Nonparametric learning with matrix-valued predictors in high dimensions

Chanwoo Lee[1], Lexin Li[2], Hao Helen Zhang[3], and Miaoyan Wang[1]

[1]Department of Statistics, University of Wisconsin-Madison

[2]Department of Biostatistics and Epidemiology, University of California at Berkeley

[3]Department of Mathematics, University of Arizona

Abstract

We consider the problem of learning the relationship between a binary label response and a high-dimensional matrix-valued predictor. Such data problems are important in brain imaging studies, sensor network localization, and personalized medicine. Existing supervised analysis typically takes a parametric procedure by imposing a pre-specified relationship between variables. Parametric methods, however, often perform poorly under misspecified models, especially in the high dimension, low sample size settings. Here, we develop a learning reduction framework to address a range of learning tasks from classification to regression that account for matrix-valued predictors. The proposal achieves interpretable prediction using a low-rank two-way sparse representation of the target function. Unlike earlier approaches, our regression model is distribution-free and adapts to the possibly non-smooth, non-linear function of interest. Estimation consistency, and convergence rate are established. We demonstrate the advantage of our method over previous approaches through numerical analyses and applications.

Keywords: Nonparametric learning, high-dimensional matrix-valued predictors, sparse and low-rank models, classification, regression, feature selection

1 Introduction

Consider a statistical learning setting where we would like to model the relationship between a matrix-valued predictor $\boldsymbol{X} \in \mathbb{R}^{d\times d}$ and a binary label response $Y \in \{-1, 1\}$. Matrix-valued predictors ubiquitously arise in modern applications. One example is from electroencephalography studies of alcoholism. The data set records voltage value measured from 64 channels of electrodes on 256 subjects for 256 time points (**?**). Each feature is a 256×64 matrix and the response is a binary indicator of subject being alcoholic or control. Another example is pedestrian detection from image

data. Each image is divided into 9 regions where local orientation statistics are generated with a total of 22 numbers per region. This yields a 22×9 matrix feature and a binary label response indicating whether the image is pedestrian (**?**). Our motivating problem comes from brain network analysis. In this study, each individual in the sample is represented by their own brain network, and the nodes (brain regions of interest) shared across all networks are mapped onto a common atlas. Human connectom project (**?**) has 68 brain nodes extracted from six functional brain regions. A connectivity is computed for every pair of nodes, resulting in an adjacency matrix of size 68×68 for each individual. This brain connectivity network is used to predict the individual's disease probability.

The key challenge with matrix-valued predictors is the high-dimensional, complex structure in the feature space. One approach is to transform the feature matrices to vectors and apply classical methods such as Lasso (**?**). However, this vectorization would destroy the structural information of the data matrices. Matrix-valued predictors represent various aspects of data features, including global structure (e.g. clustering patterns, community hubs) and local structure (e.g. node degrees, edge connections). The vectorization cannot consider those aspects. Traditionally, many learning problems are addressed through distribution assumption such as logistic regression, linear discriminant analysis (LDA) (**?**), and group lasso (**?**). In many applications, however, it is often difficult to justify the assumptions. Groupo lasso is only designated for network data and the model assumptions made on logistic regression or LDA become challenging for matrix features because of high dimensionality. For those reasons, nonparametric approaches such as k-nearest neighbors, decision trees, and convolutional neural network (CNN) (**?**) have been popular in prediction problem. In the above examples and many other studies, however, researchers are interested in *interpretable prediction*, where the goal is to not only make accurate prediction but also identify features that are informative to the prediction. The current nonparametric methods lack interpretability that can help researchers to find out new scientific facts.

We propose new learning framework that respects the matrix structure of the predictors and produces interpretable results without distribution assumption. We consider three main learning problems: classification, level set estimation, and regression in a relation to the above examples. We utilize a low-rank two-way sparse matrix representation of the targeted regression function. The

representation enables efficient variable selection in the high-dimensional matrix learning. We achieve theoretical guarantees for classification errors and derive rates of convergence of the proposed regression in L1 sense. Our numerical analyses suggest that the proposed method is highly competitive and applicable.

Notation: Let f be a function from $\mathcal{X}$ to $\mathbb{R}$. We denote sign$f$ as the sign function such that sign$f(\boldsymbol{X}) = 1$ if $f(\boldsymbol{X}) > 0$ and sign$f(\boldsymbol{X}) = -1$ if $f(\boldsymbol{X}) \leq 0$, for all $\boldsymbol{X} \in \mathcal{X}$. Let $\mathbb{1}(\cdot)$ denote the indicator function. A set A uniquely determines an indicator function $\mathbb{1}(\boldsymbol{X} \in A)$. We use the shorthand $\text{sign}(\boldsymbol{X} \in A) \stackrel{\text{def}}{=} 2\mathbb{1}(\boldsymbol{X} \in A) - 1$ to denote the sign function induced by the set A. Given a d_1-by-d_2 matrix $\boldsymbol{B}$, we use $\boldsymbol{B}_i$ to denote the i-th row of $\boldsymbol{B}$, where $i \in [d_1]$. We use $\|\cdot\|_p$ to denote the p-norm for vectors for $p \geq 0$. The (p, q)-norm of a matrix $\boldsymbol{B}$ is defined as $\|\boldsymbol{B}\|_{p,q} = \|\boldsymbol{b}\|_q$, where $\boldsymbol{b} = (\|\boldsymbol{B}_1\|_p, \ldots, \|\boldsymbol{B}_{d_1}\|_p) \in \mathbb{R}^{d_1}$ consists of the p-norms for each of the rows in $\boldsymbol{B}$. In particular, $\|\boldsymbol{B}\|_{1,0} = \#\{i \in [d_1] \colon \boldsymbol{B}_i \neq 0\}$ denotes the number of non-zero rows in $\boldsymbol{B}$. We denote $a_n \asymp b_n$ if $\lim_n b_n/a_n \to c$ for some constant $c >$ and denote $a_n \lesssim b_n$ if $\lim_n b_n/a_n \to 0$.

2 Three learning problems

In this section we present the main learning goals of our interest. Let $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$ denote the matrix-valued predictor, $y \in \{-1, 1\}$ denote the binary label response, and $\mathbb{P}_{\boldsymbol{X},Y}$ denote the unknown joint probability distribution over the pair $(\boldsymbol{X}, y)$. Suppose that we observe a sample of n training data points, $(\boldsymbol{X}_1, y_1), \ldots, (\boldsymbol{X}_n, y_n)$, identically and independently distributed (i.i.d.) according to $\mathbb{P}_{\boldsymbol{X},y}$. Let $(\boldsymbol{X}_{\text{new}}, y_{\text{new}})$ be a new test point drawn independently from the same distribution. Our goal is to predict y_{new} given the new feature value $\boldsymbol{X}_{\text{new}}$. When no confusion arises, we often omit the subscript "new" and simply write $(\boldsymbol{X}, y)$ for the prototypical test point. Note that y is a Bernoulli random variable with conditional probability $p(\boldsymbol{X}) \stackrel{\text{def}}{=} \mathbb{P}(y = 1|\boldsymbol{X})$, and we generally make no parametric assumptions on the marginal distribution $\mathbb{P}_{\boldsymbol{X}}$ or form of $p(\boldsymbol{X})$.

We consider three learning problems: classification, level set estimation, and regression.

2.1 *Classification*: Classification is the problem of predicting the label $y \in \{-1, 1\}$ to which a new observation $\boldsymbol{X}$ belongs. A prediction rule (also called a classifier) decides that $y = 1$ if $\boldsymbol{X} \in S$ and $y = 0$ if $\boldsymbol{X} \notin S$, where S is a Borel subset of $\mathbb{R}^{d_1 \times d_2}$. Because a set S uniquely defines an $\{-1, 1\}$-valued sign function in the predictor space, we will use the term "classifier" to denote both the set

S and its associated sign function. We formulate the classification problem as choosing a classifier $S \in \mathcal{S}$, from a given set of candidate classifiers $\mathcal{S}$, that minimizes the expected classification error,

$$R(S) = \mathbb{P}\left[y \neq \text{sign}(\boldsymbol{X} \in S)\right]. \tag{1}$$

The expected classification error $R(S)$ is also referred to as the classification risk. When the candidate set $\mathcal{S}$ consists of all Borel subsets of $\mathbb{R}^{d_1 \times d_2}$, the minimizer of (2.1) is called the Bayes classifier. In practice the population joint distribution $\mathbb{P}_{\boldsymbol{X},y}$ is unknown, so the objective function (2.1) and the minimizer need to be estimated through the data. Our first goal is to estimate the Bayes classifier for matrix classification:

Question 1. When matrix dimension $d_1 d_2$ far exceeds the sample size n, how to efficiently perform matrix classification without much assumption on $\mathbb{P}_{\boldsymbol{X},y}$?

2.2 *Level set estimation*: The problem of level set estimation generalizes the classification task. For a given $\pi \in (0,1)$, the π-level set of the conditional probability function $p(\boldsymbol{X})$ is defined as

$$S_{\text{bayes}}(\pi) = \{\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2} : p(\boldsymbol{X}) \geq \pi\}.$$

It is known that the set $S_{\text{bayes}}(\pi)$ optimizes the weighted classifier risk (**???**). Specifically, among all Borel subsets of $\mathbb{R}^{d_1 \times d_2}$, the set $S_{\text{bayes}}(\pi)$ is the global minimizer of the expected π-weighted classification error,

$$R_\pi(S) = \mathbb{E}\left[w_\pi(y)\mathbb{1}(y \neq \text{sign}(\boldsymbol{X} \in S))\right], \tag{2}$$

where we define $w_\pi(y) = 1 - \pi$ or π depending on $y = 1$ or -1. In light of (2.1) and (2.2), the level set problem is an extension of the usual classification from equal weight $\pi = 1/2$ to general weight $\pi \in (0,1)$. Accurate level set estimation plays an important role in applications of geographical elevation maps, imaging contour detection, and motion tracking. We consider the following question:

Question 2. How to simultaneously estimate the level set and identify important variables in the matrix-valued predictors $\boldsymbol{X}$, for the goal of interpretable prediction?

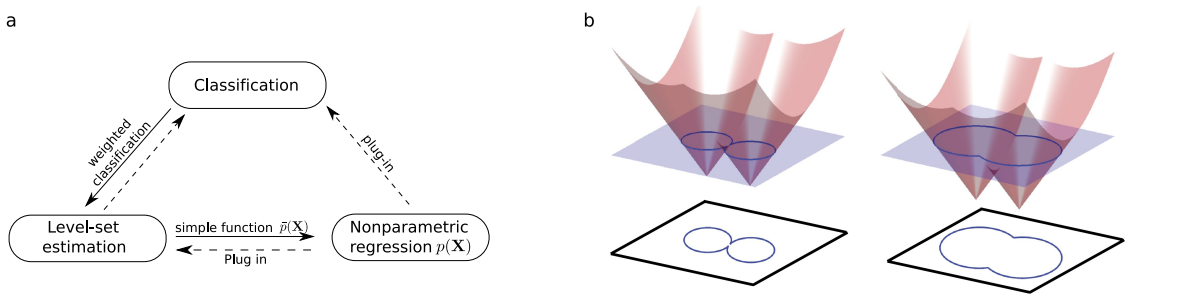

Figure 1: (a) Our learning reduction approach (solid line) to three learning problems and classical plug-in approaches (dashed line). (b) Schematic diagram for nonparametric function estimation via level set estimation. Figure (b) is modified from **?**.

2.3 *Nonparametric regression*: The problem of nonparametric regression is to estimate the conditional mean $\mathbb{E}(y|\boldsymbol{X})$ as a multivariable function in the predictor space. In the our contexts, the nonparametric regression is equivalent to the estimation problem of conditional probability $p(\boldsymbol{X}) = \mathbb{P}(y = 1|\boldsymbol{X}) = \frac{1}{2}(\mathbb{E}(y|\boldsymbol{X}) + 1)$. Throughout the paper we will focus on $p(\boldsymbol{X})$ and refer to it as the regression function. Our final goal is the function estimation:

Question 3. How to learn the regression function $p(\boldsymbol{X})$ in the high-dimensional matrix space?

The three problems of our interest represent a range of learning tasks with increasing difficulties. Classification is a special case of level set estimation with $\pi = 1/2$, whereas the level set is a discrete approximation of the regression function. A common approach is to address regression first, and then solve the earlier two using plug-in estimates (Figure 1a). This procedure, however, undermines the fact that regression is generally harder than the other two. Indeed, as we show in Section 4, regression has a slower convergence rate compared to that of classification. Ignorance of the increased complexity violates Vapnik's maxim: *When solving a given problem, one should try to avoid solving a more general problem as an intermediate step.* We follow Vapnik's principle to develop a "learning reduction" approach by relating the regression to classification, the latter of which is more fundamental and easier to address. In the next section, we propose to address classification first and solve the regression based on "learning reduction" framework (Figure 1b).

3 Estimation

Our building block is to use level sets to estimate regression function p through classifications. The level set approach bridges the two sides of a same coin – characteristic (set indicator) functions in functional analysis and weighted classifications in statistical learning. Let $\Pi = \{\frac{1}{H}, \frac{2}{H} \ldots, \frac{(H-1)}{H}\}$

be a sequence of evenly spaced points in $[0,1]$, where $H \in \mathbb{N}_+$ is the resolution parameter. We introduce an H-step function from $\mathbb{R}^{d_1 \times d_2}$ to $[0,1]$,

$$\hat{p}(\boldsymbol{X}) = \frac{1}{2H} \sum_{\pi \in \Pi} \text{sign}(\boldsymbol{X} \in \hat{S}(\pi)) + \frac{1}{2},$$

where, for every $\pi \in \Pi$, the set $\hat{S}(\pi) \subset \mathbb{R}^{d_1 \times d_2}$ is estimated classifiers we will obtain from (3). Figure 2 shows the schematic diagram of our approaches. We use a sequence of weighted classifications to estimate the level sets $S_{\text{bayes}}(\pi)$ in the matrix space, and then approximate the target regression function $p(\boldsymbol{X})$ via the function $\hat{p}(\boldsymbol{X})$.

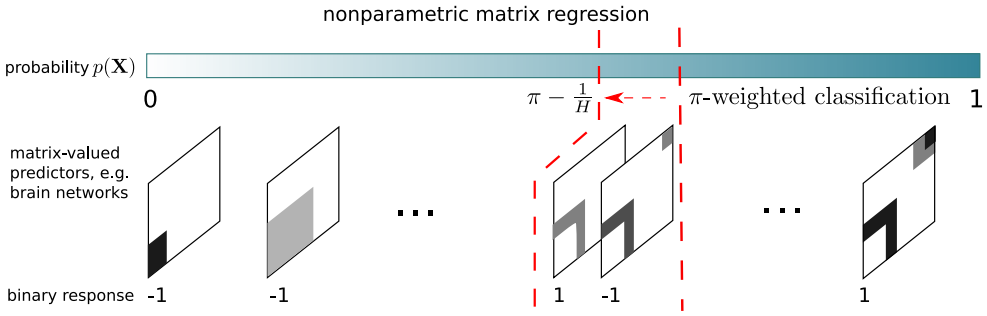

Figure 2: Matrix nonparametric regression via weighted classification. We use a sequence of π-weighted classification to find level sets in the matrix space, and then approximate the target regression function.

For the level set estimation $\hat{S}(\pi)$, we consider the weighted matrix classification with general $\pi \in (0,1)$. The usual classification is naturally obtained by setting $\pi = 1/2$. We propose to estimate the matrix classifier based on penalized empirical surrogate risk minimization,

$$\hat{S}(\pi) = \{\boldsymbol{X} \colon \hat{f}_\pi(\boldsymbol{X}) \geq 0\}, \quad \text{where} \quad \hat{f}_\pi = \min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} w_\pi(y_i) \ell\left(y_i f(\boldsymbol{X}_i)\right) + \lambda \|f\|_F^2 \right\}, \tag{3}$$

where $w_\pi(y) = 1 - \pi$ if $y = 1$ and $w_\pi(y) = \pi$ if $y = -1$ is the weight for two labels, $\mathcal{F}$ is the considered function family, the surrogate loss $\ell(z) \colon \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$ is a non-increasing function of the margin $z = yf(\boldsymbol{X})$, $\lambda > 0$ is the penalty parameter, and we define the penalization term $\|f\|_F$, according to $\mathcal{F}$. Examples of large-margin loss functions are hinge loss $\ell(z) = (1-z)_+$ for support vector machines, logistic loss $\ell(z) = \log(1+e^{-z})$ for important vector machines, and ψ-loss $\ell(z) = 2\min(1, (1-z)_+)$, where $z_+ = \max(z, 0)$. The properties and choices of surrogate loss have been studied earlier (**??**). We choose hinge loss for parsimony; our framework applies equally to

other common large-margin losses (**?**).

The estimation (3) generalizes the population formulation (2.2) from three aspects. First, the population expectation in (2.2) is replaced by the empirical sample average, which is common in statistical learning problems with i.i.d. assumption. Second, we add the ridge penalization $\lambda\|f\|_F^2$ to control the magnitude of the classifiers. The oracle tuning parameter λ depends on the sample size and the problem dimension. In practice, we choose λ in a data-adaptive fashion via cross validation. Third, we replace the binary loss in (2.2) by a more manageable large-margin loss. This relaxation allows us to leverage efficient large-margin algorithms while maintaining desirable statistical performance under mild assumptions.

We propose the linear function family $\mathcal{F}$ with low-rank two-way sparse matrix coefficients,

$$\mathcal{F}(r, s_1, s_2) = \{f\colon \boldsymbol{X} \mapsto \langle \boldsymbol{X}, \boldsymbol{B}\rangle + b \;\big|\; \mathrm{rank}(\boldsymbol{B}) \le r,\ \mathrm{supp}(\boldsymbol{B}) \le (s_1, s_2),\ \boldsymbol{B} \in \mathbb{R}^{d_1 \times d_2},\ b \in \mathbb{R}\}, \quad (4)$$

where $\mathrm{rank}(\boldsymbol{B})$ denotes the rank of the coefficient matrix, and $\mathrm{supp}(\boldsymbol{B})$ denotes the two-way sparsity, with $s_1 = \|\boldsymbol{B}\|_{1,0}$ and $s_2 = \|\boldsymbol{B}^T\|_{1,0}$ being the numbers of non-zero rows and columns of $\boldsymbol{B}$, respectively. We define penalization term in (3) as $\|f\|_F = \|\boldsymbol{B}\|_F$. For the theory, we assume that (r, s_1, s_2) are known; the adaptation to unknown (r, s_1, s_2) is described in Section 5. The low-rank two-way sparse classifier (3) enables efficient variable selection in the high-dimensional matrix learning, thereby achieving interpretability.

4 Statistical properties

To address accuracy of weighted classification and regression, we need to control the behavior of the regression function near the level set boundaries $\partial S_{\mathrm{bayes}}(\pi) = \{p(\boldsymbol{X}) = \pi\}$. We call a level $\pi \in [0, 1]$ a mass point if the level set boundary $\partial S_{\mathrm{bayes}}(\pi)$ has non-zero measure under $\mathbb{P}_{\boldsymbol{X}}$. Let $\mathcal{N} = \{\pi \in [0, 1]\colon \mathbb{P}[p(\boldsymbol{X}) = \pi] \neq 0\}$ denote the collection of mass points in $p(\boldsymbol{X})$.

Definition 1 (Global regularity)**.** A function $p(\boldsymbol{X})$ is α-globally regular, if

(i) $p(\boldsymbol{X})$ has finitely many mass points, i.e., $|\mathcal{N}| \le C'$ for some constant $C' < \infty$; and

(ii) there exists a global constant $C > 0$ such that, for all $\pi \notin \mathcal{N}$,

$$\mathbb{P}(|p(\boldsymbol{X}) - \pi| \leq t) \leq Ct^{\alpha/(1-\alpha)}, \quad \text{for } t \in (0, \rho(\pi, \mathcal{N}))\,,$$

where $\rho(\pi, \mathcal{N}) \stackrel{\text{def}}{=} \min_{\pi' \in \mathcal{N}} |\pi - \pi'|$ denotes the distance from π to the nearest mass point in $\mathcal{N}$. When $\mathcal{N} = \phi$, we define $\rho(\pi, \mathcal{N}) = 1$.

Definition 1 uniformly control the behavior of $p(\boldsymbol{X})$ across possible π. If Deifinition 1(ii) holds for fixed $\pi \in [0, 1]$, we call a function $p(\boldsymbol{X})$ $(\pi, \alpha)-$locally regular. The α quantifies the concentration of probability mass $p(\boldsymbol{X})$ and determine the estimation accuracy. $\mathcal{N}$ is a collection of the weights whose classification task is difficult because the deviation of $p(\boldsymbol{X})$ from π is too small. We make the number of poor weighted classifications negligible in regression from Definition 1(i).

We introduce some additional notations to establish the level set estimation accuracy. Let $f_{\text{bayes},\pi}(\boldsymbol{X}) = I(\boldsymbol{X} \in S_{\text{bayes}}(\pi)) = I(\boldsymbol{X} \colon p(\boldsymbol{X}) \geq \pi)$ denote the set indicator function corresponding to the π-level set, we will refer it to as the level set function. Let $R_\pi(\text{sign}f) = \mathbb{E}\left[w_\pi(y)\mathbb{1}\{y \neq \text{sign}f(\boldsymbol{X})\}\right]$ denote the weighted classification risk, and $R_{\ell,\pi}(f) = \mathbb{E}\left[w_\pi(y)\ell(yf(\boldsymbol{X}))\right]$ denote the surrogated weighted classification risk.

Assumption 1 (Approximation error)**.** There exists a sequence of functions $f_n^* \in \mathcal{F}(r, s_1, s_2)$ for which the surrogate excess risk vanishes; i.e., $R_{\ell,\pi}(f_n^*) - R_{\ell,\pi}(f_{\text{bayes},\pi}) \leq a_n \to 0$ for some vanishing sequence $a_n \to 0$ as $n, d_1, d_2 \to \infty$. Denote $J_n = \|f_n^*\|_F^2$, where we allow J_n to grow with n.

Assumption 1 ensures the vanishing difference in the surrogate risk attained by $\mathcal{F}(r, s_1, s_2)$ and $f_{\text{bayes},\pi}$.

We now provide the accuracy guarantee for the level set estimation (3). For simplicity of notation, we assume $d_1 = d_2 = d$ and $\|\boldsymbol{X}\|_F \leq 1$ with probability 1. Furthermore, the intercept b in $\mathcal{F}$ is assumed known; this is purely for technical convenience. The result demonstrates the statistical consistency of our classifier even when the matrix dimension d far exceeds the sample size n.

Theorem 4.1 (Accuracy for weighted classification)**.** Fix $\pi \in (0, 1)$. Consider the problem of π-level set estimation for a (π, α)-locally regular function $p(\boldsymbol{X})$ with $\alpha \in [0, 1]$. Let $\hat{f}_\pi$ be the level set estimate in (3) with penalty parameter $\lambda \asymp \left(\frac{r(s_1+s_2)\log d}{nJ_n}\right)^{1/(2-\alpha)}$. Suppose Assumption 1 holds.

Then, with probability at least $1 - d^{-C_1 r(s_1+s_2)}$, the classification excess risk is bounded by

$$R_\pi(\operatorname{sign}\hat{f}_\pi) - R_\pi(f_{\text{bayes},\pi}) \leq \max\left\{ C_2 \left(\frac{r(s_1+s_2)\log d}{n} \right)^{1/(2-\alpha)}, a_n \right\}, \tag{5}$$

where $C_1, C_2 > 0$ are two constants. Notice that the usual classification corresponds to $\pi = 1/2$.

To gain insight from the results (4.1), consider the case when the statistical error dominates the approximation error in that $\left(\frac{r(s_1+s_2)\log d}{n}\right)^{1/(2-\alpha)} \gtrsim a_n$. Then, the bound (4.1) immediately implies the classification consistency in the high dimensional regime $d, n \to \infty$, as long as the matrix dimension d grows sub-exponentially in sample size n; i.e., $d = o(e^n)$. This remarkable sample complexity highlights the benefit of low-rank two-way sparse model and structural risk minimization approaches to matrix classification.

We now present accuracy guarantees for our nonparametric matrix regression. There are three sources of error to consider in our learning framework: the statistical error in classification due to finite sample size, the approximation error due to the size of the function space $\mathcal{F}(r, s_1, s_2)$, and an additional approximation error due to learning reduction from classification to regression.

Theorem 4.2 (Accuracy for nonparametric matrix regression)**.** Let $p(\boldsymbol{X})$ be a α-globally regular function with $\alpha \in [0, 1]$. Suppose Assumption 1 holds for all $\pi \in \Pi \setminus \mathcal{N}$. Set the penalty parameter $\lambda \asymp \left(\frac{r(s_1+s_2)\log d}{n}\right)^{1/(2-\alpha)}$. Then, there exists a constant $C > 0$, such that, with probability at least $1 - d^{-C(s_1+s_2)}$,

$$\mathbb{E}|\hat{p}(\boldsymbol{X}) - p(\boldsymbol{X})| \lesssim \underbrace{\frac{1}{H}}_{\text{reduction error}} + \underbrace{\left(\frac{r(s_1+s_2)\log d}{n}\right)^{\frac{\alpha}{2-\alpha}} + H\left(\frac{r(s_1+s_2)\log d}{n}\right)^{\frac{1}{2-\alpha}}}_{\text{statistical error}} + \underbrace{a_n^\alpha}_{\text{approximation error}},$$

Theorem 4.2 demonstrates the high dimensional consistence of our nonparametric matrix regression, provided that the true level set boundaries are well approximated by $\mathcal{F}(r, s_1, s_2)$.

Corollary 4.1 (High-dimensional consistency)**.** Consider the same set-up as in Theorem 4.2. Assume $a_n \lesssim \left(\frac{r(s_1+s_2)\log d}{n}\right)^{1/(2-\alpha)}$ and set $H \asymp \left(\frac{n}{r(s_1+s_2)\log d}\right)^{1/(4-2\alpha)}$. Then, there exists a constant

$C > 0$, with probability at least $1 - d^{-Cr(s_1+s_2)}$,

$$\mathbb{E}|\hat{p}(\boldsymbol{X}) - p(\boldsymbol{X})| \lesssim \left(\frac{r(s_1+s_2)\log d}{n}\right)^{\min(1/2,\alpha)/(2-\alpha)}, \quad \text{as } d, n \to \infty \text{ while } d = o(e^n), \tag{6}$$

We conclude this section by comparing the errors in regression (4.1) and classification (4.1). We confirm that the regression error $(n^{-1}\log d)^{\alpha/(4-2\alpha)}$ is slower than the corresponding classification rate $(n^{-1}\log d)^{\alpha/(2-\alpha)}$. Our learning reduction approach successfully handle three problems sequentially and achieve theoretical guarantees based on Vapnik's maxim. This general principle may benefit other learning frameworks by adapting methods for the problem at hand from another one which is better understood.

Algorithm 1: Matrix classification and level-set estimation (ADMM)

Input: Data $(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_n, y_n) \in \mathbb{R}^{d_1 \times d_2} \times \{-1, 1\}$, rank r, support (s_1, s_2), hyperparameter λ
Initialize: $\boldsymbol{B}, \boldsymbol{S}, \boldsymbol{\Lambda} = \boldsymbol{0}, \rho = 0.1$
Solve $L(\boldsymbol{B}, \boldsymbol{S}, \boldsymbol{\Lambda}, \rho) = \frac{1}{n}\sum_{i=1}^n w_\pi(y_i)\ell(y_i\langle \boldsymbol{X}_i, \boldsymbol{B}\rangle) + \lambda\|\boldsymbol{B}\|_F^2 + \rho\|\boldsymbol{B} - \boldsymbol{S}\|_F^2 + \langle \boldsymbol{\Lambda}, \boldsymbol{B} - \boldsymbol{S}\rangle$
Do until converges

Update $\boldsymbol{B}$ fixing $\boldsymbol{S}, \boldsymbol{\Lambda}, \rho$:
Solve $L(\boldsymbol{B}|\boldsymbol{S}, \boldsymbol{\Lambda}, \rho) = \frac{1}{n}\sum_{i=1}^n w_\pi(y_i)\ell(y_i\langle \boldsymbol{X}_i, \boldsymbol{B}\rangle) + (\lambda + \rho)\|\boldsymbol{B} - \bar{\boldsymbol{S}}\|_F^2$
$\boldsymbol{B} = \arg\min_{\boldsymbol{B}} L(\boldsymbol{B}|\boldsymbol{S}, \boldsymbol{\Lambda}, \rho)$.

Update $\boldsymbol{S}$ fixing $\boldsymbol{B}, \boldsymbol{\Lambda}, \rho$:
Solve $L(\boldsymbol{S}|\boldsymbol{B}, \boldsymbol{\Lambda}, \rho) = \|\boldsymbol{S} - \bar{\boldsymbol{B}}\|_F^2$, subject to $\boldsymbol{S} \in \mathcal{F}(r, s_1, s_2)$.
$\boldsymbol{S} = \arg\min_{\boldsymbol{S}\in\mathcal{F}(r,s_1,s_2)} L(\boldsymbol{S}|\boldsymbol{B}, \boldsymbol{\Lambda}, \rho)$.

Update $\boldsymbol{\Lambda} = \boldsymbol{\Lambda} + 2\rho(\boldsymbol{B} - \boldsymbol{S})$.

Update $\rho = 1.1\rho$.

Output: Classifier $\hat{f}\colon \boldsymbol{X} \mapsto \langle \boldsymbol{X}, \boldsymbol{B}\rangle$ and π-level set $\hat{S}(\pi) = \{\boldsymbol{X} \in \mathbb{R}^{d_1\times d_2}\colon \hat{f}(\boldsymbol{X}) \geq 0\}$

5 Numerical experiments

In this section, we evaluate the empirical performance of our method, and compare the accuracy with other common approaches. The simulation covers a range of nonlinear, nonsmooth models which do not necessarily follow the assumptions in our proposal. This allows us to fairly assess the performance of various approaches under practical applications. Unless otherwise stated, we consider non-symmetric adjacency matrix $\boldsymbol{X} \in \mathbb{R}^{d\times d}$ and use $y \in \{-1, 1\}$ to denote the response label of the individual. We develop and use Algorithm 1 for optimization in (3).

5.1 Impacts of sample size, matrix dimension, and model complexity

We examine the finite sample accuracy of our proposed method using four experiments. The simulation is based on prospective model $y \sim \text{Ber}(p(\boldsymbol{X}))$, where $p(\boldsymbol{X})$ is a nonlinear function of the network $\boldsymbol{X}$. We simulate networks $\boldsymbol{X}$ using random matrices with i.i.d. entries from Uniform[0,1]. Conditional on the network, a latent multi-variate response $(z_1, z_2) = (\langle \boldsymbol{B}_1, \boldsymbol{X} \rangle, \langle \boldsymbol{B}_2, \boldsymbol{X} \rangle)$ is generated, where $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathcal{F}(r, s, s)$. We then generate nonparametric network effects by applying quantile (nonlinear) transformation to (z_1, z_2) so that the resulting marginals $(\bar{z}_1, \bar{z}_2)$ follow a distribution $N(0,1) \times N(0,1)_-$. where $N(0,1)_-$ denotes the normal distribution but restricted to $\mathbb{R}_-$. We refer to $(\bar{z}_1, \bar{z}_2)$ as the nonlinear predictor. The final Bernoulli probability is set $p(\boldsymbol{X}) = (1 + \exp(-\bar{z}_1))^{-1} \in [1/2, 1)$ if $\bar{z}_1 \geq 0$ and $p(\boldsymbol{X}) = (1 + \exp(-\bar{z}_2))^{-1} \in (0, 1/2)$ otherwise. We set the network dimension $d = 20, 30, \ldots, 60$ and training sample size $n = 100, 150, \ldots, 400$. In each simulation study, we report the summary statistics across 30 replicates.

The first experiment assesses the impact of sample size to classification. We consider the setting $d = 30$. Figure 3a plots the resulting density of $p(\boldsymbol{X})$ induced from the nonlinear function p and distribution of $\boldsymbol{X}$. Figure 3b plots classification error, measured by the excess risk $R(\hat{S}_{\text{bayes}}) - R(S_{\text{bayes}})$ evaluated on test data, versus the sample size for three different model settings $(r, s) = (2, 2), (2, 3)$ and $(4, 4)$. As seen from Figure 3b, the classification error decays polynomially in sample size, which is consistent with our theoretical results. We find that a higher rank or a denser coefficient leads to a higher classification error, as reflected by the upward shift of the curve as (r, s) increases. Indeed, a higher (r, s) implies a higher complexity in the model space, thus increasing the generalization error of classification.

The second experiment evaluates the impact of matrix dimension to classification accuracy. We fix the sample size $n = 200$ and let the matrix dimension d increase. Figure 3c plots the classification error versus the dimension d. We find that the error increases slowly and is well controlled by the log rate. The ability to effectively control massive noisy features highlights the benefit of our method in high dimensions.

The third experiment investigates the task of matrix regression. We set the smoothing parameter $H = 20$ and aggregated the multiple level-sets as in our proposal (3). Figure 3d shows the regression

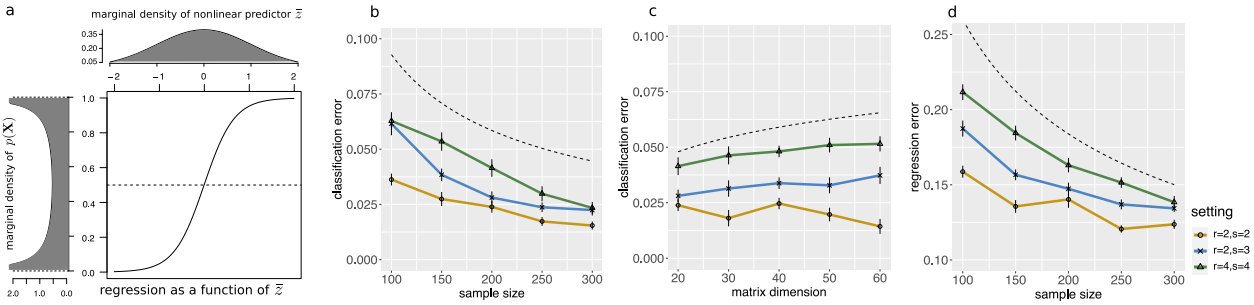

Figure 3: Performance for matrix classification and regression. (a) Relationship between the nonlinear predictor $\bar{z}$ and the Bernoulli probability $p(\boldsymbol{X})$. The distribution for each is plotted on the top and left side, respectively. (b) classification error with sample size. (c) classification error with matrix dimension. (d) regression error with sample size. The dash line in panels (b)-(d) represent theoretical rates $\mathcal{O}(n^{-2/3})$, $\mathcal{O}(\log d)$, and $\mathcal{O}(n^{-1/2})$, respectively.

error measured by L-1 risk, evaluated on test data. Again, we find that the regression error decays polynomially in sample size. Note that our matrix-valued feature has ambient dimension $30 \times 30 = 900$ whereas the sample size is on the order of hundreds. Nevertheless, our nonparametric method consistently learns the function p from limited data without a priori functional form.

5.2 Comparison with other methods

Next, we compare our Nonparametric matrix regression (**NonparaM**) with several popular alternative methods: Unstructured regular lasso (**Lasso**, **?**), Parametric regression for network predictors with group lasso (**LogisticM**, **?**), and Convolutional Neural Network (**CNN**) with two hidden layers implemented in Keras (**?**). We choose a range of representative methods and aims to investigate the benefit of each approach. The **Lasso** serves as a baseline to assess the gain of matrix-valued predictors over vector-valued predictors. Among the three methods that allow matrix-valued inputs, **CNN** and **NonparaM** provide nonparametric approaches and **LogisticM** provide a parametric approach.

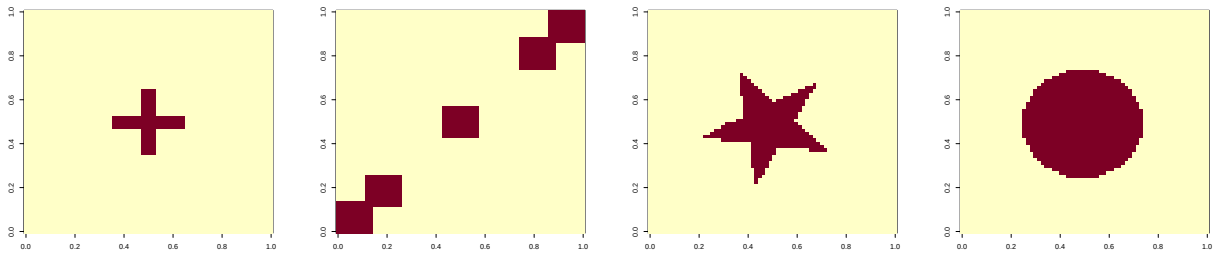

Figure 4: Rank of support matrices: 3 (cross), 7 (block), almost full-rank (star), almost full-rank (circle)

For fair comparison, we adopt similar simulation setup as in **?**, expecting that we add more challenging network patterns in order to assess model misspecification. We draw a sample of π's i.i.d. from Uniform[0,1], and simulate $y|\pi \sim \text{Ber}(\pi)$. Given $\pi \in [0,1]$, the network predictor is simulated from $\boldsymbol{X}|\pi = (\pi - 1/2)^2\boldsymbol{B}_\pi + \boldsymbol{B}_0 + \boldsymbol{E}$, where $\boldsymbol{B}_0$ is a baseline matrix, $\boldsymbol{B}_\pi \in \{0,1\}^{d\times d}$ indicates the active regions of connectivity among d nodes, and $\boldsymbol{E}$ is the noise matrix consisting of i.i.d. N(0, 0.01) entries. The support of the matrix $\boldsymbol{B}_\pi$ varies depending on the range in which $\pi \in [0,1]$ falls. We divide the range $[0,1]$ into several equally spaced intervals, e.g., $[0,1/4],\ldots,[3/4,1]$. Figure 4 illustrates the four patterns of $(\boldsymbol{B}_\pi)_{\pi\in[0,1]}$. We refer to the four patterns as "cross", "block", "star", and "circle". The baseline $\boldsymbol{B}_0$ is set to be a matrix with entry 1 in the depicted region and zero otherwise. We set $d = 68$, a training size $n = 160$, and a test size 80.

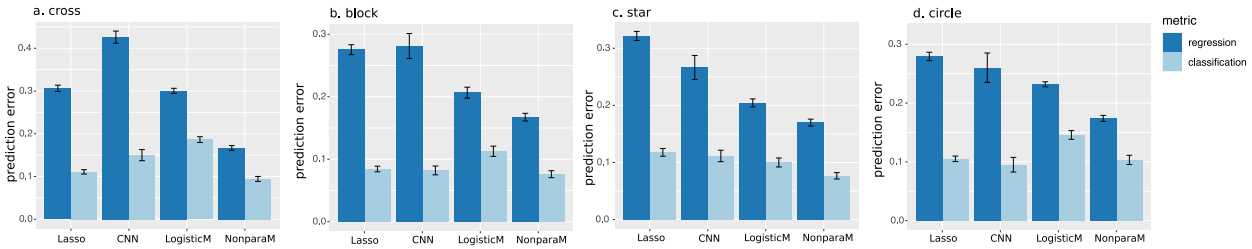

Figure 5: Comparison of prediction errors between various methods. (a)-(d) represent four different activation patterns.

Figure 5 compares the out-of-sample prediction between different methods. For regression problem, we find that **NonparaM** consistently outperforms others, and the reduction in error is substantial. For example, the relative reduction using **NonparaM** over the next best approach, **LogisticM**, is over 20% for models 1 and 4, and over 15% for models 2 and 3. The results show the benefit of our nonparametric approache by allowing more flexible functional space. Furthermore, we find that neither **Lasso** nor **CNN** has satisfactory regression performance. One possible reason is that these two methods fail to appropriately incorporate the network structure of the predictors. The **Lasso** takes vectorized matrices as inputs and therefore losses the two-way pairing information. On the other hand, **CNN** assumes spacial ordering within the $d = 64$ row/column indices. Although local similarity is an appropriate model for imaging, the indexing of rows/columns are meaningless for networks. Methods that are index-invariant (**LogisticM** and **NonparaM**) show better performance. The results demonstrate the advantages of our nonparametric approach on the task of network regression.

We also report the accuracy on classification which is an intermediate step of regression. Figure 5 shows the favorable performance of our method especially when compared to **LogisticM**. Among the four models of active regions, our method performs the best in all three. The only exception is the circle model where the **CNN** has a lower error by a slight margin. This is perhaps due to the fact that circle pattern is nearly full rank which favors complicated models such as **CNN**. Nevertheless, our method **NonparaM** achieves stable performance in spite of its simplicity.

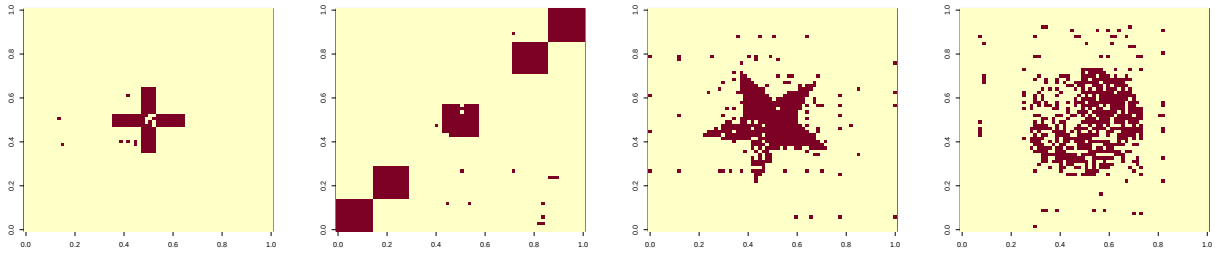

Figure 6: Rank of support matrices: 3 (cross), 7 (block), almost full-rank (star), almost full-rank (circle)

Figure 6 shows the example results returned by **NonparaM**. The results demonstrate that our **NonparaM** enjoys the accurate prediction as a nonparametric approach, while inheriting the interpretability of parametric approaches.

6 Conclusion

We have developed the learning framework for the relationship between a binary label response and a high-ddimensional matrix-valued predictor. Our method respects the matrix structure of the predictors and produces interpretable result with nonparametric approach. Both theoretical and numerical results are provided to demonstrate the comepetetive performance of our estimation. The work unlocks several directions of future research. Extension to multilclass probability estimation and to nonlinear boundaries through kernel methods would be of interest. Application to nonparametic way for matrix completion and denoising problem warrants future research.

References

Agresti, A. and B. A. Coull (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician 52*(2), 119–126.

Bartlett, P. L., M. I. Jordan, and J. D. McAuliffe (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association 101*(473), 138–156.

Chollet, F. et al. (2015). Keras. https://keras.io.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software 33*(1), 1.

Gibou, F., R. Fedkiw, and S. Osher (2018). A review of level-set methods and some recent applications. *Journal of Computational Physics 353*, 82–109.

Relión, J. D. A., D. Kessler, E. Levina, S. F. Taylor, et al. (2019). Network classification with applications to brain connectomics. *The Annals of Applied Statistics 13*(3), 1648–1677.

Scott, C. (2011). Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *ICML*.

Scott, C. and M. Davenport (2007). Regression level set estimation via cost-sensitive classification. *IEEE Transactions on Signal Processing 55*(6), 2752–2757.

Shashua, A., Y. Gdalyahu, and G. Hayun (2004). Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *IEEE Intelligent Vehicles Symposium, 2004*, pp. 1–6. IEEE.

Wang, J., X. Shen, and Y. Liu (2008). Probability estimation for large-margin classifiers. *Biometrika 95*(1), 149–167.

Wang, L., D. Durante, R. E. Jung, and D. B. Dunson (2017). Bayesian network–response regression. *Bioinformatics 33*(12), 1859–1866.

Willett, R. M. and R. D. Nowak (2007). Minimax optimal level-set estimation. *IEEE Transactions on Image Processing 16*(12), 2965–2979.

Zhou, H. and L. Li (2014). Regularized matrix regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(2), 463–483.