Generative Semi-Supervised Classification

Tong Wang[*]
The Chinese University of Hong Kong, Hong Kong SAR, China
Shanshan Song[*]
The Chinese University of Hong Kong, Hong Kong SAR, China
Guohao Shen
The Hong Kong Polytechnic University, Hong Kong SAR, China
Yuanyuan Lin[†]
The Chinese University of Hong Kong, Hong Kong SAR, China
and Jian Huang[†]
The Hong Kong Polytechnic University, Hong Kong SAR, China

August 2, 2023

Abstract

We propose a deep generative approach to semi-supervised classification. The proposed approach is based on the idea that learning a conditional class probability function is equivalent to learning a conditional generator function for the conditional class probability. This idea leads to the construction of an objective function for learning a classifier and a conditional class generator that naturally combines the information from both labeled and unlabeled observations. We take advantage of the approximation power of deep neural networks to approximate the classifier and the conditional generator nonparametrically. We establish consistency and convergence properties of the resulting estimators with respect to certain metric under suitable conditions. Moreover, we conduct numerical studies with simulated and real datasets to evaluate the proposed method and illustrate that the proposed method outperforms an existing semi-supervised classification method as well as supervised methods trained with only labeled samples, especially when the size of the labeled sample is small.

Keywords: Conditional generator, deep neural networks, error analysis, generative learning, semi-supervised learning.

[*] Tong Wang and Shanshan Song are co-first authors.

[†] Co-corresponding authors: Yuanyuan Lin (email: ylin@sta.cuhk.edu.hk) and Jian Huang (email: j.huang@polyu.edu.hk)

1 Introduction

With the rapid development of modern technologies, large amounts of data are increasingly collected. However, for various types of data, such as images, texts, voices and genomic data, most of the samples are unlabeled, and only a small number of them have the label information, due to the difficulty and high cost to label a large amount of data. Semi-supervised learning methods have been developed to analyze such type of datasets. The main question for semi-supervised learning is how to take advantage of unlabeled data to improve prediction and estimation.

There is an extensive literature on semi-supervised classification. Several nonparametric semi-supervised classification methods have been proposed. Examples include self-training (Yarowsky, 1995; Rosenberg et al., 2005), co-training (Blum and Mitchell, 1998), and entropy minimization Grandvalet and Bengio (2004). With self-training, a classifier is trained iteratively by using its own prediction on the unlabeled data and incorporating part of the pseudo-labeled data into the training data. The co-training technique trains two separated classifiers with the labeled data based on two sub-feature sets, and each classifier "teaches" the other one with a few pseudo-labeled observations selected from its prediction on the unlabeled data. Entropy minimization incorporates unlabelled data in the standard supervised learning framework. With additional information on the similarity of labelled and unlabelled examples, some graph-based approaches were developed by Zhou et al. (2004) and Zhu and Lafferty (2005).

Recently, several papers have studied the problems of estimating certain target parameters and prediction rules in semi-supervised learning. Chakrabortty and Cai (2018) studied a class of efficient and adaptive estimator in semi-supervised setting for efficiency improvement. Cheng et al. (2021) proposed a robust and efficient semi-supervised estimator for estimating the average treatment effects in analyzing the electronic health records data. Gronsbell and Cai (2018) proposed semi-supervised approaches for efficient evaluation of model prediction performance. In addition, semi-supervised inference under high-dimensional settings have been considered by Cai and Guo (2020), Deng et al. (2020) and Chakrabortty et al. (2022). These works make certain parametric assumptions on the data distribution, such as the marginal distribution of the predictor or the conditional distribution of the response given the predictor.

In classification problems, to learn a good classifier, an essential step is to learn the conditional distribution of a response vector Y given a predictor vector X (denoted by $P_{Y|X}$), as the conditional distribution gives a full description of the relationship between Y and X. Traditional nonparametric conditional density estimation methods, including nearest neighbors method (Bhattacharya and Gangopadhyay, 1990), smoothing methods (Fan et al., 1996; Bott and Kohler, 2017), regression-based method (Izbicki and Lee, 2017), works well under low-dimensional settings, but generally cannot work well or even fail for high-dimensional data. The generative adversarial networks (GANs) proposed by Goodfellow et al. (2014) have demonstrated excellent ability in learning complex distributions for high-dimensional data. Inspired by the idea of GANs, Zhou et al. (2022) proposed a deep generative approach to sample from a conditional distribution $P_{Y|X}$ based on a unified formulation of conditional distribution and the noise-outsourcing lemma.

In this paper, we propose a novel deep generative semi-supervised method for classification. Our main idea is to introduce a generator function for the conditional class probability so that the unlabeled data can be effectively utilized in learning a classifier. The conditional class probability function and the conditional generator function are estimated nonparametrically based on neural network approximation. We also study the theoretical properties of the resulting estimators. For simplicity, we refer to our proposed method as the generative semi-supervised classification (GSSC).

Our contributions are as follows:

1. We propose a new semi-supervised classification method by introducing a conditional class probability generator for incorporating unlabeled observations. Both the estimated conditional class probabilities and the estimated conditional generator can be used for classification tasks. Our method allows the predictor X to be high-dimensional and can accommodate continuous and discrete predictors.

2. We prove the consistency of the estimated conditional generator in the sense that, the distribution of the conditional generator converges to the conditional distribution of Y given X. We also establish the convergence rate of the generator function in terms of the total variation norm under some mild conditions.

3. Our numerical experiments demonstrate that, the new method exhibits better prediction accuracy than the supervised counterpart (uses the labeled data only) and other semi-supervised methods, such as the entropy minimization method (Grandvalet and Bengio, 2004), especially when the size of the labeled data is relatively small and the size of the unlabeled data is large.

The rest of the paper is organized as follows. In Section 2, we describe the semi-supervised data structure and introduce the proposed GSSC method. We present the implementation details of the proposed method in Section 3. Theoretical results are presented in Section 4. In Section 5 we conduct several numerical experiments using simulated data examples and benchmark data sets to evaluate the performance of our proposed method. Concluding remarks are given in Section 6. Technical proofs and additional numerical results are deferred to Appendix.

2 Generative semi-supervised classification

Let $X \in \mathcal{X}$ be a d-vector of predictors and $Y \in \mathcal{Y}$ be a categorical response vector with K categories ($2 \leq K < \infty$), where $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y}$ is a finite set of class labels. Suppose that the marginal distribution of X is P_X, and the marginal distribution of Y is P_Y, and the joint distribution of (X, Y) is $P_{X,Y}$. Let $e_k = (0, \ldots, 0, 1, 0, \ldots, 0)^\top \in \mathbb{R}^K$ be a unit vector with the kth element 1, $1 \leq k \leq K$. A vector like e_k is also named as one-hot vector. From now on, we use the one-hot encoding and write $\mathcal{Y} = \{e_1, \ldots, e_K\}$. When the label of X is observed, its corresponding response vector Y taking value e_k means that X belongs to the k-th category. A typical semi-supervised data consists of two sources of data: (i) A *labeled* data set $\mathcal{L} = \{(X_i, Y_i) : i = 1, 2, \ldots, n\}$, where $\{(X_i, Y_i)\}_{i=1}^n$ are n independent and identically distributed (i.i.d.) observations sampled from $P_{X,Y}$; (ii) An *unlabeled* data set $\mathcal{U} = \{X_i : i = n+1, n+2, \ldots, n+N, N \geq 1\}$, where $\{X_i\}_{i=n+1}^{n+N}$ are N i.i.d. observations sampled from P_X.

In semi-supervised classification problems, N is usually large but n is relatively small. How to effectively use unlabeled data is the central concern. Below we describe the proposed GSSC method for dealing with this question, by carefully constructing objective functions that can effectively combine information from labeled and unlabeled data. Then the classification functions can be computed based on the proposed objective functions.

2.1 Conditional class probability and its generator

In this subsection, we give a brief description of using a conditional generator to characterize a conditional class probability function. This is the basis of our proposed GSSC method.

First, a key quantity in classification is the conditional class probability function $P(Y = y|X = x)$. It is computationally convenient to represent the conditional class probability using the softmax operator, that is,

$$\mathbb{P}(Y = y|X = x) = y^\top \text{Softmax}(H(x)),\ x \in \mathcal{X}, y \in \mathcal{Y} = \{e_1, \ldots, e_K\}, \tag{1}$$

where $H(x) = (h_1(x), \ldots, h_K(x))^\top$ and $\text{Softmax}(\cdot)$ is the softmax function defined as $\text{Softmax}(a) = (\exp(a_1), \ldots, \exp(a_K))^\top / \{\sum_{k=1}^K \exp(a_k)\}$ for $a = (a_1, \ldots, a_K)^\top \in \mathbb{R}^K$. For notational simplicity, denote $S(x) = \text{Softmax}(H(x))$, so the kth component $S_k(x)$ of $S(x)$ is

$$S_k(x) = \frac{\exp(h_k(x))}{\sum_{k=1}^K \exp(h_k(x))},\ k = 1, \ldots, K. \tag{2}$$

Then, $\mathbb{P}(Y = y|X = x) = y^\top S(x),\ y \in \mathcal{Y}$. In other words, we are simply expressing the conditional class probabilities as $S(x) = (S_1(x), \ldots, S_K(x))$, where

$$S_k(x) = \mathbb{P}(Y = y_k|X = x), x \in \mathcal{X},\ k = 1, \ldots, K. \tag{3}$$

Then the task of classification is to estimate these conditional class probabilities.

We can also estimate the conditional class probability based on the notion of conditional generators. Let η be an m-dimensional random vector independent of X with a known distribution P_η. For instance, one may take the distribution of η to be the standard uniform distribution on $[0, 1]^m$ or the standard multivariate normal distribution $N(\mathbf{0}, \boldsymbol{I}_m)$. Note that m is allowed to be different from K, the number of categories of Y. We also denote the conditional distribution of Y given $X = x$ as $\mathbb{P}_{Y|X=x}$. The noise-outsourcing lemma in probability theory (Kallenberg, 2002) guarantees that under minimal conditions, there exists a function $G^\star : \mathbb{R}^m \times \mathcal{X} \mapsto \mathcal{Y}$ such that the conditional distribution of $G^\star(\eta, X)$ given $X = x$ is the same as $\mathbb{P}_{Y|X=x}$. It then follows from the independence assumption of η and X that, for any $x \in \mathcal{X}$, we have

$$G(\eta, x) \sim P_{Y|X=x}, \ x \in \mathcal{X}. \tag{4}$$

Then, one may sample from the conditional distribution of Y given $X = x$ as follows:

(i) first sample an $\eta \sim P_\eta$;

(ii) then calculate $G^\star(\eta, x)$.

The value $G^\star(\eta, x)$ can be regarded as a sample from $P_{Y|X=x}$. Such a $G^\star$ is referred to as a conditional generator. Again, since η and X are independent, finding a conditional generator $G^\star$ is equivalent to finding a $G^\star$ such that the joint distribution of $(X, G^\star(\eta, X))$, denoted by $P_{X,G^\star(\eta,X)}$, is the same as the joint distribution of (X, Y). In other words, we can estimate $G^\star$ by matching the distribution $P_{X,G^\star(\eta,X)}$ with $P_{X,Y}$. A detailed description of conditional generative learning is given in Zhou et al. (2022).

In the present problem, we can identify $S(x)$ with $P_{Y|X=x}$. To be specific, let $S^*(x) = P_{Y|X=x}$ be the underlying conditional class probability function. Since $G^*(\eta, x) \sim P_{Y|X=x}$ and $Y \in \mathcal{Y}$ is a one-hot vector, we have

$$\mathbb{E}_\eta G^*(\eta, x) = \mathbb{E}(Y|X = x) = S^*(x). \tag{5}$$

Therefore, if we can estimate G^*, an estimate of S^* can be obtained easily based on an empirical average using a random sample from the reference distribution. The relationship

(5) provides a way to incorporate unlabeled data into the estimation of a classifier in a semi-supervised setting.

2.2 Population objective function

We first describe the population objective functions for labeled and unlabeled data. This naturally leads to an empirical objective function for GSSC when a random sample is available,

2.2.1 Cross-entropy loss for labeled data

First, when the label is available, we observe the pair (X, Y). The negative log-likelihood function (or the cross-entropy loss) is $-y^\top \log S(x)$, where the logarithm operates on S componentwise, i.e., $\log S = (\log S_1, \dots, \log S_k)^\top$ for $S = (S_1, \dots, S_K)^\top$. So at the population level, the negative log-likelihood criterion for (X, Y) is

$$L_{\mathrm{CE}}(S) = -\mathbb{E}[Y^\top \log S(X)]. \tag{6}$$

This is also often referred to as the cross entropy classification objective function.

2.2.2 Least squares loss for unlabeled data

For unlabeled data, we only observe X, but Y is not observed. The main question in semi-supervised learning is how to incorporate unlabeled data X in estimating the classifier S. Note that (6) is minimized when $S^\star(x) := \mathbb{E}(Y|X = x)$. Without observing Y, our key idea is to approximate $\mathbb{E}(Y|X = x)$ through a conditional generator function described above. We introduce an m-dimensional random vector $\eta \sim P_\eta$, where it is easy to sample from P_η, and a generator function $G : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ satisfying (5). As described earlier in (5),

$S^\star(x)$ and $\mathbb{E}_{\eta\sim P_\eta}G(\eta, x)$ provide the same information about $\mathbb{E}(Y|X = x)$ from different angles. This motivates us to consider the following least squares loss for matching $S^\star(\cdot)$ and $\mathbb{E}_{\eta\sim P_\eta}G(\eta, \cdot)$ with the unlabeled data

$$L_X(G, S) = \mathbb{E}\|\mathbb{E}_\eta G(\eta, X) - S(X)\|_2^2, \tag{7}$$

where $\|\cdot\|_2$ is the Euclidean norm. Note that $L_X(G, S)$ is minimized when $\mathbb{E}_\eta G(\eta, x) = S(x)$ for almost all x with respect to P_X.

2.2.3 Kullback-Liebler generative loss for labeled data

Recall that S is simply a softmax reexpression of the conditional class probabilities $P(Y = y|X = x)$ as given in (3) and the generator function G is a sampler from $P(Y = y|X = x)$ as given in (4). Therefore, G and S provide the same information about $P(Y = y|X = x)$ from different angles. With labeled data, S can be estimated based on an empirical version of (6). Similarly, we can also use labeled data for supervising the estimation of G.

We estimate G using the conditional generative learning approach (Zhou et al., 2022), which generalizes the generative adversarial networks (GANs) (Goodfellow et al., 2014) to the conditional learning setting. The starting point of our approach is to change the problem of learning a conditional distribution to that of learning an unconditional joint distribution, based on the observation that

$$G(\eta, x) \sim P_{Y|X=x} \text{ if and only if } (X, G(\eta, X)) \sim (X, Y).$$

Therefore, we can learn the conditional generator G by matching the joint distributions of $(X, G(\eta, X))$ and (X, Y). We use the Kullback-Liebler divergence $\mathbb{D}_{\mathrm{KL}}(P_{X,G}\|P_{X,Y})$ to measure the difference between $P_{X,G}$ and $P_{X,Y}$. The variational representation of the KL-divergence is

$$\mathbb{D}_{\mathrm{KL}}(P_{X,G} \| P_{X,Y}) = \sup_{D} L_{\mathrm{KL}}(D,G) + 1,$$

where

$$L_{\mathrm{KL}}(D,G) = \mathbb{E}_{(X,\eta)\sim P_X P_\eta}[D(X, G(\eta, X))] - \mathbb{E}_{(X,Y)\sim P_{X,Y}}[\exp(D(X,Y))]. \tag{8}$$

Based on the basic property of the KL divergence, a G^* satisfies $P_{X,G^*} = P_{X,Y}$ if and only if

$$G^* \in \arg\min_{G} \mathbb{D}_{\mathrm{KL}}(P_{X,G} \| P_{X,Y}) = \arg\min_{G} \max_{D} L_{\mathrm{KL}}(D,G).$$

So we can use the KL divergence as a loss function for learning G. Its variational form is computationally more convenient, since the population expectations can be directly estimated by empirical averages when a random sample is available.

2.2.4 The overall objective function

To combine information from labeled and unlabeled data, we propose an overall objective function for GSSC

$$L_{\mathrm{ALL}}(G,S,D) = \lambda_\ell \{L_{\mathrm{CE}}(S) + L_{\mathrm{KL}}(D,G)\} + \lambda_u L_X(G,S),$$

where $\lambda_l, \lambda_u \geq 0$ are given weights for the contributions from labeled and unlabeled data, respectively. The identifiability of (G,S) is discussed in Section 4.1.

2.3 Empirical objective function

When labeled data $\mathcal{L} = \{(X_i, Y_i) : i = 1, 2, \ldots, n\}$ and unlabeled data $\mathcal{U} = \{X_i, i = n+1, \ldots, n+N\}$ are available, we can construct the empirical versions of L_{CE}, L_{KL} and L_X given in Subsection 2.2 as follows:

$$\begin{aligned}
\widehat{L}_{\mathrm{CE}}(S) &= -\frac{1}{n}\sum_{i=1}^{n} Y_i^{\top} \log S(X_i), \\
\widehat{L}_{\mathrm{KL}}(G, D) &= \frac{1}{n}\sum_{i=1}^{n}[D\{X_i, G(\eta_i, X_i)\} - \exp\{D(X_i, Y_i)\}], \\
\widehat{L}_X(G, S) &= \frac{1}{N}\sum_{i=n+1}^{n+N} \Big\| \frac{1}{N}\sum_{j=n+1}^{n+N} G(\eta_j, X_i) - S(X_i)\Big\|_2^2, \qquad (9)
\end{aligned}$$

where $\{\eta_i, i = 1, \ldots, n+N\}$ is a random sample from the known reference distribution P_η. An overall empirical objective function for GSSC is

$$\widehat{L}_{\mathrm{All}}(G, S, D) = \lambda_\ell\big\{\widehat{L}_{\mathrm{CE}}(S) + \widehat{L}_{\mathrm{KL}}(G, D)\big\} + \lambda_u \widehat{L}_X(G, S).$$

Computationally, we parameterize S, G and D by three neural network functions $S_{\boldsymbol{\omega}}$, $G_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\phi}}$ with parameters (weights and biases) $\boldsymbol{\omega}$, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. We then estimate these parameters by

$$\{\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}\} = \arg\min_{\boldsymbol{\omega}, \boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \widehat{L}_{\mathrm{All}}(G_{\boldsymbol{\theta}}, S_{\boldsymbol{\omega}}, D_{\boldsymbol{\phi}}).$$

We denote the resulting estimators of (S, G, D) by $\hat{S} = S_{\hat{\boldsymbol{\omega}}}$, $\hat{G} = G_{\hat{\boldsymbol{\theta}}}$, and $\hat{D} = D_{\hat{\boldsymbol{\phi}}}$, respectively.

Remark 1. *For computational simplicity, one may also consider approximating* $L_X(G, S)$ *by*

$$\frac{1}{N}\sum_{i=n+1}^{n+N} \Big\| \frac{1}{|J|}\sum_{j\in J} G(\eta_j, X_i) - S(X_i)\Big\|_2^2, \qquad (10)$$

where J is a subset of $\{\eta_i, i = 1, \dots, n+N\}$ with $|J| \ll n + N$, $|J|$ is the cardinality of J. In our numerical experiments, the proposed method works well with $|J| = 1$.

3 Implementation

In this section,we first describe the neural networks used in the approximation of S, G and D, and next present the computational algorithm in detail.

3.1 ReLU Feedforward Neural Networks

We first give a brief description of feedforward neural networks (FNN) with rectified linear unit (ReLU) activation function. Let $\mathcal{F}(s, t, \mathcal{W}, \mathcal{H}, \mathcal{M}, \mathcal{B})$ denote a class of ReLU-activated FNNs $f_{\boldsymbol{\xi}} : \mathbb{R}^s \to \mathbb{R}^t$ with parameter $\boldsymbol{\xi}$, width $\mathcal{W}$, depth $\mathcal{H}$, size $\mathcal{M}$ and $f_{\boldsymbol{\xi}}$ satisfying $\|f_{\boldsymbol{\xi}}\|_\infty \leq \mathcal{B}$ for some $0 < \mathcal{B} < \infty$, where s, t are positive integers and $\|f_{\boldsymbol{\xi}}\|_\infty := \sup_{x \in \mathbb{R}^s} \|f_{\boldsymbol{\xi}}(x)\|_\infty$. Let $\sigma(x) := \max(x, 0)$ denote the ReLU activation function (defined for each element of x if x is a vector).

The multi-layer perceptron (MLP) is an important subclass of feedforward neural networks. The structure of a typical MLP is a composition of a series of functions

$$f_{\boldsymbol{\xi}}(x) = \mathcal{L}_{\mathcal{H}} \circ \sigma \circ \mathcal{L}_{\mathcal{H}-1} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0(x), \quad x \in \mathbb{R}^{p_0},$$

where for $y = (y_1, \dots, y_{p_i})^\top$, $\mathcal{L}_i(y) = W_i y + b_i$, $W_i \in \mathbb{R}^{p_{i+1} \times p_i}$ is a weight matrix, $b_i \in \mathbb{R}^{p_{i+1}}$ is the bias vector in the i-th linear transformation and p_i is the width of the i-th layer for $i = 0, 1, \dots, \mathcal{H}$. Especially, $p_0 = s$ corresponds to the dimension of network input, and $p_{\mathcal{H}+1} = t$ corresponds to the dimension of network output. Thus, $\boldsymbol{\xi} = \{(W_i, b_i)\}_{i=0}^{\mathcal{H}}$. The depth $\mathcal{H}$ refers to the number of hidden layers; the width $\mathcal{W}$ is the maximum width of the hidden layers, i.e., $\mathcal{W} = \max\{p_1, \dots, p_{\mathcal{H}}\}$; the size $\mathcal{M} = \sum_{i=0}^{\mathcal{H}}\{(p_i + 1) \times p_{i+1}\}$ refers to the total number of parameters in the network. An MLP has fully-connected consecutive layers and no connections between non-adjacent layers. Thus, for an MLP class $\mathcal{F}(s, t, \mathcal{W}, \mathcal{H}, \mathcal{M}, \mathcal{B})$, its size satisfies

$$\mathcal{M} \leq \mathcal{W}(s+1) + (\mathcal{W}^2 + \mathcal{W})(\mathcal{H} - 1) + (\mathcal{W} + 1)t = O(\mathcal{W}^2\mathcal{H}).$$

Note that the network parameters $\mathcal{W}, \mathcal{H}, \mathcal{M}, \mathcal{B}$ may depend on n or N, the size of the labeled data and unlabeled data, but the dependence is omitted for notational simplicity. For computational efficiency, a general feedforward neural network may not be fully connected between consecutive layers but still has no connections between non-adjacent layers, then the size of such a network is reduced.

We use three FNNs $\mathcal{S}, \mathcal{G}$ and $\mathcal{D}$ with parameters specified below to approximate the functions S, G and D. For ease of exposition, we use subscripts to denote the network parameters of the function class $\mathcal{S}, \mathcal{G}, \mathcal{D}$.

- For $\mathcal{S}$: Define

$$\begin{aligned} \mathcal{S} \equiv \{g : \mathcal{X} \to \mathcal{S}_u^K, g = \mathrm{Softmax}(f_1, \dots, f_K),\ \min(g(v)) \geq c_0 \,\text{for all}\, v \in \mathcal{X}, \\ f_k \in \mathcal{F}(d, 1, \mathcal{W}_{\mathcal{S}}, \mathcal{H}_{\mathcal{S}}, \mathcal{M}_{\mathcal{S}}, \mathcal{B}_{\mathcal{S}}), k = 1, \dots, K\}, \end{aligned} \tag{11}$$

  where $\mathcal{S}_u^K = \{v \in \mathbb{R}^K : \sum_{k=1}^K v_k = 1, v_k \geq 0, k = 1, \dots, K\}$, $\min(g(v))$ represents the smallest element of the vector $g(v)$, and $0 < c_0 < 1$ is some small constant. In practice, the constraint $\min(g(v)) \geq c_0$ in $\mathcal{S}$ can be satisfied by doing truncation for $f_k, k = 1, \dots, K$. The function class $\mathcal{S}$ is a ReLU-activated FNNs with a softmax output layer, which is applied to ensure the network output is a valid probability vector. For any function $S_{\boldsymbol{\omega}}$ with parameter $\boldsymbol{\omega}$ in $\mathcal{S}$, its width, depth and bound are no longer $(\mathcal{W}_{\mathcal{S}}, \mathcal{H}_{\mathcal{S}}, \mathcal{B}_{\mathcal{S}})$, but smaller than $(K\mathcal{W}_{\mathcal{S}}, \mathcal{H}_{\mathcal{S}} + 1, 1)$ due to the construction of $\mathcal{S}$.

- For $\mathcal{G}$: Let $\mathcal{G} \equiv \mathcal{F}(m + d, K, \mathcal{W}_{\mathcal{G}}, \mathcal{H}_{\mathcal{G}}, \mathcal{M}_{\mathcal{G}}, \mathcal{B}_{\mathcal{G}})$ be a ReLU-activated FNNs consisting of functions $G_{\boldsymbol{\theta}} : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ with parameter $\boldsymbol{\theta}$, width $\mathcal{W}_{\mathcal{G}}$, depth $\mathcal{H}_{\mathcal{G}}$, size $\mathcal{M}_{\mathcal{G}}$ and bound $\mathcal{B}_{\mathcal{G}} \leq 1$.

- For $\mathcal{D}$: Let $\mathcal{D} \equiv \mathcal{F}(K + d, 1, \mathcal{W}_{\mathcal{D}}, \mathcal{H}_{\mathcal{D}}, \mathcal{M}_{\mathcal{D}}, \mathcal{B}_{\mathcal{D}})$ be a ReLU-activated FNNs consisting of functions $D_{\boldsymbol{\phi}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ with parameter $\boldsymbol{\phi}$, width $\mathcal{W}_{\mathcal{D}}$, depth $\mathcal{H}_{\mathcal{D}}$, size $\mathcal{M}_{\mathcal{D}}$ and bound $\mathcal{B}_{\mathcal{D}}$.

3.2 Computation

The training algorithm for GSSC is presented in Algorithm 1.

We set $\lambda_l = \lambda_u = 1$ in the numerical experiments presented in Section 5.

4 Theoretical results

In this section, we first show that the conditional class probability and the generator function can be identified as the minimizer of the population objective function. We then establish the convergence properties of GSSC.

```
Algorithm 1 Training GSSC
  Require: (a) Labeled data $\{(Y_i, X_i)\}_{i=1}^{n}$; (b) Unlabeled data $\{X_i\}_{i=n+1}^{n+N}$; (c) Minibatch size $v \leq \min(n, N)$.
  for number of training iterations do
      Sample $v$ pairs $\{(Y_{bj}, X_{bj})\}_{j=1}^{v}$ from $\{(Y_i, X_i)\}_{i=1}^{n}$.
      Sample $v$ values $\{\tilde{X}_{bj}\}_{j=1}^{v}$ from $\{X_i\}_{i=n+1}^{n+N}$.
      Generate $v$ noises $\{\eta_j\}_{j=1}^{v}$ from $N(\mathbf{0}, \boldsymbol{I}_m)$.
      Update $D_{\boldsymbol{\phi}}$ by ascending its stochastic gradient:
          $\nabla_{\boldsymbol{\phi}} \frac{1}{v} \sum_{j=1}^{v} [D_{\boldsymbol{\phi}}(X_{bj}, G_{\boldsymbol{\theta}}(\eta_j, X_{bj})) - \exp(D_{\boldsymbol{\phi}}(X_{bj}, Y_{bj}))]$.
      Update $G_{\boldsymbol{\theta}}$ by descending its stochastic gradient:
          $\nabla_{\boldsymbol{\theta}} \frac{1}{v} \sum_{j=1}^{v} [\lambda_l D_{\boldsymbol{\phi}}(X_{bj}, G_{\boldsymbol{\theta}}(\eta_j, X_{bj})) + \lambda_u \| G_{\boldsymbol{\theta}}(\eta_j, \tilde{X}_{bj}) - S_{\boldsymbol{\omega}}(\tilde{X}_{bj}) \|_2^2]$.
      Update $S_{\boldsymbol{\omega}}$ by ascending its stochastic gradient:
          $\nabla_{\boldsymbol{\omega}} \frac{1}{v} \sum_{j=1}^{v} \left[ \lambda_l Y_{bj}^{\top} \log S_{\boldsymbol{\omega}}(X_{bj}) - \lambda_u \| G_{\boldsymbol{\theta}}(\eta_j, \tilde{X}_{bj}) - S_{\boldsymbol{\omega}}(\tilde{X}_{bj}) \|_2^2 \right]$.
  end for
```

4.1 Identifiability

We first consider the existence and consistency of the estimated class probability $\hat{S}$ and the estimated conditional generator $\hat{G}$. Define

$$(G^{\star}, S^{\star}, D^{\star}) = \arg \min_{G,S} \max_{D} L_{\mathrm{ALL}}(G, S, D). \tag{12}$$

Lemma 1. *Let (X, Y) be a random pair taking values in $\mathcal{X} \times \mathcal{Y}$ with joint distribution $P_{X,Y}$. Then, there exists a random vector $\eta \sim$ Uniform$[0,1]^m$ for any given $m \geq 1$ such that the Borel-measurable function $G^{\star}$ defined in (12) always exists and satisfies $(X, G^{\star}(\eta, X)) \sim P_{X,Y}$. Moreover, for the S^{*} given in (12), we have $S^{\star}(x) = \mathbb{E}(Y|X = x)$.*

Lemma 1 shows the existence and identifiability of $G^{\star}$ and $S^{\star}$. It imples that minimizing the population-level loss function in (12) yields a function $G^{\star} : [0,1]^m \times \mathcal{X} \to \mathcal{Y}$ satisfying that, the conditional distribution of $G^\star(\eta, X)$ given $X = x$ is the same as the distribution of Y given $X = x$ for almost all $x \in \mathcal{X}$. This leads to $\mathbb{E}\{G^\star(\eta, X)|X\} = \mathbb{E}(Y|X) = S^\star(X)$.

4.2 Non-asymptotic error bounds

Without loss of generality, we assume that $\mathcal{E} \times \mathcal{X} \subset [0, 1]^{m+d}$, where $\mathcal{E}$ is the domain of the random noise η, i.e., $\eta \in \mathcal{E} \subset \mathbb{R}^m$ for $m \geq 1$. For $x \in \mathbb{R}^K$, define $\|x\|_1 = \sum_{j=1}^K |x_j|$, $\|x\|_2 = (\sum_{j=1}^K |x_j|^2)^{1/2}$. Let $\mathbb{N}_0$ be the set of natural numbers and $\lfloor\beta\rfloor$ be the largest integer strictly smaller than β. For any $\beta > 0$ and $\mathcal{Z} \subset \mathbb{R}^s$, the ball of β-Hölder functions with radius $0 < B < \infty$ is defined as

$$C^\beta(\mathcal{Z}, B) = \left\{ f : \mathcal{Z} \to \mathbb{R}, \max_{\|\alpha\|_1 \leq \lfloor\beta\rfloor} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \leq B, \max_{\|\boldsymbol{\alpha}\|_1 = \lfloor\beta\rfloor} \sup_{x,y \in \mathcal{Z}, x \neq y} \frac{|\partial^{\boldsymbol{\alpha}} f(x) - \partial^{\boldsymbol{\alpha}} f(y)|}{\|x - y\|_2^{(\beta - \lfloor\beta\rfloor)}} \leq B \right\},$$

where $\partial^\alpha = \partial^{\alpha_1} \cdots \partial^{\alpha_s}$ with $\alpha = (\alpha_1, \ldots, \alpha_s)^\top \in \mathbb{N}_0^s$ and s is a positive integer. The function class $\mathcal{P}(\mathcal{Z}, \beta, B)$ of β-smooth conditional class probabilities is defined as

$$\mathcal{P}(\mathcal{Z}, \beta, B) = \left\{ p = (p_1, \ldots, p_K)^\top : \mathcal{Z} \to \mathcal{S}_u^K, \quad p_k \in C^\beta(\mathcal{Z}, B), k = 1, \ldots, K \right\},$$

where $\mathcal{S}_u^K = \{(v_1, \ldots, v_K)^\top : v_k \geq 0, \sum_{k=1}^K v_k = 1\}$.

Next, we shall establish the consistency of $\hat{G}$. Define $p_{X,Y}(x, y) = p_{X|Y=y}(x)\mathbb{P}(Y = y)$, where $p_{X|Y=y}(x)$ denotes the conditional density function of X given $Y = y$. For a fixed G, $p_{X,G(\eta,X)}(x, y)$ is defined in a similar way. The following conditions are imposed.

(C1) There exists a small constant $0 < c_1 < 1$ such that $\min(S^\star(x)) \geq c_1$ for all $x \in \mathcal{X}$, where $S^\star : \mathcal{X} \to \mathcal{S}_u^K$ is the true conditional class probability and $\min(S^\star(x))$ is the smallest element of $S^\star(x)$.

(C2) The true conditional class probability $S^\star \in \mathcal{P}(\mathcal{X}, \beta_S, B_S)$ for $0 < B_S < \infty$.

(C3) (i) Any underlying generator $G^\star : \mathcal{E}\times\mathcal{X} \to \mathcal{Y}$ satisfies that $G^\star(\eta, x) \equiv \text{Onehot}(h^\star(\eta, x))$, where $h^\star(\cdot) = (h_1^\star(\cdot), \ldots, h_K^\star(\cdot))^\top$ is a K-dimensional vector function with $\{h_k^\star(\cdot)\}_{k=1}^K \subset C^{\beta_h}(\mathcal{E} \times \mathcal{X}, B_h)$ and $0 < B_h < \infty$. Here, $\text{Onehot} : \mathbb{R}^K \to \mathcal{Y}$ is a one-hot encoding function, mapping any vector $a = (a_1, \ldots, a_K)^\top \in \mathbb{R}^K$ into a unit vector $e_{k^\star}$ with the $k^\star$-th element 1, where $k^\star$ is the order of the maximum element of a. (ii) There exist two constants $0 < c_2 < \infty$ and $r \in (0, \infty]$ such that for any $t > 0$,

$$\mathbb{P}\left[\max\nolimits_{(1)}\{h^\star(\eta, X)\} - \max\nolimits_{(2)}\{h^\star(\eta, X)\} \leq t\right] \leq c_2 t^r,$$

where $\max_{(1)}\{h^\star(\eta, X)\}$ and $\max_{(2)}\{h^\star(\eta, X)\}$ denote the maximal and the second maximal element of $h^\star(\eta, X)$, respectively.

(C4) There exist constants $\beta_D > 0$ and $0 < B_D < \infty$ such that for any $G \in \mathcal{G}$ and $y_0 \in \mathcal{Y}$, the function $D_G(x, y_0) \in C^{\beta_D}(\mathcal{X}, B_D)$, i.e., $D_G(x, y_0) = \log\{p_{X,G(\eta,X)}(x, y_0)/p_{X,Y}(x, y_0)\}$ is a β_D-hölder smooth function defined on $\mathcal{X}$ with constant B_D.

Condition (C1) implies that the conditional class probability of each category is bounded away from 0, which is a regular condition to ensure the feasibility of finding a vector of estimated conditional class probabilities in the network class $\mathcal{S}$ defined in (11). Condition (C1) is made for technical convenience and it might be relaxed by using the intriguing small value bound condition discussed in Bos and Schmidt-Hieber (2022). Condition (C2) is a smoothness assumption of $S^\star$. Condition (C3)(i) requires that the true generator $G^\star$ is a composition of K β_h-Hölder smooth functions and the one-hot encoding function. Condition (C3)(ii) is analogous to the Tsybakov's noise condition for binary classification problems (Tsybakov, 2004; Kim et al., 2021; Shen et al., 2021). It is a noise condition for multi-class classification and the parameter r is called the noise exponent. Recall that $\mathcal{Y}$ is the set of K-dimensional one-hot vectors. For all $y_0 \in \mathcal{Y}$ and any $G \in \mathcal{G}$, Condition (C4) is a smoothness condition for $D_G(x, y_0)$ with respect to x.

To lighten the notation, we write $\gamma_D := \beta_D/d$, $\gamma_G := \beta_h/(m+d)$ and $\gamma_S := \beta_S/d$, then γ_D, γ_G and γ_S are smoothness-to-dimension ratios. For convenience, we define two functions needed to specify the required neural network structures:

$$\mathcal{W}(\beta, s, N) := 38(\lfloor \beta \rfloor + 1)^2 3^s s^{\lfloor \beta \rfloor + 1} \lceil N \rceil \lceil \log_2(8N) \rceil,$$

$$\mathcal{H}(\beta, s, M) := 21(\lfloor \beta \rfloor + 1)^2 \lceil M \rceil \lceil \log_2(8M) \rceil + 2s,$$

where $\lceil \beta \rceil$ denotes the smallest integer no less than β, and $\lfloor \beta \rfloor$ denotes the largest integer strictly smaller than β. We then specify the network parameters of $\mathcal{G}$, $\mathcal{D}$ and $\mathcal{S}$ below.

(NG): The class $\mathcal{G} \equiv \mathcal{F}(m+d, K, \mathcal{W}_{\mathcal{G}}, \mathcal{H}_{\mathcal{G}}, \mathcal{M}_{\mathcal{G}}, \mathcal{B}_{\mathcal{G}})$ has parameters: width $\mathcal{W}_{\mathcal{G}} = \mathcal{W}(\beta_h, m+d, n^{1/\{2(2r \cdot \gamma_G + 1)\}}) + 4K$, depth $\mathcal{H}_{\mathcal{G}} = \mathcal{H}(\beta_h, m+d, \log n) + 2\lceil \log_2 K \rceil + 2$, size $\mathcal{M}_{\mathcal{G}} \leq O(\mathcal{W}_{\mathcal{G}}^2 \mathcal{H}_{\mathcal{G}})$ and bound $\mathcal{B}_{\mathcal{G}} = 1$, where the inequality holds in an element-wise sense.

(ND): The class $\mathcal{D} \equiv \mathcal{F}(K+d, 1, \mathcal{W}_{\mathcal{D}}, \mathcal{H}_{\mathcal{D}}, \mathcal{M}_{\mathcal{D}}, \mathcal{B}_{\mathcal{D}})$ has parameters: width $\mathcal{W}_{\mathcal{D}} = 2(2\lceil \sqrt{K} \rceil + 1)K + 2K\mathcal{W}(\beta_D, d, n^{1/\{2(2\gamma_D + 1)\}})$, depth $\mathcal{H}_{\mathcal{D}} = \mathcal{H}(\beta_D, d, \log n) + 2\lceil \beta_D/d \rceil + 5$, size $\mathcal{M}_{\mathcal{D}} \leq O(\mathcal{W}_{\mathcal{D}}^2 \mathcal{H}_{\mathcal{D}})$ and bound $\mathcal{B}_{\mathcal{D}} = B_D$.

(NS): The class $\mathcal{S}$ in (11) has parameters: $\mathcal{W}_{\mathcal{S}} = 2\{3^4(\lceil n^{1/\{2(2\gamma_S+1)\}}/\log n \rceil + 1) + \mathcal{W}(\beta_S, d, n^{1/\{2(2\gamma_S+1)\}}/\log n)\}$, $\mathcal{H}_{\mathcal{S}} = \mathcal{H}(\beta_S, d, \log n) + 12\lceil \log n \rceil + 19 + 2d$, $\mathcal{M}_{\mathcal{S}} \leq O(\mathcal{W}_{\mathcal{S}}^2 \mathcal{H}_{\mathcal{S}})$ and $\mathcal{B}_{\mathcal{S}} = -\log c_0$.

To establish the consistency of $\hat{G}$ in the sense that, for any $x \in \mathcal{X}$ with $p_X(x) > 0$, the distribution of $\hat{G}(\eta, x)$ converges to the conditional distribution of Y given $X = x$, we will first derive a slightly stronger result that the total variation distance between $p_{X,Y}$ and

$p_{X,\hat{G}(\eta,X)}$, defined as

$$\|p_{X,\hat{G}(\eta,X)} - p_{X,Y}\|_{L_1} := \int_{\mathcal{X}} \sum_{y\in\mathcal{Y}} \left| p_{X,\hat{G}(\eta,X)}(x,y) - p_{X,Y}(x,y) \right| dx,$$

converges to 0. In the error bounds stated below, we will use the notation

$$\gamma^* = \max\{\gamma_D^{-1}, (r\cdot\gamma_G)^{-1}, \gamma_S^{-1}\}.$$

Theorem 1. *Suppose that Conditions (C1) - (C4) hold and the network parameters of $\mathcal{G}$, $\mathcal{D}$ and $\mathcal{S}$ satisfying* (**NG**), (**ND**) *and* (**NS**) *respectively. Also, suppose that $\lambda_\ell > 0$ and $\lambda_u > 0$ are fixed. Then,*

$$\begin{aligned}&\mathbb{E}_{\mathcal{L},\mathcal{U},\{\eta_i\}_{i=1}^{n+N}} \left\{ \|p_{X,\hat{G}(\eta,X)} - p_{X,Y}\|_{L_1}^2 \right\} \\ \leq& C_2 \left[n^{-\frac{1}{2+\gamma^*}} (\log n)^{\frac{5}{2}} + N^{-\frac{1}{2}} (\log N)^{\frac{1}{2}} n^{\frac{1}{2(2r\cdot\gamma_G+1)} \vee \frac{1}{2(2\gamma_S+1)}} (\log n)^2 \right],\end{aligned} \tag{13}$$

where $\mathbb{E}_{\mathcal{L},\mathcal{U},\{\eta_i\}_{i=1}^{n+N}}$ is taken with respect to the available data $\mathcal{L}\cup\mathcal{U}$ and $\{\eta_i\}_{i=1}^{n+N}$ sampled independently from P_η, and C_2 is a positive constant independent of n, N.

Theorem 1 gives non-asymptotic upper bound for the expected squared total variation norm between $p_{X,Y}$ and $p_{X,\hat{G}(\eta,X)}$. The proof of Theorem 1, deferred to Appendix, rests on the empirical process theory (Van der Vaart and Wellner, 1996; Anthony and Bartlett, 1999; Bartlett et al., 2019) and the latest theoretical advancements on approximating continuous functions by deep neural networks (Shen et al., 2019; Lu et al., 2021; Jiao et al., 2021). The main challenge in our theoretical analysis is that three neural networks are involved and they are trained based on different loss functions. Note that the convergence rate of $\mathbb{E}_{\mathcal{L},\mathcal{U},\{\eta_i\}_{i=1}^{n+N}}\{\|p_{X,\hat{G}(\eta,X)} - p_{X,Y}\|_{L_1}^2\}$ depends on the noise exponent r. Moreover, when

$n/N \to \rho \in [0, \infty)$ as $\min(n, N) \to \infty$, the upper bound in Theorem 1 reduces to

$$C_2' n^{-\frac{1}{2+\gamma^*}} (\log n)^{\frac{5}{2}},$$

where C_2' is a positive constant depending on ρ and decreases as ρ gets smaller. This suggests that large-sized unlabeled data will help improve the convergence rate of the upper bound for $\mathbb{E}_{\mathcal{L},\mathcal{U},\{\eta_i\}_{i=1}^{n+N}}\{\|p_{X,\hat{G}(\eta,X)} - p_{X,Y}\|_{L_1}^2\}$.

A direct consequence of Theorem 1 is that $(X, \hat{G}(\eta, X))$ converges in distribution to (X, Y). It further implies that for almost all $x \in \mathcal{X}$, the conditional distribution of $\hat{G}(\eta, x)$ given $X = x$ converges to the conditional distribution of Y given $X = x$. We state it in the next corollary.

Corollary 1. *Suppose that those conditions of Theorem 1 hold and as* $\min(n, N) \to \infty$, $N^{-1/2}(\log N)^{1/2}\ n^{\frac{1}{2(2r\cdot\gamma_G+1)} \vee \frac{1}{2(2\gamma_S+1)}} (\log n)^2 \to 0$. *Then,*

$$\mathbb{E}_{X\sim P_X}\left\{\sum_{y\in\mathcal{Y}} |P_{\hat{G}(\eta,X)|X}(y) - P_{Y|X}(y)|\right\} \xrightarrow{P} 0,$$

as $\min(n, N) \to \infty$, *where* $P_{\hat{G}(\eta,X)|X=x}(\cdot)$ *and* $P_{Y|X=x}(\cdot)$ *are the conditional distribution functions of* $\hat{G}(\eta, X)$ *and* Y *given* $X = x$, *respectively.*

Theorem 1 and Corollary 1 provide theoretical support for GSSC under suitable conditions. Based on Theorem 1, we can establish the following convergence property for the vector ofthe estimated class probabilities $\hat{S}$.

Theorem 2. *Suppose that those conditions of Theorem 1 hold. Then,*

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{L},\mathcal{U},\{\eta_i\}_{i=1}^{n+N}}\left[\left\{\mathbb{E}_{X\sim P_X}\|\hat{S}(X)-S^\star(X)\|_1\right\}^2\right] \\
\leq & C_3\left[n^{-\frac{1}{2+\gamma^*}}(\log n)^{\frac{5}{2}}+N^{-\frac{1}{2}}(\log N)^{\frac{1}{2}}n^{\frac{1}{2(2r\cdot\gamma_G+1)}\vee\frac{1}{2(2\gamma_S+1)}}(\log n)^2\right],
\end{aligned} \tag{14}
$$

where C_3 is a positive constant independent of n, N.

The proof of Theorem 2 is deferred to Appendix. Theorem 2 provides the non-asymptotic upper bound for the expected squared L_1 metric of $\hat{S}$ and the true $S^\star$. Theorem2 indicates that a large amount of unlabeled data can help reduce the error upper bound of $\hat{S}$. To see this, when $n/N \to \rho \in [0,\infty)$ as $\min(n,N)\to\infty$, the upper bound of (14) reduces to

$$
C_3' n^{-\frac{1}{2+\gamma^*}}(\log n)^{\frac{5}{2}},
$$

where C_3' is a positive constant depending on ρ.

4.3 Error decomposition

We now provide a high-level description of the proof of the results stated above. First, for notational simplicity, we define

$$
\widetilde{L}_X(G,S)=\frac{1}{N}\sum_{i=n+1}^{n+N}\|\mathbb{E}_{\eta\sim P_\eta}G(\eta,X_i)-S(X_i)\|_2^2.
$$

Let $\mathbb{L}_{\mathrm{KL}}(G)=\sup_D L_{\mathrm{KL}}(D,G)$. For two vectors x and y of the same length, let x/y be the element-wise division of x and y.

Our approach is based on decomposing the error into the following terms that are easier to analyze.

$$\begin{aligned}
\Delta_1 &= \sup_{D} L_{\mathrm{KL}}(D, \hat{G}) - \sup_{D \in \mathcal{D}} L_{\mathrm{KL}}(D, \hat{G}), \\
\Delta_2 &= \sup_{D \in \mathcal{D}, G \in \mathcal{G}} |L_{\mathrm{KL}}(D, G) - \widehat{L}_{\mathrm{KL}}(D, G)|, \\
\Delta_3 &= \sup_{S \in \mathcal{S}} |L_{\mathrm{CE}}(S) - \widehat{L}_{\mathrm{CE}}(S)|, \\
\Delta_4 &= \sup_{G \in \mathcal{G}, S \in \mathcal{S}} |L_{\mathrm{X}}(G, S) - \widetilde{L}_{\mathrm{X}}(G, S)|, \\
\Delta_5 &= \sup_{G \in \mathcal{G}, S \in \mathcal{S}} |\widehat{L}_{\mathrm{X}}(G, S) - \widetilde{L}_{\mathrm{X}}(G, S)|, \\
\Delta_6 &= \inf_{G \in \mathcal{G}, S \in \mathcal{S}} \Big\{ \sup_{D} L_{\mathrm{ALL}}(G, S, D) - \sup_{D} L_{\mathrm{ALL}}(G^\star, S^\star, D) \Big\}.
\end{aligned}$$

According to their definitions, Δ_1 and Δ_6 involve approximation of smooth functions using neural networks, and Δ_2 to Δ_5 involve approximation of population expectations using empirical averages. Therefore, we refer to Δ_1 and Δ_6 as approximation errors, and Δ_2 to Δ_5 as stochastic errors.

Lemma 2. *Suppose that Condition (C1) holds. Then, for the generator estimator $\hat{G}$ defined in Section 2.3 with given positive weights λ_ℓ, λ_u,*

$$\|p_{X,\hat{G}(\eta,X)} - p_{X,Y}\|_{L_1}^2 \leq 2\Delta_1 + 4\Delta_2 + 4\Delta_3 + \frac{4\lambda_u}{\lambda_\ell}\Delta_4 + \frac{4\lambda_u}{\lambda_\ell}\Delta_5 + \frac{2}{\lambda_\ell}\Delta_6.$$

The proof of Theorem 1 is based on controlling each error term on the right side of the inequality in this lemma.

Next, we give an error decomposition for the square of the expected L_1 error of the conditional class probability function.

Lemma 3. *Suppose that Condition (C1) holds. Then, for the generator estimator $\hat{S}$ defined* *in Section 2.3 with given positive weights λ_ℓ, λ_u satisfying $\lambda_\ell + \lambda_u = 1$, we have*

$$\Big[\mathbb{E}_{X \sim P_X}\{\|\hat{S}(X) - S^\star(X)\|_1\}\Big]^2 \leq \Delta_1 + 2\Delta_2 + 2\Delta_3 + \frac{2\lambda_u}{\lambda_\ell}\Delta_4 + \frac{2\lambda_u}{\lambda_\ell}\Delta_5 + \frac{1}{\lambda_\ell}\Delta_6.$$

This error decomposition is used for the proof of Theorem 2.

The stochastic errors can be analyzed using The empirical process theory (Van der Vaart and Wellner, 1996; Anthony and Bartlett, 1999; Bartlett et al., 2019; Bartlett and Mendelson, 2002). The approximation errors can be controlled using the results for the approximation properties of deep neural networks (Shen et al., 2019; Lu et al., 2021; Jiao et al., 2021).

5 Numerical studies

In this section, we conduct numerical studies to assess the performance of GSSC via a simulated dataset and three real datasets.

For comparison, we consider several supervised methods (only labeled data are used in the training), including the classical logistic regression (LR, implemented in the R package glmnet.) (Cox, 1958), the nonparametric logistic regression (NLR, implemented by the neural networks) (Hastie and Tibshirani, 1987), and the conditional generative adversarial networks (cGAN) (Mirza and Osindero, 2014; Zhou et al., 2022). We also include the semi-supervised entropy minimization (SSEM) (Grandvalet and Bengio, 2004) in the comparison using both labeled and unlabeled data. For our proposed GSSC method, we report the classification performance based on the estimated conditional class probabilities (GSSC-S) and the estimated generator (GSSC-G). We implement all methods, except for LR, in Pytorch and adopt the stochastic gradient descent algorithm Adam (Kingma and Adam,

2015) for training the neural networks. Four datasets, including one simulated dataset and three real datasets, are used to illustrate the performance, and the sample sizes of each dataset used for training, validation, and testing are reported in Table 1. For all experiments, the noise random vector η is sampled from the standard multivariate normal distribution.

Table 1: The sample sizes of datasets used in the experiments

	Labeled data	Unlabeled data		Validation	Test
Dataset		S	SS		
Two Moon	15	0	$10,000$	2×10^3	2×10^3
Crop	15	0	42,500	10^3	10^4
MFCCs	50	0	5,077	10^3	10^3
MNIST	$16 \sim 50$	0	$10^2 \sim 2 \times 10^4$	10^3	10^4

Notes: S refers to the supervised methods; SS refers to the semi-supervised methods. The unlabeled data used in the semi-supervised method are randomly sampled from the dataset with labels removed. Two Moon is the synthetic two-moon dataset; MFCCs is the Anuran Calls (MFCCs) dataset; Crop is the optimal-radar dataset of the crop; MNIST is the MNIST handwritten digits dataset. For the MNIST dataset, different sizes of labeled and unlabeled data are tried.

Additional numerical results, including the visualization of the real datasets, and the performance of our proposed GSSC with different dimensions of the noise vector η, are provided in Appendix.

5.1 The two-moon synthetic dataset

We first consider the nonlinear two-moon example. Let $X \in \mathbb{R}^2$ be the predictor vector and let $Y \in \{0, 1\}$ be the class label. The two-moon data generating model is

$$Y = \begin{cases} 0 & \text{if } X = (\cos(\alpha) + \frac{1}{2} + \epsilon_1, \sin(\alpha) - \frac{1}{6} + \epsilon_2), \\ 1 & \text{if } X = (\cos(\alpha) - \frac{1}{2} + \epsilon_3, -\sin(\alpha) + \frac{1}{6} + \epsilon_4), \end{cases}$$

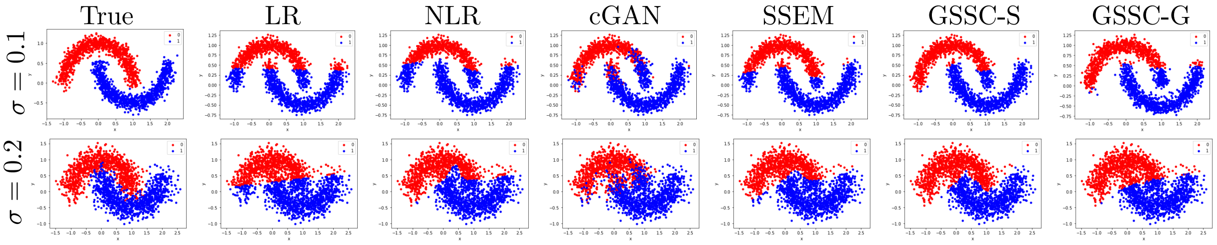
where $\alpha \sim \text{Uniform}[0, \pi]$, $\varepsilon_1, \ldots, \varepsilon_4$ are i.i.d following $N(0, \sigma^2)$ with $\sigma = 0.1$ or 0.2. The 3 FNNs used in this simulation have two fully-connected hidden layers with widths 32 and 16. The noise $\eta \sim N(\mathbf{0}, \boldsymbol{I}_2)$.

Table 2 reports the classification accuracy of different methods on the test dataset. One can see that when the size of the labeled data is small and the size of unlabeled data is large, our proposed method based on the estimated class probabilities has highest classification accuracy. We also provide a figure to display the classification results on the test data in Appendix.

Table 2: Classification accuracy for the two-moon synthetic data.

	Supervised methods			Semi-supervised methods		
σ	LR	NLR	cGAN	SSEM	GSSC-S	GSSC-G
0.1	0.823	0.889	0.855	0.895	0.958	0.900
0.2	0.846	0.887	0.814	0.899	0.916	0.904

Figure 1: Comparison of classification results on the two-moon test data of different methods.



Notes: LR is the parametric logistic regression; NLR is the nonparametric logistic regression; cGAN is the conditional gan; SSEM is the semi-supervised entropy minimization method; GSSC-S and GSSC-G are the proposed semi-supervised classification methods with prediction based on the estimated S and G respectively.

5.2 Crop classification for the Optimal-radar dataset

We apply our method to a real dataset with high-dimensional X. We conduct crop classification for the optical-radar dataset (Khosravi and Alavipanah, 2019), which is available at UCI machine learning repository. The response is *crop type* with 5 categories: Corn, Canola, Soybeans, Oats and Wheat. There are 174 continuous covariates, including 98 radar features and 76 optical features collected on two dates. The three networks used in our method are two-layer fully connected FNNs with 64 nodes. The noise $\eta \sim N(\mathbf{0}, \boldsymbol{I}_{30})$. Table 3 reports the classification accuracy on the test dataset. The classification accuracy of our proposed method is higher than 90% and outperforms other competitors.

Table 3: Classification accuracy for the optimal-radar dataset

Supervised methods			Semi-supervised methods		
LR	NLR	cGAN	SSEM	GSSC-S	GSSC-G
0.446	0.854	0.834	0.843	0.937	0.936

5.3 Anuran Calls (MFCCs) dataset

We apply our method to analyze the anuran calls dataset[1]. The goal is to classify the anuran species through their calls. The dataset contains 22 features, which are Mel-frequency cepstral coefficients (MFCCs), commonly used in speech recognition systems. There are total 7127 syllables (samples) belonging to 3 families: Dentobatidae, Hylidae, Leptodactylidae. We define *Family* as the label $Y \in \{0, 1, 2\}$ and MFCCs as the predictor vector $X \in \mathbb{R}^{22}$. Note that the sizes of the three classes Leptodactylidae, Hylidae and Dentobatidae in the dataset are 62%, 30% and 8% respectively, which is somethat imbalanced.

We randomly select 50 samples as labeled data, 5077 label-removed samples as unlabeled data, 1000 samples as validation data and 1000 samples as test data. Note that the sizes of the three classes Leptodactylidae, Hylidae and Dentobatidae in the dataset are 62%, 30% and 8% respectively, which is somethat imbalanced. For the 50 labeled samples, 30 samples are Leptodactylidae, 17 samples are Hylidae and 3 samples are Dentobatidae. The two networks for approximating S and G have two hidden layers with width (50,20). The discriminator network has two hidden layers with width (50,25). The noise $\eta \sim N(\mathbf{0}, \boldsymbol{I}_3)$.

[1] `https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+$%$28MFCCs$$%$29`

To examine the prediction power, we perform classification on the test data and present the results in Table 4, which shows that our proposed method outperforms the three supervised methods and performs comparably to the semi-supervised entropy minimization method. Especially, when comparing the performance of conditional distribution generator, we notice that cGAN tends to group all samples into one class, while GSSC-G gives more accurate classification.

Table 4: Classification accuracy for the Anuran Calls dataset.

Supervised methods			Semi-supervised methods		
LR	NLR	cGAN	SSEM	GSSC-S	GSSC-G
0.875	0.907	0.623	0.920	0.922	0.916

5.4 MNIST handwritten digits dataset

We also apply our method to an image classification problem with the MNIST handwritten digits dataset[2]. Each image is sized as a 28×28 matrix with gray color intensity from 0 to 1, and paired with a label in $\{0, 1, \ldots, 9\}$. So the dimension of X is $28 \times 28 = 784$ and the dimension of the one-hot Y is 10.

The neural networks used in this experiment are specified as follows: (a) the network $\mathcal{S}$ is a small Alexnet with (32,64,128) filters in its first three convolutional layers and two fully-connected layers with 1,024 and 512 nodes; (b) the discriminator network is a fully-connected network with 2 hidden layers, and widths 256 and 256; (c) the generator network consists of two convolutional layers with 10 and 20 filters to extract the features of the image, and then concatenates with the random noise η by two fully-connected layers with widths 330 and 128. The noise $\eta \sim N(\mathbf{0}, \boldsymbol{I}_{100})$.

Table 5 reports the classification accuracy on the testing data of different methods

[2] `http://yann.lecun.com/exdb/mnist/`

with 5,000 unlabeled samples and varying sizes of labeled data for training, and Table 6 presents the classification accuracy of the proposed method with 50 labeled samples and varying sizes of unlabeled data. Table 5 shows that our proposed method has significantly higher classification accuracy compared with other competitors. Further, with only 50 labeled samples, our method can achieve a 93.7% test accuracy. Table 6 indicates that our proposed method works reasonably well as long as the unlabeled samples are used, as it has a higher classification accuracy than the supervised methods as shown in Table 5. Moreover, the classification accuracy of our method increases as the size of the unlabeled data gets larger.

Table 5: Classification accuracy of the methods with different sizes of labeled samples on MNIST test data.

	Supervised methods			Semi-supervised methods		
n	LR	NLR	cGAN	SSEM	GSSC-S	GSSC-G
16	0.099	0.610	0.580	0.612	0.632	0.613
24	0.103	0.634	0.583	0.623	0.843	0.715
32	0.132	0.641	0.593	0.646	0.819	0.799
40	0.144	0.668	0.632	0.663	0.820	0.812
50	0.163	0.715	0.700	0.756	0.937	0.928

Notes: n is the size of labeled data and the size of unlabeled data is 5000.

Table 6: Classification accuracy of the proposed generative semi-supervised method with different sizes of unlabeled samples on MNIST test data.

N	100	200	1000	2000	5000	20000
GSSC-G	0.772	0.803	0.873	0.924	0.928	0.920
GSSC-S	0.837	0.834	0.893	0.938	0.937	0.913

Notes: N is the size of unlabeled data and the size of labeled data is 50.

6 Conclusions

In this paper, we have proposed GSSC, a novel generative semi-supervised classification method. This method uses a conditional class generator function for the class probability so that the unlabeled data can be naturally included in learning the classifier. We use deep neural networks to approximate the conditional generator and the conditional class probability function. This enables us to learn a classification rule nonparametrically. Our theoretical analysis shows that the estimated conditional class generator and the conditional class probability function are consistent under mild conditions. Our numerical studies demonstrate that GSSC has good or better performance relative to the existing methods considered in the comparisons.

In this work, we have used the least squares loss (7) to match the conditional class probability and the expected conditional generator function. This loss allows us to incorporate unlabeled data in the analysis. Other types of losses can be used here. For example, we can also use a cross entropy type of loss here. It would be interesting to examine the performance of different losses empirically and theoretically in the future. Another interesting problem is to study whether the proposed method can be extended to the case with possible covariate shift, that is, the marginal distribution of the covariate in the labeled data may be different from that of the covariate distribution of unlabeled data. We hope to study these problems in the future.

Acknowledgements

Y. Lin is supported by the Hong Kong Research Grants Council (Grant No. 14306219 and 14306620), the National Natural Science Foundation of China (Grant No. 11961028) and

Direct Grants for Research, The Chinese University of Hong Kong. G. Shen and J. Huang are partially supported by the research grant P0042888 from The Hong Kong Polytechnic University. The authors wish to thank Professor Fang Fang for his helpful comments on the paper.

Supplementary materials

The supplementary materials contain technical proofs of the theoretical results in Section 4 and additional numerical experiments.

References

Anthony, M. and Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations.* Cambridge University Press.

Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. (2019). Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *J. Mach. Learn. Res.*, 20(1):2285–2301.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3(Nov):463–482.

Bhattacharya, P. K. and Gangopadhyay, A. K. (1990). Kernel and Nearest-Neighbor Estimation of a Conditional Quantile. *Ann. Stat.*, 18(3):1400 – 1415.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In Bartlett, P. L. and Mansour, Y., editors, *Proc. 11th Conf. Comput. Learn. Theory*, pages 92–100.

Bos, T. and Schmidt-Hieber, J. (2022). Convergence rates of deep relu networks for multiclass classification. *Electron. J. Stat.*, 16(1):2724–2773.

Bott, A.-K. and Kohler, M. (2017). Nonparametric estimation of a conditional density. *Ann. Inst. Stat. Math.*, 69(1):189–214.

Cai, T. T. and Guo, Z. (2020). Semisupervised inference for explained variance in high dimensional linear regression and its applications. *J. R. Statist. Soc. B*, 82(2):391–419.

Chakrabortty, A. and Cai, T. (2018). Efficient and adaptive linear regression in semi-supervised settings. *Ann. Stat.*, 46(4):1541–1572.

Chakrabortty, A., Dai, G., and Carroll, R. J. (2022). Semi-supervised quantile estimation: Robust and efficient inference in high dimensional settings. *arXiv:2201.10208*.

Cheng, D., Ananthakrishnan, A., and Cai, T. (2021). Efficient and robust semi-supervised estimation of average treatment effects in electronic medical records data. *Biometrics*, 77(2):413–423.

Cox, D. R. (1958). The regression analysis of binary sequences. *J. R. Statist. Soc. B*, 20(2):215–232.

Deng, S., Ning, Y., Zhao, J., and Zhang, H. (2020). Optimal semi-supervised estimation and inference for high-dimensional linear regression. *arXiv:2011.14185*.

Fan, J., Yao, Q., and Tong, H. (1996). Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proc. 27th Adv. Neural Inf. Process. Syst.*, pages 2672–2680.

Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised learning by entropy minimization. In Lawrence K. Saul, Y. W. and Bottou, L., editors, *Proc. 17th Adv. Neural Inf. Process. Syst.*, pages 529–536.

Gronsbell, J. L. and Cai, T. (2018). Semi-supervised approaches to efficient evaluation of model prediction performance. *J. R. Statist. Soc. B*, 80(3):579–94.

Hastie, T. and Tibshirani, R. (1987). Non-parametric logistic and proportional odds regression. *J. R. Statist. Soc. C*, 36(3):260–276.

Izbicki, R. and Lee, A. B. (2017). Converting high-dimensional regression to high-dimensional conditional density estimation. *Electron. J. Stat.*, 11(2):2800–2831.

Jiao, Y., Shen, G., Lin, Y., and Huang, J. (2021). Deep nonparametric regression on approximately low-dimensional manifolds. *arXiv preprint arXiv:2104.06708*.

Kallenberg, O. (2002). *Foundations of Modern Probability*, volume 2. Springer.

Khosravi, I. and Alavipanah, S. K. (2019). A random forest-based framework for crop mapping using temporal, spectral, textural and polarimetric observations. *Int. J. Remote Sens.*, 40(18):7221–7251.

Kim, Y., Ohn, I., and Kim, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Netw.*, 138:179–197.

Kingma, D. P. and Adam, J. B. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *Proc. 3rd Int. Conf. Learn. Represent.*

Lu, J., Shen, Z., Yang, H., and Zhang, S. (2021). Deep network approximation for smooth functions. *SIAM J. Math. Anal.*, 53(5):5465–5506.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *Proc. 7th IEEE Appl. Comput. Vis. Workshops*, pages 29–36.

Shen, G., Jiao, Y., Lin, Y., and Huang, J. (2021). Non-asymptotic excess risk bounds for classification with deep convolutional neural networks. *arXiv preprint arXiv:2105.00292*.

Shen, Z., Yang, H., and Zhang, S. (2019). Deep network approximation characterized by number of neurons. *arXiv preprint arXiv:1906.05497*.

Tsybakov, A. B. (2004). *Introduction to Nonparametric Estimation*. Springer.

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

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. 33rd Annual Meet. Assoc. Comput. Linguist.*, pages 189–196.

Zhou, D., Hofmann, T., and Schölkopf, B. (2004). Semi-supervised learning on directed graphs. In Saul, L., Weiss, Y., and Bottou, L., editors, *Proc. 17th Adv. Neural Inf. Process. Syst.*, pages 1633–1640.

Zhou, X., Jiao, Y., Liu, J., and Huang, J. (2022). A deep generative approach to conditional sampling. *J. Am. Stat. Assoc.*, 0:1–12.

Zhu, X. and Lafferty, J. (2005). Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proc. 22nd Int. Conf. Mach. Learn.*, pages 1052–1059.