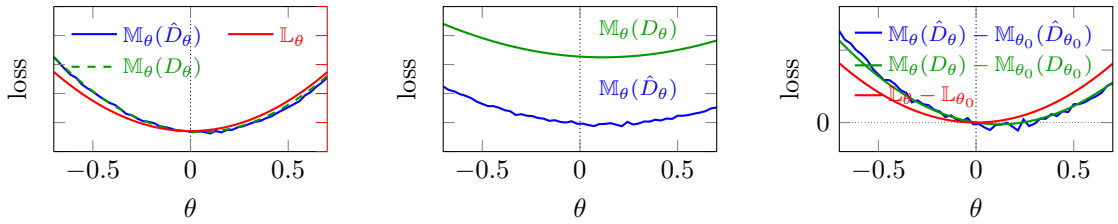
Figure 1: The logistic location model. The curvature of oracle and estimated cross-entropy losses matches the log likelihood (a). This makes the adversarial estimator comparable with MLE (c) and as good as the oracle estimator (b). The standard errors (se) are multiplied by $\sqrt{n}$. The vertical dots indicate the true parameter θ_0.

with unit scaling, i.e., $p_\theta(x) = \Lambda(x-\theta)(1-\Lambda(x-\theta))$. The oracle discriminator is given by $D_\theta(x) = \Lambda(-\theta - 2\log(1+e^{-x}) + 2\log(1+e^{-(x-\theta)}))$. The synthetic data is generated as $X_{i,\theta} = T_\theta(Z_i) \coloneqq \theta + Z_i$ where Z_i follows the standard logistic distribution. We set $n = m = 300$ and run 500 replications.

To yield a discriminator capable of representing the oracle, we consider $D(x;\lambda) = \Lambda(\lambda_0 - 2\log(1+e^{-x}) + 2\log(1+e^{-x+\lambda_1}))$ parameterized by $\lambda \in \mathbb{R}^2$. This class of discriminator is "correctly specified" in that the oracle discriminator is a special case, $\lambda_\theta \coloneqq (-\theta, \theta)^\top$; thus, it allows us to ignore the approximation error for D_θ and focus on aspects conducive to efficiency. Nevertheless, we also present results with a nonparametric estimator, a shallow neural network, at the end of this section.

An intuition behind efficiency is that the curvature of $\mathbb{M}_\theta(\hat{D}_\theta)$ at θ_0 is proportional to the Fisher information. Figure 1a illustrates this point. First, the curvature of $\mathbb{L}_\theta$ is a quarter of the Fisher information, and so is the curvature of the oracle loss $\mathbb{M}_\theta(D_\theta)$ (Lemma S.3). Second, the estimated loss $\mathbb{M}_\theta(\hat{D}_\theta)$ traces $\mathbb{M}_\theta(D_\theta)$ very well. As a result, the curvature of $\mathbb{M}_\theta(\hat{D}_\theta)$ becomes a quarter of the Fisher. This is somewhat surprising since $\hat{D}_\theta$ is estimated separately for each θ (Algorithm 1, line 8); the plot of $\mathbb{M}_\theta(\hat{D}_\theta)$ could have been zigzag if maximization was noisy each time.

An important practice that effects a "smooth" $\mathbb{M}_\theta(\hat{D}_\theta)$ is to use a deterministic algorithm for the inner maximization. Here, we use Matlab's `fminsearch` for maximization, which employs a deterministic algorithm. However, if some stochastic optimization is to be used, we advise that the random seed be reset to the same value each time maximization is carried out. For a logistic discriminator with differentiable T_θ, Section 4.2.1 shows that the estimated loss $\mathbb{M}_\theta(\hat{D}_\theta)$ will be smooth in θ if $\{Z_i\}$ is



(a) $m = n$. The curve of $\mathbb{M}_\theta(\hat{D}_\theta)$ matches the oracle.

(b) $m = 2n$. The level is off, but the curvature is right.

(c) $m = 2n$. Demeaned (b) to highlight the curvature.

Figure 2: Use of a neural network discriminator on the logistic location model.

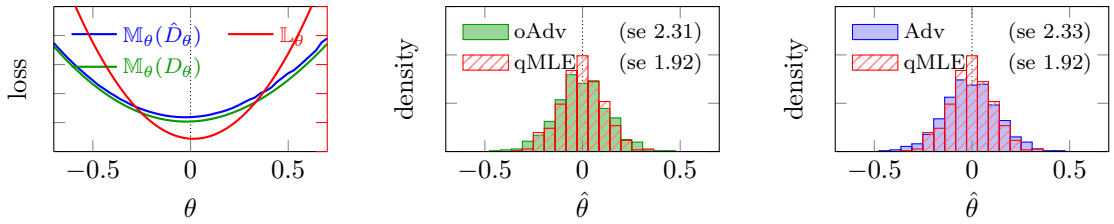
fixed and the exact maximum is attained at the inner step for each θ.

With the curvature of $\mathbb{M}_\theta(\hat{D}_\theta)$ matching $\mathbb{M}_\theta(D_\theta)$, the asymptotic variance of the adversarial estimator is $1 + n/m$ times the inverse Fisher (Corollary 4, Section 4.3). In this example, the theoretical asymptotic standard deviation of MLE is 1.73 while that of the adversarial estimator is 2.45, which are closely reproduced in Figures 1b and 1c.[10]

Similar results hold when m is increased (figures omitted); the curvatures of $\mathbb{M}_\theta(D_\theta)$ and $\mathbb{M}_\theta(\hat{D}_\theta)$ match closely with that of $\mathbb{L}_\theta$, and the adversarial estimator gets closer to the MLE. For example, when $m = 3{,}000$ (so $m = 10n$), the simulation standard error of the adversarial estimator decreases to 1.82 (which is almost identical to the theoretical asymptotic standard error of 1.82).

To see how a nonparametric discriminator fares, we also try a shallow neural network discriminator. The input is a one-dimensional observation X; there are three nodes in one hidden layer with a hyperbolic tangent activation function; the output is a sigmoid function. The neural network discriminator is trained for each θ using Matlab's `train` function, which is deterministic. Figure 2a shows that the estimated loss $\mathbb{M}_\theta(\hat{D}_\theta)$ still gives a good approximation to $\mathbb{M}_\theta(D_\theta)$. It is notable that as we increase m, the *level* of $\mathbb{M}_\theta(\hat{D}_\theta)$ departs from that of $\mathbb{M}_\theta(D_\theta)$, but the *curvature* is still correctly estimated (Figure 2b). If we adjust the level, it becomes clear that the curvature matches that of the log likelihood (Figure 2c). According to our theory, the quality of the adversarial estimator hinges on the curvature of $\mathbb{M}_\theta(\hat{D}_\theta)$—but not on the level of $\mathbb{M}_\theta(\hat{D}_\theta)$—being close to that of $\mathbb{M}_\theta(D_\theta)$. Thus, the resulting estimator is very close to the oracle (figures omitted).

[10] All theoretical asymptotic standard deviations of the adversarial estimators in Section 3 are calculated with a general version of Theorem 3 that allows $\lim_{n\to\infty} n/m > 0$ (Kaji et al., 2022, Theorem 3).



(a) Loss and quasi-log likelihood.

(b) Oracle adversarial estimator and quasi-MLE.

(c) Adversarial estimator and quasi-MLE.

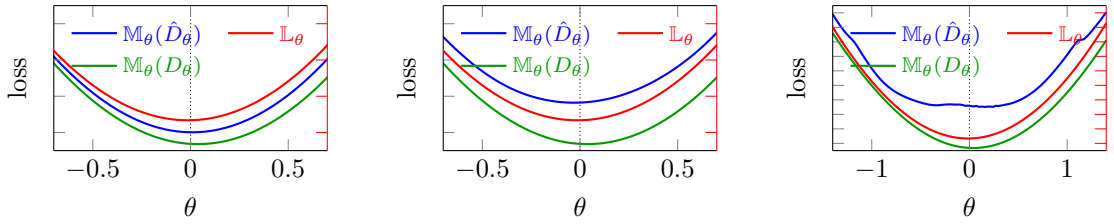
Figure 3: The normally-misspecified logistic location model. The adversarial estimator is comparable with quasi-MLE.

We also examine whether bootstrap works for the adversarial estimator. The bootstrap consists of 500 replications where both $\{X_i\}_{i=1}^n$ and $\{Z_i\}_{i=1}^m$ are resampled with replacement, but where we hold fixed the specification of the discriminator. The bootstrap standard error is 2.28 for the logistic discriminator and 2.55 for the neural network discriminator, both of which are close to the theoretical limit 2.45.

3.1.2 Normality under Misspecification

We now explore how the adversarial estimator behaves under misspecification. Suppose we misspecify the model to be a normal location family with unit variance, $p_\theta(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{(x-\theta)^2}{2})$, while the true distribution is still the standard logistic distribution with variance $\pi^2/3 \approx 3.3$. The oracle discriminator is $D_\theta(x) = \Lambda(\log\sqrt{2\pi} - x + \frac{1}{2}(x-\theta)^2 - 2\log(1+e^{-x}))$. Here, we use the correctly specified discriminator $D(x;\lambda) = \Lambda(\lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 \log(1+e^{-x}))$ parameterized by $\lambda \in \mathbb{R}^4$.

Figure 3a shows that the curvature of $\mathbb{L}_\theta$ is much steeper than $\mathbb{M}_\theta(D_\theta)$ due to misspecification (particularly to misspecification of variance). However, the estimated loss $\mathbb{M}_\theta(\hat{D}_\theta)$ still estimates the curvature of the oracle loss correctly. Figure 3b shows that the oracle adversarial estimator is approximately normal and comparable with quasi-MLE. A slight inflation of the variance is due to the fact that the adversarial estimator uses the synthetic data and gets affected by their randomness while quasi-MLE does not. Figure 3c shows that the adversarial estimator is very close to the oracle one. The theoretical asymptotic standard deviation of the adversarial estimator is 2.27 while of quasi-MLE is 1.81. The results for the increased synthetic sample size m and for the neural network discriminator are analogous to Section 3.1.1 and hence omitted for space.



(a) Loss with 3 moments. (b) Loss with 7 moments. (c) Loss with 11 moments.

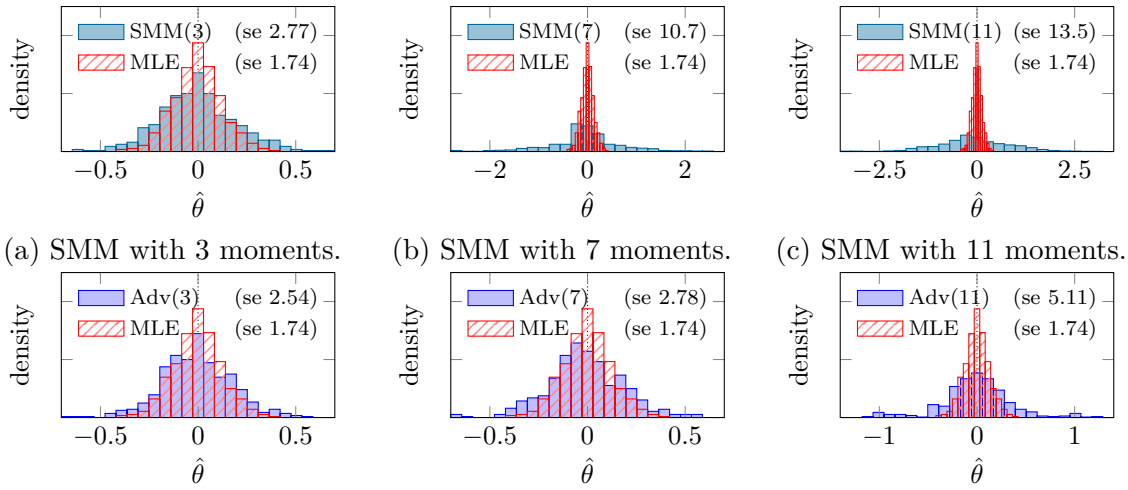
Figure 4: The logistic location model with increasing numbers of inputs. The curvature of the cross-entropy loss is very close to the log likelihood up to 7 moments and is still good for 11 moments.

3.1.3 Comparison with SMM

Finally, we compare the adversarial estimator with SMM. As discussed, the adversarial estimator with a logistic discriminator is asymptotically equivalent to SMM. However, it is known that stacking up many moments yields poor finite-sample performance of SMM. To compare our estimator in this regard, the logistic location model is a particularly interesting one. Unlike the normal distribution, the sample average is not a sufficient statistic for the mean of a logistic distribution. Indeed, the collection of order statistics is known to be a minimal sufficient statistic. Technically speaking, therefore, the higher-order moments $\mathbb{E}[X_i^2]$, $\mathbb{E}[X_i^3]$, ... do contribute to identifying the mean. This motivates the following exercise.

For SMM, we consider matching (1) three moments $\mathbb{E}[X_i]$, $\mathbb{E}[X_i^2]$, $\mathbb{E}[X_i^3]$, (2) seven moments $\mathbb{E}[X_i], \ldots, \mathbb{E}[X_i^7]$, and (3) eleven moments $\mathbb{E}[X_i], \ldots, \mathbb{E}[X_i^{11}]$. Since the optimally-weighted SMM beats the unweighted SMM in all cases in our simulation, we only present the optimally-weighted SMM for comparison; the weights are estimated with the actual sample. For the adversarial estimator, we use the same set of moments as the inputs to the discriminator. In particular, the discriminator is the logistic classifier of the form $D(x;\lambda) = \Lambda(\lambda_0 + \lambda_1 x + \cdots + \lambda_d x^d)$ for $d = 3, 7, 11$ parameterized by $\lambda \in \mathbb{R}^{1+d}$. In contrast to the one in Section 3.1.1, this discriminator is "misspecified" but is good enough to yield a reasonable estimator for θ. As discussed in Appendix S.1, the optimally-weighted SMM is asymptotically equivalent to the adversarial estimator with this choice of discriminator. However, their finite-sample properties are subject to debate. For this exercise, we decrease the sample sizes to $n = m = 200$ to emphasize the finite-sample performance.

Figure 4 shows the plots of the cross-entropy loss and the log likelihood for varying



(a) SMM with 3 moments. (b) SMM with 7 moments. (c) SMM with 11 moments.

(d) Adversarial estimator with 3 moments. (e) Adversarial estimator with 7 moments. (f) Adversarial estimator with 11 moments.

Figure 5: The logistic location model with increasing numbers of inputs. Precision of the optimally-weighted SMM rapidly deteriorates as the number of moments increases. The adversarial estimator is much less sensitive. The standard errors (se) are multiplied by $\sqrt{n}$.

numbers of inputs. It is noteworthy that the curvature of the estimated loss $\mathbb{M}_\theta(\hat{D}_\theta)$ is very close to the oracle one up to seven moments. We see nonnegligible deviation of the curvature for eleven moments but, as we see below, it is still sharp enough to yield a much better estimator than SMM.

The first row of Figure 5 shows the histogram of the optimally-weighted SMM. The horizontal scales of the figures are adjusted to match the distribution of SMM; MLE is the same for all figures and serves as the reference point. We see that the precision of SMM deteriorates quickly as the number of moments increases. For eleven moments, the standard error is eight times as large as MLE. The second row of Figure 5 presents the adversarial estimator. Even for seven inputs, the adversarial estimator is as tight as the MLE, and for eleven moments, it is still comparable (three-times larger standard error). This shows that the adversarial estimator is less sensitive to the number of moments than is SMM. This can be an advantage especially when we do not know which moments to match.

We also note that since the moments are highly correlated, the estimation of the discriminator gives warnings of multicollinearity, but it does not impair the quality of the subsequent estimator $\hat{\theta}$. This is insightful for a more general neural network

discriminator since neural network weights are not identified uniquely. This observation is in line with our theory which depends on the quality of the estimator $\hat{D}_\theta$ for D_θ but not on the quality of the estimator $\hat{\lambda}_\theta$ for λ_θ.

The improvement of our method relative to SMM is analogous to the improvement of empirical likelihood relative to GMM (Imbens, 2002). SMM, like GMM, suffers from substantial bias when the number of moments is large; our method, like empirical likelihood, has better finite-sample and large-sample properties at the expense of computational cost. The idea of both comes from treating the nuisance component as a kind of a nonparametric maximum likelihood problem. Meanwhile, both SMM and GMM retain the advantage of simplicity to easily accommodate time series settings.

3.2 The Roy Model

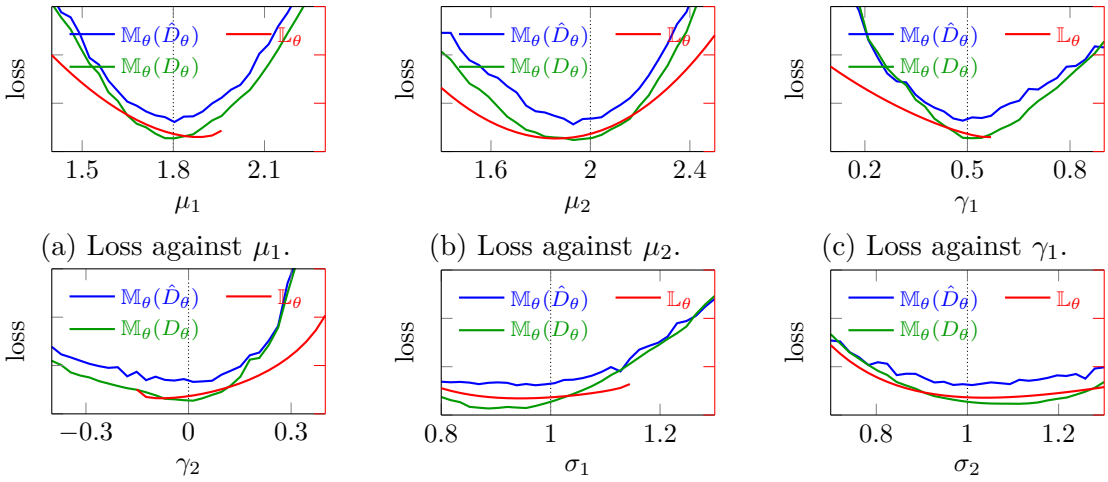
We consider the following model of self-selection, for which the likelihood is not tractable under some configurations of the parameter values. Suppose there are two sectors and two periods. In each period, an agent chooses the sector to work in to maximize her present and discounted future expected wages. The wage w_{i1s} for agent i in period 1 in sector s is determined by $\log w_{i1s} = \mu_s + \varepsilon_{i1s}$, and the wage w_{i2s} for agent i in period 2 in sector s by $\log w_{i2s} = \mu_s + \gamma_s \mathbb{1}\{d_{i1} = s\} + \varepsilon_{i2s}$ where d_{i1} is the sector choice of agent i in period 1. The parameter μ_s represents the base wage in sector s and γ_s the returns to experience in sector s. The error terms are observable to the agent in respective periods (so she observes $\varepsilon_{i1\cdot}$ in period 1 and $\varepsilon_{i2\cdot}$ in period 2) and are distributed as

$$\begin{bmatrix} \varepsilon_{i11} \\ \varepsilon_{i12} \\ \varepsilon_{i21} \\ \varepsilon_{i22} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho_s\sigma_1\sigma_2 & \rho_t\sigma_1^2 & \rho_s\rho_t\sigma_1\sigma_2 \\ \rho_s\sigma_1\sigma_2 & \sigma_2^2 & \rho_s\rho_t\sigma_1\sigma_2 & \rho_t\sigma_2^2 \\ \rho_t\sigma_1^2 & \rho_s\rho_t\sigma_1\sigma_2 & \sigma_1^2 & \rho_s\sigma_1\sigma_2 \\ \rho_s\rho_t\sigma_1\sigma_2 & \rho_t\sigma_2^2 & \rho_s\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \right).$$

Observable to us is the quartet $X_i = (\log w_{i1}, d_{i1}, \log w_{i2}, d_{i2})$ of realized log wages and sector choices in both periods. They are functions of above variables by $w_{i1} = w_{i1d_{i1}}$, $d_{i1} = \arg\max_{s\in\{1,2\}} w_{i1s} + \beta\mathbb{E}[w_{i2} \mid d_{i1} = s]$, $w_{i2} = w_{i2d_{i2}}$, and $d_{i2} = \arg\max_{s\in\{1,2\}} w_{i2s}$ where β is the discount factor. We fix $\beta = 0.9$, so β is not a free parameter.

3.2.1 Comparison with MLE

As a first exercise, we show that the adversarial estimator has a computational advantage over MLE. To this end, we fix $\rho_t = 0$ to have a tractable likelihood.



(a) Loss against μ_1. (b) Loss against μ_2. (c) Loss against γ_1.

(d) Loss against γ_2. (e) Loss against σ_1. (f) Loss against σ_2.

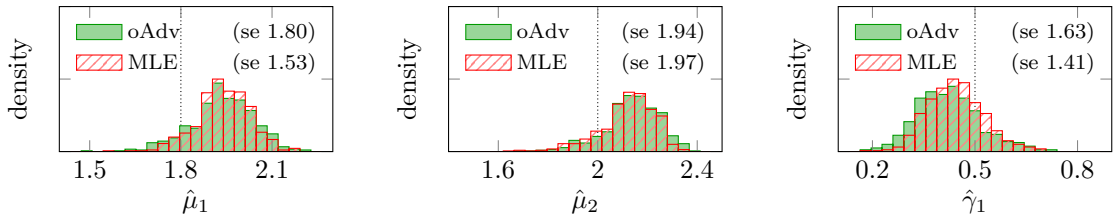
Figure 6: The loss for the Roy model. In the region where $\mathbb{L}_\theta$ is not plotted, the real data X is not supported on the corresponding model P_θ, so $\mathbb{L}_\theta = \infty$. The figure for ρ_s is omitted.

Thus, the parameter of interest is $\theta = (\mu_1, \mu_2, \gamma_1, \gamma_2, \sigma_1, \sigma_2, \rho_s)$. The true value is $\theta_0 = (1.8, 2, 0.5, 0, 1, 1, 0.5)$. We set the sample sizes at $n = m = 300$.

Although the likelihood is available, the correct functional form of D_θ is not easy to derive. Therefore, we skip the correctly specified discriminator and use the neural network discriminator for the feasible adversarial estimator. The neural network has one hidden layer with ten nodes with a hyperbolic tangent activation function. The input is X_i without transformation. The output layer uses a sigmoid function.

Note that if $w_{i11} + \beta\mathbb{E}[w_{i2} \mid d_{i1} = 1] < \beta\mathbb{E}[w_{i2} \mid d_{i1} = 2]$, there is no way that agent i chooses sector 1 in period 1. Therefore, if we see a pair $(w_{i1}, d_{i1}) = (w_{i11}, 1)$ that satisfies this inequality for a particular θ, this observation is not supported by P_θ. This is indeed a common phenomenon. Figure 6 plots the loss and the log likelihood against each parameter, holding all other parameters to the truth. The range of the figures reflects the range of MLE and the adversarial estimator. In this "relevant" region, we see that $\mathbb{L}_\theta$ sometimes breaks off; this is because the discontinued part does not support the real data so $\mathbb{L}_\theta$ is infinity.

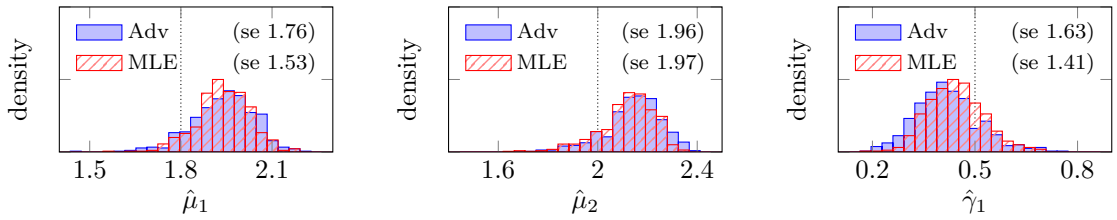
Aside from possible inefficiency, this is not a problem for MLE insofar as the likelihood maximizer can be found. However, there may be difficulties when the initial value of θ does not support the real data. In fact, if we do not pick the initial value carefully, Matlab's `fminsearch` wanders around the unsupported region and returns



(a) Oracle adversarial estimator for μ_1.

(b) Oracle adversarial estimator for μ_2.

(c) Oracle adversarial estimator for γ_1.



(d) Adversarial estimator for μ_1.

(e) Adversarial estimator for μ_2.

(f) Adversarial estimator for γ_1.

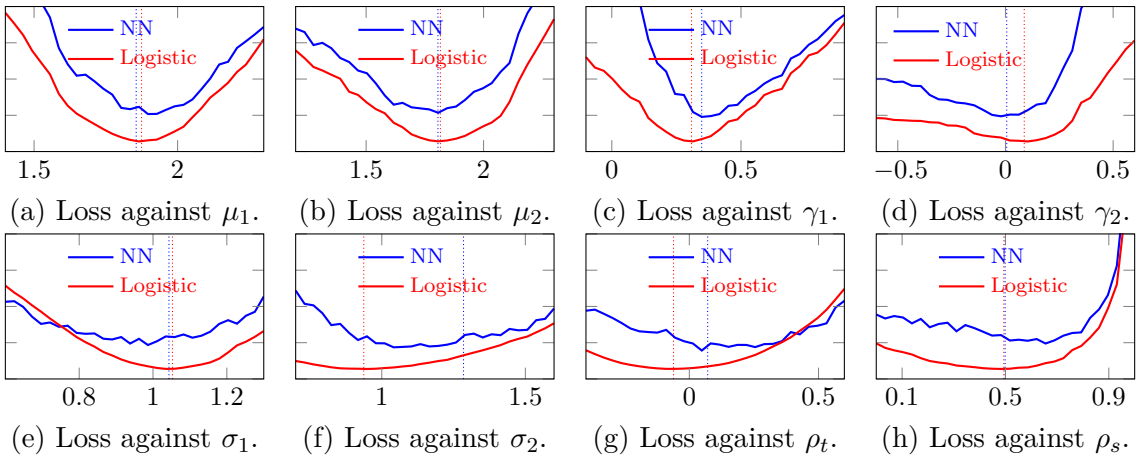
Figure 7: The oracle adversarial estimator and the adversarial estimator for the Roy model with $\rho_t = 0$. Figures for other parameters are omitted.

a meaningless value after the iteration limit is reached. Meanwhile, Figure 6 indicates that such a problem does not occur for the cross-entropy loss; indeed, $\mathbb{M}_\theta(D_\theta)$ extends a nice curve throughout the "unsupported" region. The key is in the robustness of the sample Jensen–Shannon divergence,

$$\frac{1}{2}\mathbb{M}_\theta(D_\theta) = \frac{1}{2n}\sum_{i=1}^{n}\log\frac{p_0(X_i)}{p_0(X_i)+p_\theta(X_i)} + \frac{1}{2m}\sum_{i=1}^{m}\log\frac{p_\theta(X_{i,\theta})}{p_0(X_{i,\theta})+p_\theta(X_{i,\theta})}.$$

When a single observation X_i is not on the support of p_θ, the corresponding fraction is 1, which does not ruin the sum so we can still calculate a meaningful distance using remaining observations; hence the curve continues. Moreover, even if the entire sample $\{X_i\}$ goes outside the support, the divergence still works as long as (some of) the synthetic data are on the support of p_0 and the second sum is informative. It is only when both the entire real sample $\{X_i\}$ and the synthetic sample $\{X_{i,\theta}\}$ are outside the supports of p_θ and p_0, respectively, that the Jensen–Shannon divergence gets fixated at 0 and loses guidance on θ_0.[11] This is the intuition for why the adversarial estimator does not suffer from the support issue in the Roy model. We can also see this as a virtue of estimating the likelihood ratio as opposed to the raw likelihood.

[11] If the supports of p_0 and $\{p_\theta\}$ are fully disjoint, the Jensen–Shannon projection θ_0 is not defined.



(a) Loss against μ_1. (b) Loss against μ_2. (c) Loss against γ_1. (d) Loss against γ_2.

(e) Loss against σ_1. (f) Loss against σ_2. (g) Loss against ρ_t. (h) Loss against ρ_s.

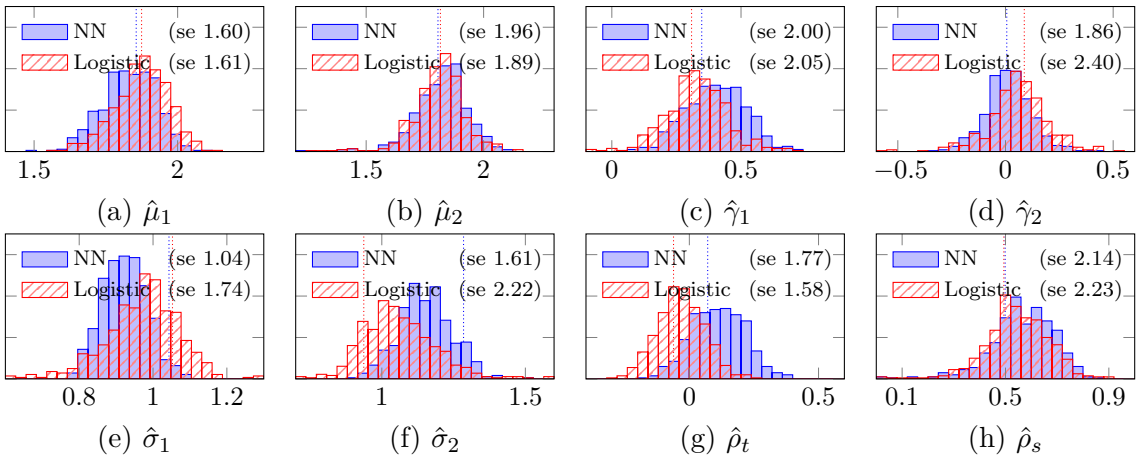
Figure 8: The logistic loss is smooth and corroborates orthogonality. The neural network loss also indicates orthogonality, albeit a bit rough.

However, the cross-entropy loss breaks down when the distributions become completely disjoint, which can often be an issue in high-dimensional data such as images. As the Wasserstein distance is known to be capable of handling these distributions, the GAN literature in computer science has mostly moved to the Wasserstein loss. Importantly, the results of this paper do not cover the Wasserstein loss. For the use of Wasserstein GAN in economics, we refer the reader to Athey et al. (2020).

Figure 6 also illustrates that, despite having discrete observables (sector choices), the objective function is smooth thanks to continuous observables (wages), so there is no need for smoothing even when we employ gradient-based methods. The resulting estimators are comparable with MLE just as in the previous examples (Figure 7).

3.2.2 Case with Intractable Likelihood

Now, we illustrate the whole procedure of estimation and inference using the Roy model with intractable likelihood. Let us consider the same model as in Section 3.2.1 but without assuming $\rho_t = 0$, so the parameter is $\theta = (\mu_1, \mu_2, \gamma_1, \gamma_2, \sigma_1, \sigma_2, \rho_t, \rho_s)$. The true values are the same as before. We first pre-estimate the model with a logistic discriminator and then estimate it with a neural network discriminator using the logistic estimator as the initial value. Since it is natural to speculate that identification comes from the moments of the log wages, we consider the logistic discriminator of the form $D(\log w_1, d_1, \log w_2, d_2; \lambda) = \Lambda(\lambda_0 + \lambda_1 \log w_1 + \lambda_2 d_1 + \lambda_3 \log w_2 + \lambda_4 d_2 + \lambda_5 (\log w_1)^2 + \lambda_6 (\log w_2)^2 + \lambda_7 \log w_1 \log w_2)$.



(a) $\hat{\mu}_1$ (b) $\hat{\mu}_2$ (c) $\hat{\gamma}_1$ (d) $\hat{\gamma}_2$

(e) $\hat{\sigma}_1$ (f) $\hat{\sigma}_2$ (g) $\hat{\rho}_t$ (h) $\hat{\rho}_s$

Figure 9: The bootstrap samples and bootstrap standard errors (multiplied by $\sqrt{n}$).

As the curvature of the logistic loss is quite sharp, we may in practice stop here and go with the logistic estimator. For illustration, we move on to the neural network discriminator with the same configuration as in Section 3.2.1. The loss is plotted as the blue line in Figure 8. The vertical blue dotted lines indicate the neural network estimator.[12]

Next, we use bootstrap to compute the standard errors. We resample both the actual data $\{X_i\}_{i=1}^n$ and the simulation shocks $\{Z_i\}_{i=1}^m$ with replacement, pre-estimate the model with the logistic discriminator, and then estimate the model with the neural network discriminator. Figure 9 shows the bootstrap samples of the logistic estimator (red) and the neural network estimator (blue). We see that the neural network estimator is comparable to the logistic estimator. Note that the neural network discriminator takes as inputs the raw quartet but not the higher-order moments fed into the logistic discriminator. Thus the neural network with one hidden layer of ten nodes "figures out" the correct moments to match and produces an estimator comparable with the logistic discriminator whose inputs are deliberately chosen.

Table 1 presents the estimates and the standard errors (not multiplied by $\sqrt{n}$). Along with the adversarial estimator, we present the results of SMM. SMM matches the same seven moments as the inputs to the logistic discriminator, namely $\mathbb{E}[\log w_{i1}]$, $\mathbb{E}[d_{i1}]$, $\mathbb{E}[\log w_{i2}]$, $\mathbb{E}[d_{i2}]$, $\mathbb{E}[(\log w_{i1})^2]$, $\mathbb{E}[(\log w_{i2})^2]$, and $\mathbb{E}[\log w_{i1} \log w_{i2}]$. The optimal weights are estimated with the actual data. We see that the adversarial estimators

[12]Note that the global minimizer is not the same as the local minimizers of the figures since the other parameters are fixed at the logistic estimator.

Table 1: Estimates and bootstrap standard errors for the Roy model for one replication.

	μ_1	μ_2	γ_1	γ_2	σ_1	σ_2	ρ_t	ρ_s
Logistic D	1.88	1.81	0.33	0.06	0.97	1.06	−0.03	0.54
	(0.09)	(0.11)	(0.12)	(0.14)	(0.10)	(0.13)	(0.09)	(0.13)
Neural network D	1.83	1.82	0.40	0.01	0.92	1.15	0.11	0.57
	(0.09)	(0.11)	(0.12)	(0.11)	(0.06)	(0.09)	(0.10)	(0.12)
SMM	1.88	1.81	0.33	0.06	0.95	1.08	−0.03	0.56
	(0.10)	(0.12)	(0.13)	(0.16)	(0.11)	(0.14)	(0.09)	(0.14)
Truth	1.80	2.00	0.50	0.00	1.00	1.00	0.00	0.50

are slightly more precise than the SMM.

3.3 Challenges of the Adversarial Estimator

Not every aspect of our method is superior to alternatives. First, the theoretical results in this paper do not cover time series data. The Roy model has a dynamic choice of individuals, but we have many i.i.d. observations of individuals. This is not to say that the adversarial framework cannot be extended thereto, but it would require a careful design of the discriminator to incorporate the structure of the serial correlation.

Second, the adversarial estimator can be time-consuming. The adversarial estimator with a logistic discriminator is as fast as SMM, but one with a neural network discriminator can take a long time to train. In the logistic location model, both MLE and the adversarial estimator with a logistic discriminator take less than a second, while the adversarial estimator with a neural network discriminator takes about 30 seconds on a laptop without a GPU or parallelization. For this, we recommend pre-estimation with a logistic discriminator or other existing methods to start with a good initial value.

The third drawback is a possible roughness of the loss surface. As seen in Section 3.1.1, a logistic discriminator tends to yield a very smooth objective function (Figure 1) while a neural network discriminator may sometimes get bumpy and have spurious local minima (Figure 2). Some degree of roughness can be smoothed with the choice of a training method or an increased number of iterations; additionally, we can estimate the discriminator several times and take the average and/or use an op-

timization method tailored for noisy functions. If the initial value is good enough, we may also employ grid search in the neighborhood to skip estimation of the gradient. At any rate, we recommend plotting the loss surface before computing the estimator.

Fourth, being comparable with MLE, the asymptotic variance of the adversarial estimator depends on the score and Hessian (Theorem 3), which is not easy to compute given the intractable likelihood. Therefore, we may resort to resampling methods like bootstrap to obtain a variance estimator, which can cost additional time.

4 STATISTICAL PROPERTIES

This section states the asymptotic properties of our estimator. For more general results, we refer the reader to the earlier version in February 2022, Kaji et al. (2022).

Let $Z_i \sim P_Z$ be a common random shock used in simulation. The simulated observation $X_{i,\theta} \sim P_\theta$ is then constructed by transforming Z_i through a map, $X_{i,\theta} = T_\theta(Z_i)$. For a function f, the sample averages of $f(X_i)$ and $f(X_{i,\theta})$ are denoted by $\mathbb{P}_0 f \coloneqq \frac{1}{n}\sum_{i=1}^n f(X_i)$ and $\mathbb{P}_\theta f \coloneqq \frac{1}{m}\sum_{i=1}^m f(X_{i,\theta})$. Their population counterparts are denoted as $P_0 f \coloneqq \int f(x)dP_0$ and $P_\theta f \coloneqq \int f(x)dP_\theta$. We denote the population objective function by $M_\theta(D) \coloneqq P_0 \log D + P_\theta \log(1-D)$ as well as the previously defined sample objective function $\mathbb{M}_\theta(D) \coloneqq \mathbb{P}_0 \log D + \mathbb{P}_\theta \log(1-D)$. We also define the distance on Θ by $h(\theta_1,\theta_2) \coloneqq \sqrt{\int(\sqrt{p_{\theta_1}} - \sqrt{p_{\theta_2}})^2}$.

Suppose that observables can be written as $X_i = (Y_i, W_i)$ where θ affects only the conditional distribution of Y_i given W_i. Such W_i is called the covariate. In the maximum likelihood literature, it is known that an efficient estimator is obtained by maximizing the conditional likelihood of Y_i given W_i, so the marginal distribution of W_i can be left unspecified. The same observation holds true in the adversarial framework. Namely, the oracle discriminator D_θ does not depend on the marginal distribution of W_i, so the distributions P_0 and P_θ can be regarded as specifying only the conditional distribution of Y_i given W_i. In our theory, we save notational complexity by allowing this implicitly. One possible complication this might bring is the method to draw covariates for the simulated data. In Section 3.1.2, we set $n = m$ and use the same sets of covariates in the actual data. Another possibility is to bootstrap the covariates.

4.1 Consistency

The adversarial estimator is consistent if the estimated loss $\mathbb{M}_\theta(\hat{D}_\theta)$ converges uniformly to the oracle loss $\mathbb{M}_\theta(D_\theta)$ and $\hat{\theta}$ finds a global minimizer. As the maximized cross-entropy loss is effectively bounded between $2\log(1/2)$ and 0, uniform convergence on Θ is not an unreasonable assumption.

Theorem 1 (Consistency of generator)**.** *Suppose that for every open $G \subset \Theta$ containing θ_0, we have $\inf_{\theta \notin G} M_\theta(D_\theta) > M_{\theta_0}(D_{\theta_0})$, that $\{\log D_\theta : \theta \in \Theta\}$ and $\{\log(1 - D_\theta) \circ T_\theta : \theta \in \Theta\}$ are P_0- and P_Z-Glivenko–Cantelli respectively, that $\sup_{\theta\in\Theta} |\mathbb{M}_\theta(\hat{D}_\theta) - \mathbb{M}_\theta(D_\theta)| \to 0$ in probability, and that $\hat{\theta}$ satisfies $\mathbb{M}_{\hat{\theta}}(\hat{D}_{\hat{\theta}}) \leq \inf_{\theta\in\Theta} \mathbb{M}_\theta(\hat{D}_\theta) + o_P^*(1)$. Then, $h(\hat{\theta}, \theta_0) \to 0$ in probability.*

This theorem does not assume that the generative model is parametric, so it also applies to possibly "nonparametric" generators.

4.2 Rate of Convergence

To obtain a rate of convergence of the generator, we assume that the structural model is parametric.

Assumption 1 (Parametric generative model)**.** Θ is (a subset of) a Euclidean space; p_θ is differentiable in θ at every $\theta \in \Theta$ for every $x \in \mathcal{X}$ with the derivative continuous in both x and θ; the maximum eigenvalue of the Fisher information $I_\theta = P_\theta \dot{\ell}_\theta \dot{\ell}_\theta^\top$ is bounded uniformly in $\theta \in \Theta$; the minimum eigenvalue of I_θ is bounded away from 0 uniformly in $\theta \in \Theta$. The same is assumed for the "inverted" structural model $\tilde{\mathcal{P}}_\theta = \{((p_0/p_\theta) \circ T_\theta)p_Z : \theta \in \Theta\}$. □

We next assume that the synthetic sample size m grows faster than n.

Assumption 2 (Growing synthetic sample size)**.** $n/m \to 0$. □

The next assumption ensures that the estimation procedure finds a good minimum and that the derivative of the estimated loss converges to that of the oracle. The first property hinges on the estimation procedure employed, the tolerance level, etc. The second property is used in semiparametric M-estimation to obtain a regular estimator orthogonal to nuisance estimation (e.g., Klein and Spady, 1993). We revisit the plausibility of this condition in Section 4.2.1.

Assumption 3 (Approximately minimizing generator and orthogonality)**.** There exists a sequence of open balls $G_n \coloneqq \{\theta \in \Theta : h(\theta, \theta_0) < \eta_n\}$ such that $\eta_n\sqrt{n} \to \infty$, $\mathbb{M}_{\hat{\theta}}(\hat{D}_{\hat{\theta}}) \leq \inf_{\theta \in G_n} \mathbb{M}_\theta(\hat{D}_\theta) + o_P^*(n^{-1})$, and $\inf_{\theta \in G_n}[\mathbb{M}_{\hat{\theta}}(\hat{D}_{\hat{\theta}}) - \mathbb{M}_\theta(\hat{D}_\theta)] - [\mathbb{M}_{\hat{\theta}}(D_{\hat{\theta}}) - \mathbb{M}_\theta(D_\theta)] = o_P^*(n^{-1})$. □

The next assumption posits a stronger identification condition than in Theorem 1 that ensures a quadratic curvature at θ_0; this is implied by the positive definiteness of $\tilde{I}_{\theta_0}$ in Assumption 5. Also, it assumes that P_0 is "close enough" to P_{θ_0} in the sense that convergence of θ to θ_0 takes place on the support of P_0.

Assumption 4 (Smooth synthetic data generation and overlapping support)**.** There exists open $G \subset \Theta \subset \mathbb{R}^k$ containing θ_0 in which $M_\theta(D_\theta) - M_{\theta_0}(D_{\theta_0}) \gtrsim h(\theta, \theta_0)^2$. Also, $h(\theta, \theta_0)^2 = O(\int D_{\theta_0}(\sqrt{p_{\theta_0}} - \sqrt{p_\theta})^2)$ as $\theta \to \theta_0$. □

Theorem 2 (Rate of convergence of generator)**.** *Under Assumptions 1 to 4,* $h(\hat{\theta}, \theta_0) = O_P^*(n^{-1/2})$.

4.2.1 On Assumption 3

The second condition of Assumption 3, which we call orthogonality, is essential in the rate of convergence for $\hat{\theta}$ in Theorem 2. Even in the best scenario, we can only expect $\mathbb{M}_\theta(\hat{D}_\theta) - \mathbb{M}_\theta(D_\theta) = O_P(n^{-1})$, so the convergence of $\hat{D}_\theta$ alone does not grant orthogonality. The key to satisfying it is, therefore, some extent of the convergence of the *derivative* of $\hat{D}_\theta$ with respect to θ to that of D_θ. Note that this is different from the derivative of $\hat{D}_\theta$ with respect to x, so it does not follow from the convergence of the derivative of a nonparametrically estimated function. Rather, it is the structure of the nested optimization that brings about orthogonality.

Take the logistic discriminator $D(x; \lambda) = \Lambda(x^\top \lambda)$ as considered in Section 3. We can check that orthogonality holds if the following conditions are met. Let $\mathbb{E}_n f(X) \coloneqq \frac{1}{n}\sum_{i=1}^n f(X_i)$ and $\mathbb{E}_m f(X_\theta) \coloneqq \frac{1}{m}\sum_{i=1}^m f(X_{i,\theta})$ and denote the differentiation with respect to a row vector $\theta^\top$ by a dot, e.g., $\dot{\lambda}_\theta = \frac{\partial}{\partial \theta^\top}\lambda_\theta$.

1. (Smooth model) T_θ is continuously differentiable in θ for every $x \in \tilde{\mathcal{X}}$, so X_θ is continuously differentiable in θ.
2. (Finite moments) $\mathbb{E}[XX^\top]$ is positive definite; $\mathbb{E}[\|X\|^4]$, $\mathbb{E}[\|X_\theta\|^4]$, $\mathbb{E}[\|\dot{X}_\theta\|^2]$, and $\mathbb{E}[\|X_\theta\|^2\|\dot{X}_\theta\|^2]$ are bounded uniformly over θ; $\mathbb{E}_m[\|X_\theta\|^2]$, $\mathbb{E}_m[\|\dot{X}_\theta\|]$, and $\mathbb{E}_m[\|X_\theta\|\|\dot{X}_\theta\|]$ converge uniformly in θ.

3. (Smooth discriminator) λ_θ is continuously differentiable in θ.
4. (Exact maximizer) $\hat{\lambda}_\theta$ is the exact maximizer of $\mathbb{M}_\theta(D(\cdot;\lambda))$ in that the FOC for $\hat{\lambda}_\theta$ is exactly zero for every $\theta \in \Theta$.
5. (Uniform convergence rate of discriminator) $\sup_\theta \|\hat{\lambda}_\theta - \lambda_\theta\| = O_P(n^{-1/2})$.

For ease of notation, we assume that λ and θ are one-dimensional; however, the argument below applies equally to the vector case. The FOC for $\hat{\lambda}_\theta$ yields $\mathbb{E}_n[(1 - \Lambda(X\hat{\lambda}_\theta))X] - \mathbb{E}_m[\Lambda(X_\theta\hat{\lambda}_\theta)X_\theta] = 0$. This holds for every θ, so we may differentiate both sides by θ, which can be solved for the derivative of $\hat{\lambda}_\theta$ with respect to θ,

$$\dot{\hat{\lambda}}_\theta = -(\mathbb{E}_n[\Lambda(1-\Lambda)(X\hat{\lambda}_\theta)X^2] + \mathbb{E}_m[\Lambda(1-\Lambda)(X_\theta\hat{\lambda}_\theta)X_\theta^2])^{-1} \\ (\mathbb{E}_m[\Lambda(1-\Lambda)(X_\theta\hat{\lambda}_\theta)X_\theta\dot{X}_\theta]\hat{\lambda}_\theta + \mathbb{E}_m[\Lambda(X_\theta\hat{\lambda}_\theta)\dot{X}_\theta]).$$

Note that λ_θ satisfies the population FOC, which leads to the population counterpart of the same expression, so $\dot{\hat{\lambda}}_\theta$ is consistent for $\dot{\lambda}_\theta$. Moreover, by the uniform convergence assumptions, we deduce $\sup_\theta \|\dot{\hat{\lambda}}_\theta - \dot{\lambda}_\theta\| = O_P(n^{-1/2})$. Thus, the derivative of the discriminator converges.

To derive orthogonality, we first Taylor-expand it in λ around $\hat{\lambda}_\theta$. In doing so, the first-order term can be ignored thanks to Condition 4. For arbitrary θ,

$$\begin{aligned}
&\mathbb{M}_\theta(D(\cdot;\lambda_\theta)) - \mathbb{M}_\theta(D(\cdot;\hat{\lambda}_\theta)) \\
&= [\mathbb{E}_n \log\Lambda(X\lambda_\theta) + \mathbb{E}_m \log(1-\Lambda)(X_\theta\lambda_\theta)] - [\mathbb{E}_n \log\Lambda(X\hat{\lambda}_\theta) + \mathbb{E}_m \log(1-\Lambda)(X_\theta\hat{\lambda}_\theta)] \\
&= \tfrac{1}{2}(\hat{\lambda}_\theta - \lambda_\theta)^2[-\mathbb{E}_n\Lambda(1-\Lambda)(X\hat{\lambda}_\theta)X^2 + \mathbb{E}_m\Lambda(1-\Lambda)(X_\theta\hat{\lambda}_\theta)X_\theta^2] + o_P((\hat{\lambda}_\theta - \lambda_\theta)^2).
\end{aligned}$$

Next, we expand it further in θ around $\hat{\theta}$.

$$\begin{aligned}
&[\mathbb{M}_\theta(D(\cdot;\lambda_\theta)) - \mathbb{M}_\theta(D(\cdot;\hat{\lambda}_\theta))] - [\mathbb{M}_{\hat{\theta}}(D(\cdot;\lambda_{\hat{\theta}})) - \mathbb{M}_{\hat{\theta}}(D(\cdot;\hat{\lambda}_{\hat{\theta}}))] \\
&= -(\hat{\lambda}_{\hat{\theta}} - \lambda_{\hat{\theta}})(\dot{\hat{\lambda}}_{\hat{\theta}} - \dot{\lambda}_{\hat{\theta}})(\theta - \hat{\theta})[\mathbb{E}_n\Lambda(1-\Lambda)(X\hat{\lambda}_\theta)X^2 - \mathbb{E}_m\Lambda(1-\Lambda)(X_\theta\hat{\lambda}_\theta)X_\theta^2] \\
&\quad - \tfrac{1}{2}(\hat{\lambda}_\theta - \lambda_\theta)^2(\theta - \hat{\theta})\mathbb{E}_n\Lambda(1-\Lambda)(1-2\Lambda)(X\hat{\lambda}_{\hat{\theta}})X^3\dot{\hat{\lambda}}_{\hat{\theta}} \\
&\quad + \tfrac{1}{2}(\hat{\lambda}_\theta - \lambda_\theta)^2(\theta - \hat{\theta})\mathbb{E}_m\Lambda(1-\Lambda)(1-2\Lambda)(X_{\hat{\theta}}\hat{\lambda}_{\hat{\theta}})X_{\hat{\theta}}^3\dot{\hat{\lambda}}_{\hat{\theta}} \\
&\quad + \tfrac{1}{2}(\hat{\lambda}_\theta - \lambda_\theta)^2(\theta - \hat{\theta})\mathbb{E}_m\Lambda(1-\Lambda)(1-2\Lambda)(X_{\hat{\theta}}\hat{\lambda}_{\hat{\theta}})X_{\hat{\theta}}^2\dot{X}_{\hat{\theta}}\hat{\lambda}_{\hat{\theta}} \\
&\quad + (\hat{\lambda}_\theta - \lambda_\theta)^2(\theta - \hat{\theta})\mathbb{E}_m\Lambda(1-\Lambda)(X_{\hat{\theta}}\hat{\lambda}_{\hat{\theta}})X_{\hat{\theta}}\dot{X}_{\hat{\theta}} + o_P((\hat{\lambda}_\theta - \lambda_\theta)^2(1 + |\hat{\theta} - \theta|)).
\end{aligned}$$

This is $O_P(n^{-1})$ for fixed θ, so we can take a shrinking neighborhood of θ around θ_0 that contains $\hat{\theta}$ to make the supremum of this $o_P(n^{-1})$, yielding orthogonality.

If the neighborhood shrinks only slightly slower than $n^{-1/2}$, then convergence of $\hat{\lambda}_\theta$ and $\dot{\hat{\lambda}}_\theta$ can be relaxed to as slow as $o_P(n^{-1/4})$ if possibly a few more degrees of differentiability and finite moments are granted. It is also straightforward to relax the exact FOC condition to allow for errors of negligible order and to allow for nonlinear but parametric logistic discriminators, such as small neural networks. An interesting conclusion of this is that the curvature of the estimated loss converges faster than the level, as observed throughout Section 3.

For a general nonparametric discriminator, it is not trivial to obtain a similar low-level condition. Appendix S.2 develops conditions for $\hat{D}_\theta$ to converge faster than $n^{-1/4}$ (pointwise in θ), which seems necessary but is not sufficient to derive orthogonality.[13] In Section 3, the plots of $\mathbb{M}_\theta(\hat{D}_\theta)$ confirm orthogonality in examples with or without differentiability.

4.3 Asymptotic Distribution

To derive the asymptotic distribution of the adversarial estimator, we need the structural model to be differentiable as in maximum likelihood.

Assumption 5 (Twice differentiability)**.** The parameter space Θ is (a subset of) a Euclidean space $\mathbb{R}^k$. The structural model $\{P_\theta : \theta \in \Theta\}$ has a likelihood that is twice differentiable in θ at θ_0 for every $x \in \mathcal{X}$ with the derivatives continuous in both x and θ. The Fisher information matrix $I_{\theta_0} \coloneqq P_{\theta_0}\dot{\ell}_{\theta_0}\dot{\ell}_{\theta_0}^\top = -P_{\theta_0}\ddot{\ell}_{\theta_0}$ and the matrix $\tilde{I}_{\theta_0} \coloneqq 2P_{\theta_0}(D_{\theta_0}\dot{\ell}_{\theta_0}\dot{\ell}_{\theta_0}^\top + (\ddot{\ell}_{\theta_0} + \dot{\ell}_{\theta_0}\dot{\ell}_{\theta_0}^\top)\log(1 - D_{\theta_0}))$ are positive definite. T_θ is continuously differentiable in θ for every $x \in \mathcal{X}$ and P_0 has a likelihood that is continuously differentiable in x. □

Remark. The score and Hessian are related to the oracle discriminator D_θ by $\dot{\ell}_\theta = \frac{1}{D_\theta}\frac{\partial \log(1-D_\theta)}{\partial \theta} = -\frac{1}{1-D_\theta}\frac{\partial \log D_\theta}{\partial \theta}$ and $\ddot{\ell}_\theta + \dot{\ell}_\theta\dot{\ell}_\theta^\top = \frac{1}{1-D_\theta}[\frac{\partial \log D_\theta}{\partial \theta}\frac{\partial \log D_\theta}{\partial \theta^\top} - \frac{\partial^2 \log D_\theta}{\partial \theta \partial \theta^\top}]$.

Theorem 3 (Asymptotic distribution of generator)**.** *Under the conclusion of Theorem 2 and Assumptions 2, 3, and 5,*

$$\sqrt{n}(\hat{\theta} - \theta_0) = 2\tilde{I}_{\theta_0}^{-1}\sqrt{n}[\mathbb{P}_0(1 - D_{\theta_0})\dot{\ell}_{\theta_0} - \mathbb{P}_{\theta_0}D_{\theta_0}\dot{\ell}_{\theta_0} - \tilde{\mathbb{P}}_0\tau_n] + o_P^*(1) \rightsquigarrow N(0, \tilde{I}_{\theta_0}^{-1}V\tilde{I}_{\theta_0}^{-1}).$$

where $V \coloneqq \lim_{n\to\infty} 4P_{\theta_0}D_{\theta_0}(1 - D_{\theta_0})\dot{\ell}_{\theta_0}\dot{\ell}_{\theta_0}^\top$.

[13] In a similar situation where the derivative of the nuisance parameter identifies θ, Klein and Spady (1993) exploit the structure of a kernel density estimator to show the convergence of the derivative, whereby obtaining a corresponding orthogonality condition.

An efficiency result holds if the structural model is correctly specified.

Assumption 6 (Correct specification)**.** The synthetic model $\{P_\theta : \theta \in \Theta\}$ is correctly specified, that is, $P_{\theta_0} = P_0$ and $D_{\theta_0} \equiv 1/2$. □

Corollary 4 (Efficiency of generator)**.** *Under the conclusion of Theorem 3 and Assumption 6,* $\sqrt{n}(\hat{\theta} - \theta_0) = I_{\theta_0}^{-1}\sqrt{n}(\mathbb{P}_0 - \mathbb{P}_{\theta_0})\dot{\ell}_{\theta_0} + o_P^*(1) \rightsquigarrow N(0, I_{\theta_0}^{-1})$.

4.4 What If $\mathcal{D}$ Is Not Rich Enough?

Our theory assumes that $\mathcal{D}$ is a sieve that eventually is capable of representing D_θ. In finite samples, however, we do not know how well $\mathcal{D}$ approximates D_θ. Therefore, it is interesting to see what happens when $\mathcal{D}$ is not a sieve but a fixed class of functions. Although the complete treatment of this case is beyond our scope, we examine what happens to the population problem as we enrich $\mathcal{D}$, e.g., by gradually adding nodes and layers to the neural network.[14]

For simplicity, we maintain Assumptions 5, 6, and S.3 and assume that $\mathcal{D}$ contains a constant function 1/2. Let $\tilde{D}_\theta$ be the population maximizer of $M_\theta(D)$ in $\mathcal{D}$. Since $M_\theta(D) - M_\theta(D_\theta) = -2d_\theta(D, D_\theta)^2 + o(d_\theta(D, D_\theta)^2)$ by Theorem S.2, $\tilde{D}_\theta$ is equivalent to a minimizer of $d_\theta(D, D_\theta)^2$ in $\mathcal{D}$ up to $o(d_\theta(D, D_\theta)^2)$. Under Assumption 6, $\tilde{D}_{\theta_0} = D_{\theta_0} \equiv 1/2$ and $M_{\theta_0}(1/2) = M_\theta(1/2)$. By Theorem S.2,

$$\begin{aligned} M_{\theta_0}(\tilde{D}_{\theta_0}) - M_\theta(\tilde{D}_\theta) &= M_\theta(D_{\theta_0}) - M_\theta(D_\theta) + M_\theta(D_\theta) - M_\theta(\tilde{D}_\theta) \\ &= -2d_\theta(D_{\theta_0}, D_\theta)^2 + 2d_\theta(\tilde{D}_\theta, D_\theta)^2 + o(d_\theta(D_{\theta_0}, D_\theta)^2) + o(d_\theta(\tilde{D}_\theta, D_\theta)^2). \end{aligned}$$

Note that by Lemma S.6, $d_\theta(D_{\theta_0}, D_\theta)^2 = \frac{1}{2}\int \frac{p_0}{p_0+p_0}(\sqrt{p_0} - \sqrt{p_\theta})^2 + \frac{1}{2}\int \frac{p_\theta}{p_\theta+p_\theta}(\sqrt{p_0} - \sqrt{p_\theta})^2 + o(h(p_0, p_\theta)^2) = \frac{1}{2}h(p_0, p_\theta)^2 + o(h(p_0, p_\theta)^2)$. Thus, we obtain

$$M_{\theta_0}(\tilde{D}_{\theta_0}) - M_\theta(\tilde{D}_\theta) = -h(p_0, p_\theta)^2 + 2d_\theta(\tilde{D}_\theta, D_\theta)^2 + o(h(p_0, p_\theta)^2).$$

If $\mathcal{D}$ contains D_θ, the second term is zero and the Hellinger curvature allows us to estimate θ efficiently; if $\mathcal{D}$ is a singleton set that contains only 1/2, the first and second terms cancel and the objective function becomes completely flat, rendering estimation of θ impossible. Therefore, the second term represents the loss in efficiency due to the limited capacity of $\mathcal{D}$. For the regular logit case, we know that $\mathcal{D}$ is already rich

[14] The case where $\mathcal{D}$ is fixed to be the class of logistic discriminators is analyzed in Appendix S.1.

enough that the curvature admits $\sqrt{n}$-estimation. Then, as we enrich $\mathcal{D}$, it becomes more capable of minimizing $d_\theta(\tilde{D}_\theta, D_\theta)^2$, getting closer to efficiency.

5 CONCLUSION

We propose a simulation-based estimation method for structural estimation inspired by GAN. The method uses a minimax formulation between a generator given by the structural model and the adversarial discriminator given by a possibly nonparametric classifier. Under the given conditions, the estimator is $\sqrt{n}$-asymptotically normal under possible global misspecification and is efficient under correct specification.

The adversarial estimator fills the gap between SMM and MLE. When a logistic discriminator is used, the estimator is asymptotically equivalent to optimally-weighted SMM. When an oracle discriminator is used, it is asymptotically equivalent to MLE under correct specification and $m \gg n$. Simulation indicates that the estimator is robust to the curse of dimensionality compared to SMM.

APPENDIX

Proof of Theorem 1. Observe that $\mathbb{M}_{\hat{\theta}}(D_{\hat{\theta}}) - \inf_{\theta\in\Theta} \mathbb{M}_\theta(D_\theta)$ is bounded by

$$\left[\mathbb{M}_{\hat{\theta}}(\hat{D}_{\hat{\theta}}) - \inf_{\theta\in\Theta} \mathbb{M}_\theta(\hat{D}_\theta)\right] + [\mathbb{M}_{\hat{\theta}}(D_{\hat{\theta}}) - \mathbb{M}_{\hat{\theta}}(\hat{D}_{\hat{\theta}})] + \sup_{\theta\in\Theta} [\mathbb{M}_\theta(\hat{D}_\theta) - \mathbb{M}_\theta(D_\theta)].$$

The first difference is less than $o_P^*(1)$ and the latter two are $o_P^*(1)$ by assumption. Therefore, $\mathbb{M}_{\hat{\theta}}(D_{\hat{\theta}}) \leq \inf_{\theta\in\Theta} \mathbb{M}^\theta(D_\theta) + o_P^*(1)$. Let $\mathcal{M}_1 := \{\log D_\theta : \theta \in \Theta\}$ and $\mathcal{M}_2 := \{\log(1 - D_\theta) \circ T_\theta : \theta \in \Theta\}$. By the assumption of Glivenko–Cantelli, $\|\mathbb{P}_0 - P_0\|_{\mathcal{M}_1} \to 0$ and $\|\tilde{\mathbb{P}}_0 - P_Z\|_{\mathcal{M}_2} \to 0$ in outer probability as $n, m \to \infty$. By van der Vaart and Wellner (1996, Corollary 3.2.3 (i)), it follows that $\hat{\theta} \to \theta_0$ in outer probability. ■

Let $\tilde{h}(\theta_1, \theta_2) := [P_Z(\sqrt{(p_0/p_{\theta_1}) \circ T_{\theta_1}} - \sqrt{(p_0/p_{\theta_2}) \circ T_{\theta_2}})^2]^{1/2}$.

Proof of Theorem 2. Assumption 3 implies $\mathbb{M}_{\hat{\theta}}(D_{\hat{\theta}}) \leq \mathbb{M}_{\theta_0}(D_{\theta_0}) + O_P^*(n^{-1})$, so we apply van der Vaart and Wellner (1996, Theorem 3.2.5) to $\mathbb{M}_\theta(D_\theta)$. By Assumption 4, $M_\theta(D_\theta) - M_{\theta_0}(D_{\theta_0}) \gtrsim h(\theta, \theta_0)^2 \wedge c$ for some $c > 0$ globally in $\theta \in \Theta$. By Assumption 1, $\tilde{h}(\theta, \theta_0)^2 = O(h(\theta, \theta_0))$ as $\theta \to \theta_0$.

Next, we show the convergence of the sample objective function. Note that

$$(\mathbb{M}_{\theta_0} - M_{\theta_0})(D_{\theta_0}) - (\mathbb{M}_\theta - M_\theta)(D_\theta) = (\mathbb{P}_0 - P_0) \log \tfrac{D_{\theta_0}}{D_\theta} + (\tilde{\mathbb{P}}_0 - P_Z) \log \tfrac{(1-D_{\theta_0}) \circ T_{\theta_0}}{(1-D_\theta) \circ T_\theta}.$$

By Lemma S.5, $\|\log \frac{D_{\theta_0}}{D_\theta}\|_{P_0,B}^2 \le 4h(\theta,\theta_0)^2$ and $\|\log \frac{(1-D_{\theta_0})\circ T_{\theta_0}}{(1-D_\theta)\circ T_\theta}\|_{P_Z,B}^2 \le 4\tilde{h}(\theta,\theta_0)^2$. For $\delta > 0$, define $\mathcal{M}_\delta^1 := \{\log \frac{D_{\theta_0}}{D_\theta} : h(\theta,\theta_0) \le \delta\}$ and $\mathcal{M}_\delta^2 := \{\log \frac{(1-D_{\theta_0})\circ T_{\theta_0}}{(1-D_\theta)\circ T_\theta} : \tilde{h}(\theta,\theta_0) \le \delta\}$. By van der Vaart and Wellner (1996, Lemma 3.4.3),

$$\mathbb{E}^* \sup_{h(\theta,\theta_0)<\delta} \left|\sqrt{n}(\mathbb{P}_0 - P_0) \log \frac{D_{\theta_0}}{D_\theta}\right| \lesssim J_{[]}(2\delta, \mathcal{M}_\delta^1, \|\cdot\|_{P_0,B}) \left[1 + \frac{J_{[]}(2\delta,\mathcal{M}_\delta^1,\|\cdot\|_{P_0,B})}{4\delta^2\sqrt{n}}\right].$$

Let $[\ell, u]$ be an ε-bracket in $\{p_\theta\}$ with respect to h. Since $u-\ell \ge 0$ and $e^{|x|}-1-|x| \le 2(e^{x/2}-1)^2$ for every $x \ge 0$,

$$\left\|\log \frac{p_0+u}{p_0+p_{\theta_0}} - \log \frac{p_0+\ell}{p_0+p_{\theta_0}}\right\|_{P_0,B}^2 \le 4\int \left(\sqrt{\frac{p_0+u}{p_0+\ell}} - 1\right)^2 p_0 \le 4h(u,\ell)^2 \le 4\varepsilon^2.$$

Thus, $[\log \frac{p_0+\ell}{p_0+p_{\theta_0}}, \log \frac{p_0+u}{p_0+p_{\theta_0}}]$ makes a 2ε-bracket in $\mathcal{M}^1$. Hence, $N_{[]}(2\varepsilon, \mathcal{M}_\delta^1, \|\cdot\|_{P_0,B}) \le N_{[]}(\varepsilon, \mathcal{P}_\delta, h) \lesssim (\delta/\varepsilon)^r$ by Assumption 1. This induces $J_{[]}(2\delta, \mathcal{M}_\delta^1, \|\cdot\|_{P_0,B}) \lesssim \delta$. Ergo, $\mathbb{E}^* \sup_{h(\theta,\theta_0)<\delta} \left|\sqrt{n}(\mathbb{P}_0 - P_0) \log \frac{D_{\theta_0}}{D_\theta}\right| \lesssim \delta + \frac{1}{\sqrt{n}}$. Similarly, $\mathbb{E}^* \sup_{\tilde{h}(\theta,\theta_0)<\delta} |\sqrt{m}(\tilde{\mathbb{P}}_0 - P_Z) \log \frac{1-D_{\theta_0}}{1-D_\theta}| \lesssim \delta + \frac{1}{\sqrt{m}}$. Then, the result follows by van der Vaart and Wellner (1996, Theorem 3.2.5). ■

Proof of Theorem 3. By Theorem 2, $\hat{\theta}$ is consistent and $\sqrt{n}(\hat{\theta} - \theta_0)$ is uniformly tight. Assumption 3 implies $\mathbb{M}_{\hat{\theta}}(D_{\hat{\theta}}) \le \inf_{\theta \in G_n} \mathbb{M}_\theta(D_\theta) + o_P^*(n^{-1})$. Under Assumptions 2 and 5, for every compact $K \subset \Theta$, $\sqrt{\frac{n}{m}} \sup_{h\in K} |\sqrt{m}(\tilde{\mathbb{P}}_0 - P_Z)(\sqrt{n}[\log(1 - D_{\theta_0}) \circ T_{\theta+h/\sqrt{n}} - \log(1-D_{\theta_0}) \circ T_{\theta_0}])| = o_P^*(1 + \frac{n}{m})$ and $\sqrt{\frac{n}{m}} \sup_{h\in K} \|\sqrt{m}[(\mathbb{P}_{\theta_0+h/\sqrt{n}} - P_{\theta_0+h/\sqrt{n}}) - (\mathbb{P}_{\theta_0} - P_{\theta_0})] D_{\theta_0} \dot{\ell}_{\theta_0}\| = o_P^*(1)$. Let $\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0} := \sqrt{n}(\mathbb{P}_0 - P_0)(1-D_{\theta_0})\dot{\ell}_{\theta_0}$. With Assumptions 2 and 5, Lemma S.3 implies that uniformly in $h \in K$ compact,

$$n[\mathbb{M}_{\theta_0+h/\sqrt{n}}(D_{\theta_0+h/\sqrt{n}}) - \mathbb{M}_{\theta_0}(D_{\theta_0})] = -h^\top \mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0} + \frac{h^\top \tilde{I}_{\theta_0} h}{4} + o_P(1).$$

In particular, this holds for both $\hat{h} := \sqrt{n}(\hat{\theta} - \theta_0)$ and $\breve{h} := 2\tilde{I}_{\theta_0}^{-1}\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0}$, so

$$n[\mathbb{M}_{\theta_0+\hat{h}/\sqrt{n}}(D_{\theta_0+\hat{h}/\sqrt{n}}) - \mathbb{M}_{\theta_0}(D_{\theta_0})] = -\hat{h}^\top \mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0} + \tfrac{1}{4}\hat{h}^\top \tilde{I}_{\theta_0}\hat{h} + o_P^*(1),$$
$$n[\mathbb{M}_{\theta_0+\breve{h}/\sqrt{n}}(D_{\theta_0+\breve{h}/\sqrt{n}}) - \mathbb{M}_{\theta_0}(D_{\theta_0})] = -\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0}^\top \tilde{I}_{\theta_0}^{-1}\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0} + o_P(1).$$

Since G_n shrinks slower than $1/\sqrt{n}$, $\theta_0 + \breve{h}/\sqrt{n}$ is eventually contained in G_n. Since $\hat{h}$ minimizes $\mathbb{M}_\theta(D_\theta)$ up to $o_P^*(1/n)$ in G_n, the LHS of the first equation is larger than that of the second up to $o_P^*(1)$. Subtracting the two, we have $\frac{1}{4}(\hat{h} - 2\tilde{I}_{\theta_0}^{-1}\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0})^\top \tilde{I}_{\theta_0}(\hat{h} - 2\tilde{I}_{\theta_0}^{-1}\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0}) + o_P^*(1) \le 0$. Since $\tilde{I}_{\theta_0}$ is positive definite, $\hat{h} - 2\tilde{I}_{\theta_0}^{-1}\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0} = o_P^*(1)$, proving the expression of $\sqrt{n}(\hat{\theta} - \theta_0)$. The asymptotic variance is $4\tilde{I}_{\theta_0}^{-1} \operatorname{Var}(\mathbb{G}_{\theta_0}\dot{\ell}_{\theta_0})\tilde{I}_{\theta_0}^{-1}$. ■

REFERENCES

Altonji, J. G. and L. M. Segal (1996): "Small-Sample Bias in GMM Estimation of Covariance Structures," *Journal of Business & Economic Statistics*, 14, 353–366.

Asimopoulos, D. C., M. Nitsiou, L. Lazaridis, and G. F. Fragulis (2022): "Generative Adversarial Networks: a Systematic Review and Applications," *SHS Web Conferences*, 139, 03012.

Athey, S., G. Imbens, J. Metzger, and E. Munro (2020): "Using Wasserstein Generative Adversarial Networks for the Design of Monte Carlo Simulations," ArXiv:1909.02210.

Cheng, J., Y. Yang, X. Tang, N. Xiong, Y. Zhang, and F. L. and (2020): "Generative Adversarial Networks: A Literature Review," *KSII Transactions on Internet and Information Systems*, 14, 4625–4647.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014): "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 2672–2680.

Imbens, G. W. (2002): "Generalized Method of Moments and Empirical Likelihood," *Journal of Business & Economic Statistics*, 20, 493–506.

Kaji, T., E. Manresa, and G. Pouliot (2022): "An Adversarial Approach to Structural Estimation," ArXiv:2007.06169v2, February.

Kingma, D. P. and J. Ba (2015): "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Y. Bengio and Y. LeCun.

Klein, R. W. and R. H. Spady (1993): "An Efficient Semiparametric Estimator for Binary Response Models," *Econometrica*, 61, 387–421.

McFadden, D. (1989): "A Method of Simulated Moments for Estimation of Discrete Response Models Without Numerical Integration," *Econometrica*, 57, 995–1026.

Nevo, A. (2000): "A Practitioner's Guide to Estimation of Random-Coefficients Logit Models of Demand," *Journal of Economics & Management Strategy*, 9, 513–548.

Pakes, A. and D. Pollard (1989): "Simulation and the Asymptotics of Optimization Estimators," *Econometrica*, 57, 1027–1057.

van der Vaart, A. W. and J. A. Wellner (1996): *Weak Convergence and Empirical Processes: With Applications to Statistics*, New York: Springer.