Nonparametric learning with matrix-valued predictors in high dimensions

Abstract

We consider the problem of learning the relationship between a binary label response and a high-dimensional matrix-valued predictor. Prediction based on matrices or networks has recently surged in brain connectivity studies, sensor network localization, and integrative genomics. Traditional regression methods take a parametric procedure by imposing a priori functional form between variables. These parametric models, however, are inadequate for structure learning and often fail in accurate prediction. Here, we develop a learning reduction framework to address a range of learning tasks from classification to regression for matrix-valued predictors. Our proposal achieves interpretable prediction via a low-rank two-way sparse representation of the target function. Unlike earlier approaches, our method automatically learns and exploits the important features in the high-dimensional matrices. Statistical accuracy, excess risk bounds, and efficient algorithms are established. We demonstrate the advantage of our method over previous approaches through simulations and applications to human brain connectome data.

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

1 Introduction

Matrix-valued predictors ubiquitously arise in modern applications. In brain connectivity studies, for example, individuals are represented by their brain networks, and the networks quantify the connectivity patterns over a set of nodes (brain regions of interest). Human connectom project (Wang et al., 2019) has constructed brain networks for over 1,200 individuals using Desikan atlas with 68 brain nodes. Structural connectivity is measured for every pair of nodes, resulting in an adjacency matrix of size 68 × 68 for each individual. This connectivity matrix provides important information for disease prediction. Other examples include electroencephalography studies of alcoholism (Zhou and Li, 2014). Researchers measure the voltage values from 64 channels of electrodes on 256 subjects for 256 time points. The study yields a 256 × 64 matrix-valued feature, along with a binary indicator of subject being alcoholic or not. Identifying the relationship between EEG signals and alcoholism is helpful for disease diagnostics.

We consider the statistical learning problem of modeling the relationship between a matrix-valued predictor $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$ and a binary label response $y \in \{-1, 1\}$. The key challenge with matrix-valued predictors is the high-dimensional multi-way structure in the feature space. One possible approach is to transform the predictors into vectors and apply classical methods such as Lasso (Friedman et al., 2010). The practice of vectorization, however, destroys the structural information in the original predictors. Indeed, network data encoded as matrices represent various aspects of features, including global structure (e.g. clustering patterns, community hubs) and local structure (e.g. node degrees, edge connections). Learning and incorporating these features are important for prediction. There have been several recent attempts to allow matrix-valued predictors; for example, trace regression (Fan et al., 2019), network logistic regression (Relión et al., 2019), and matrix linear discriminant analysis (Hu et al., 2020). These parametric approaches impose a priori functional form between variables and often lead to inaccurate prediction in high dimensions. For these reasons, nonparametric approaches such as k-nearest neighbors, decision trees, and convolutional neural network (CNN) have been popular. Current nonparametric methods aim for accurate prediction at the cost of hard interpretability. In our motivating brain network application and many other scientific studies, however, researchers are interested in *interpretable prediction*, where the goal is to not only make accurate prediction but also identify most informative features for descriptive simplicity. Efficient methods that achieve both have yet to be developed.

Our contributions. We develop a nonparametric method that automatically exploits the matrix-valued feature space for accurate prediction. We address three matrix problems – classification, level set estimation, and regression – via a learning reduction approach. The proposal achieves interpretable prediction using a low-rank two-way sparse representation of the target functions. We establish convergence guarantees in high dimensions that permit the matrix dimension to grow with sample size. Unlike earlier approaches, our method performs efficient variable selection and adapts to the possibly non-smooth, non-linear functions of interest. Our numerical analyses and application demonstrate the outperformance of the proposed approach over previous methods.

Notation. Let $\mathcal{X} = \mathbb{R}^{d_1 \times d_2}$ be the feature space. Given a function $f\colon \mathcal{X} \to \mathbb{R}$, we use $\text{sign} f$ to denote its sign function, such that $\text{sign} f(\boldsymbol{X}) = 1$ if $f(\boldsymbol{X}) > 0$ and $\text{sign} f(\boldsymbol{X}) = -1$ otherwise. The notion of sign function also extends to sets in $\mathcal{X}$. We use $\text{sign}(\boldsymbol{X} \in A)$ to denote the sign function induced by the set $A \subset \mathcal{X}$, i.e., a function taking value 1 on the event $\{\boldsymbol{X} \in A\}$ and -1 otherwise. We use shorthand $[n] := \{1, \ldots, n\}$ to denote the n-set for $n \in \mathbb{N}_+$ and use $|\cdot|$ to denote the cardinality of sets. Let $\|\cdot\|_p$ denote the vector p-norm for $p \geq 0$, and $\|\cdot\|_F$ be the matrix Frobenious norm. Given a d_1-by-d_2 matrix $\boldsymbol{B}$, we use $\boldsymbol{B}_i$ to denote the i-th row of $\boldsymbol{B}$. The (p, q)-norm of a matrix $\boldsymbol{B}$ is defined as $\|\boldsymbol{B}\|_{p,q} = \|\boldsymbol{b}\|_q$, where $\boldsymbol{b} = (\|\boldsymbol{B}_1\|_p, \ldots, \|\boldsymbol{B}_{d_1}\|_p)^T \in \mathbb{R}^{d_1}$ 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 $\boldsymbol{B}$. An event E is said to occur "with high probability" if $\mathbb{P}(E)$ tends to 1 as the matrix dimension $d_{\min} = \min(d_1, d_2) \to \infty$. 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 $\lim_n b_n/a_n \to 0$. We use $\mathbb{1}(\cdot)$ to denote the indicator function.

2 Three learning problems

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 $(\boldsymbol{X}, y)$. In the context of binary response, y is a Bernoulli random variable with conditional probability $p(\boldsymbol{X}) \overset{\text{def}}{=} \mathbb{P}(y = 1|\boldsymbol{X})$; we generally make no parametric assumptions on the marginal distribution $\mathbb{P}_{\boldsymbol{X}}$ or form of $p(\boldsymbol{X})$.

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 future unseen test point drawn independently from the same distribution. Our goal is to predict y_{new} based on $\boldsymbol{X}_{\text{new}}$. We often omit the subscript "new" and simply write $(\boldsymbol{X}, y)$ for the prototypical test point. The relevant probabilistic statements should be interpreted as taken jointly with respect to $(\boldsymbol{X}, y)$.

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

2.1 *Matrix classification*: Classification is the problem of predicting the label $y \in \{-1, 1\}$ to which the new matrix $\boldsymbol{X}$ 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 $\mathbb{R}^{d_1 \times d_2}$. 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

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

The $R(S)$ is also called 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 (1) is called the Bayes classifier. It is known that the Bayes classifier can be written as

$$S_{\text{bayes}} = \{\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2} \colon p(\boldsymbol{X}) \geq 1/2\}. \tag{2}$$

The expression (2) is one of many equally good Bayes classifiers, because arbitrary prediction rules are allowed on the boundaries $\partial S_{\text{bayes}} = \{\boldsymbol{X} \colon p(\boldsymbol{X}) = 1/2\}$. Without loss of generality, we will use (2) as the canonical form of the Bayes classifier.

In practice the population distribution is unknown, so the objective function (1) and the minimizer needs to be estimated through the data $\{\boldsymbol{X}_i, y_i\}_{i \in [n]}$. Our first goal is to estimate the Bayes classifier for matrix classification.

Question 1. How to perform classification when matrix dimension far exceeds the sample size n?

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

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

An important fact is that the set $S_{\text{bayes}}(\pi)$ optimizes the weighted classification risk (Willett and

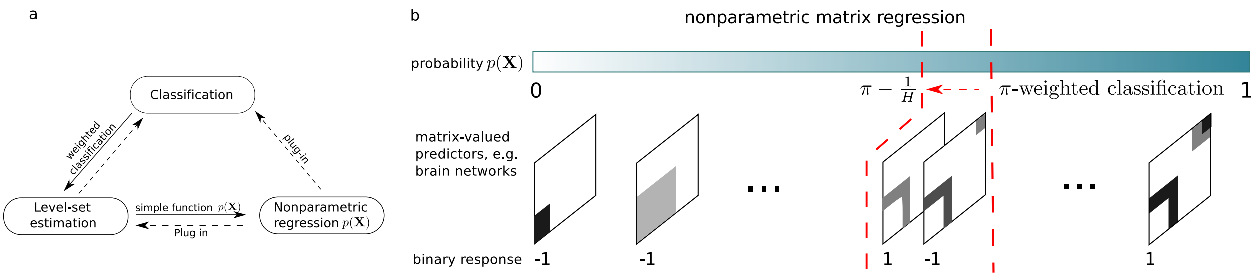
a
Classification
weighted classification
plug-in
Level-set estimation
simple function $\bar{p}(\mathbf{X})$
Plug in
Nonparametric regression $p(\mathbf{X})$
b
nonparametric matrix regression
probability $p(\mathbf{X})$
0
$\pi - \frac{1}{H}$
π-weighted classification
1
matrix-valued predictors, e.g. brain networks
...
...
binary response -1
-1
1
-1
1

Figure 1: (a) Our learning reduction approach (solid line) to the three problems of interest. The classical plug-in approaches are depicted in dashed line. (b) Matrix nonparametric regression via π-weighted classification.

Nowak, 2007; Scott and Davenport, 2007; Wang et al., 2008). 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,

$$R_\pi(S) = \mathbb{E}\left[w_\pi(y)\mathbb{1}(y \neq \text{sign}(\boldsymbol{X} \in S))\right], \tag{3}$$

where we define $w_\pi(y) = 1 - \pi$ or π depending on $y = 1$ or -1. In light of (1) and (3), the level set estimation 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 predictor space, for the goal of interpretable prediction?

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 estimating the 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. The function $p(\boldsymbol{X})$, is the global minimizer, among all measurable functions $f\colon \mathbb{R}^{d_1 \times d_2} \to [0, 1]$, to the expected squared error,

$$R_{\text{reg}}(f) = \mathbb{E}\left[y + 1 - 2f(\boldsymbol{X})\right]^2, \tag{4}$$

where $R_{\text{reg}}(f)$ is also known as regression risk. 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 $\mathcal{O}(n^{-1/2})$ compared to the rate $\mathcal{O}(n^{-1})$ 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.*

3 From classification to regression: a new deal

We develop a "learning reduction" approach (Figure 1a) by relating the regression to classification, the latter of which is more fundamental and easier. We addresses classification first and use the results to solve the regression. In general, regression requires more assumptions than classification. Our learning reduction approach bridges these two tasks using level set estimation, a problem lies somewhere in between. The connection allows us to disentangle complexity and leverage existing algorithms.

In this section we restrict our attention to the population properties of regression function $p(\boldsymbol{X})$ when the true distribution $\mathbb{P}_{\boldsymbol{X},y}$ is known. This simplified situation leads to a cleaner characterization with deterministic risk functions in (1), (3), and (4). The finite sample estimation will be presented in Section 4, in which we address the general case with unknown distribution $\mathbb{P}_{\boldsymbol{X},y}$, and the only information is through empirical (stochastic) risk estimated from the training set $(\boldsymbol{X}_i, y_i)_{i=1}^n$.

3.1 Level set approaches to nonparametric matrix regression

Figure 1b illustrates the main idea of our approaches. We use a sequence of weighted classifications to find the level sets in the matrix space, and then estimate the regression function $p(\boldsymbol{X}) = \mathbb{E}(y = 1|\boldsymbol{X})$ via level set aggregation. 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.

Specifically, let $p(\cdot)\colon \mathbb{R}^{d_1\times d_2} \to [0,1]$ be the target regression function of interest, and $S_{\text{bayes}}(\pi) = \{\boldsymbol{X}\colon p(\boldsymbol{X}) \geq \pi\}$ be the associated π-level set. 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 $\bar{p}(\cdot)\colon \mathbb{R}^{d_1\times d_2} \to [0,1]$ by

$$\bar{p}(\boldsymbol{X}) = \frac{1}{2H}\sum_{\pi\in\Pi} \text{sign}(\boldsymbol{X} \in \bar{S}(\pi)) + \frac{1}{2}, \quad \text{for all } \boldsymbol{X} \in \mathbb{R}^{d_1\times d_2}, \tag{5}$$

where, for every $\pi \in \Pi$, the set $\bar{S}(\pi) \subset \mathbb{R}^{d_1\times d_2}$ is the classifier that minimizes the π-weighted classification risk,

$$\bar{S}(\pi) \overset{\text{def}}{=} \underset{S\in\mathcal{S}}{\arg\min}\, R_\pi(S), \quad \text{where} \quad R_\pi(S) = \mathbb{E}\left[w_\pi(y)\mathbb{1}(y \neq \text{sign}(\boldsymbol{X} \in S))\right], \tag{6}$$

subject to the constraint $S \in \mathcal{S}$, with $\mathcal{S}$ being a given candidate set of classifiers. When the set $\mathcal{S}$

is rich enough, e.g., $\mathcal{S}$ consists of all Borel sets, then $\bar{S}(\pi)$ has the same risk as $S_{\text{bayes}}(\pi)$. We leave the $\mathcal{S}$ in general here; the specific choice of $\mathcal{S}$ will be described in Section 3.2.

In order to address the accuracy between $\bar{p}(\boldsymbol{X})$ and $p(\boldsymbol{X})$, we need to establish the identification of level sets $S(\pi)$ from optimization (6). The Bayes classifier $S_{\text{bayes}}(\pi)$ minimizes the weighted classification risk $R_\pi(S)$; the inverse may not be true because of possible multiple global minimizers of $R_\pi(S)$ even in the ideal scenario with no constraints on $\mathcal{S}$. The uniqueness and stability around $S_{\text{bayes}}(\pi)$ turns out to play a key role for the accurate estimation of $p(\boldsymbol{X})$.

We introduce the following notion to characterize the behavior of the regression function near the level set boundaries $\partial S_{\text{bayes}}(\pi) = \{p(\boldsymbol{X}) = \pi\}$. The condition essentially quantifies the uniqueness of level sets recovery from weighted classification.

Definition 1 (Global regularity)**.** We call a level $\pi \in [0,1]$ a mass point if the level set boundary $\partial S_{\text{bayes}}(\pi)$ has non-zero measures 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 $p(\boldsymbol{X})$. A function $p(\boldsymbol{X})$ is called α-globally regular with $\alpha \in [0,1]$, if

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

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

$$\mathbb{P}_{\boldsymbol{X}}(|p(\boldsymbol{X}) - \pi| \leq t) \leq C t^{\alpha/(1-\alpha)}, \quad \text{for } t \in (0, \rho(\pi, \mathcal{N}))\,, \tag{7}$$

where $\rho(\pi, \mathcal{N}) \overset{\text{def}}{=} \min_{\pi' \in \mathcal{N}} |\pi - \pi'|$ denotes the distance from π to the nearest mass point in $\mathcal{N}$. When $\mathcal{N} = \phi$, we define $\rho(\pi, \mathcal{N}) = 1$. When $\alpha = 1$, the right-hand-side of (7) is interpreted as zero.

Definition 1 controls the uniform behavior of $p(\boldsymbol{X})$ across possible π. If the condition (7) holds for a fixed π, we call the function $p(\boldsymbol{X})$ is (π, α)-locally regular. The exponent α quantifies the concentration of probability mass $p(\boldsymbol{X})$ around level set boundaries.

We show that Definition 1 implies the global identifiability of $S_{\text{bayes}}(\pi)$ from optimization (6) across all $\pi \notin \mathcal{N}$. For two sets $S_1, S_2 \in \mathbb{R}^{d_1 \times d_2}$, we define the probabilistic set difference

$$d_\Delta(S_1, S_2) \overset{\text{def}}{=} \mathbb{P}_{\boldsymbol{X}}(S_1 \Delta S_2) = \mathbb{P}_{\boldsymbol{X}}\{\boldsymbol{X} \colon \boldsymbol{X} \in S_1 \setminus S_2 \text{ or } S_2 \setminus S_1\},$$

and the risk difference

$$d_\pi(S_1, S_2) \overset{\text{def}}{=} R_\pi(S_1) - R_\pi(S_2).$$

Theorem 1 (Identifiability)**.** *Suppose the regression function $p(\boldsymbol{X})$ is α-globally regular with $\alpha \in [0,1]$. Then, there exists a constant $c > 0$ such that*

$$d_\Delta(S, S_{\text{bayes}}(\pi)) \leq c \left[d_\pi^\alpha(S, S_{\text{bayes}}(\pi)) + \frac{1}{\rho(\pi, \mathcal{N})} d_\pi(S, S_{\text{bayes}}(\pi)) \right], \tag{8}$$

for all sets $S \in \mathbb{R}^{d_1 \times d_2}$ and levels $\pi \notin \mathcal{N}$.

The bound (8) controls the worst-case perturbation of classifiers in the probability space $\mathbb{P}_{\boldsymbol{X}}$ with respect to weighted classification risks. When $\alpha \neq 0$, the inequality (8) immediately implies the uniqueness of $S_{\text{bayes}}(\pi)$ up to a measure-zero set in $\mathbb{P}_{\boldsymbol{X}}$ whereas $\alpha = 0$ corresponds to no identifiability. Notice that $\rho(\pi, \mathcal{N})$ becomes a constant for fixed π, implying that local regularity ensures the identifiability of Bayes set $S_{\text{bayes}}(\pi)$.

Our identifiability conclusion generalizes the earlier result for single level set estimation (Singh et al., 2009; Tsybakov et al., 2004). Inspection of the proof shows that the set estimation is more difficult at levels where the point mass concentrates (small α), as intuition would suggest. Consider a simple case when the entries of matrix $\boldsymbol{X}$ are i.i.d. drawn from Uniform$[-1,1]$; that is, $\mathbb{P}_{\boldsymbol{X}}$ is the Lebesque measure on $[0,1]^{d_1 d_2}$. Then, the best rate $\alpha \to 1$ corresponds to a clear separation with no point mass at the boundary, whereas the worst rate $\alpha \to 0$ corresponds to a heavy mass of $p(\boldsymbol{X})$ near the boundary. A typical intermediate case is $\alpha = 1/2$ when $p(\boldsymbol{X})$ has non-degenerate first-order Taylor expansion around π. We omit the proof for space considerations.

Now we reach the main result of this section by putting together the proposal (5), (6) and Theorem 1. Define the regression excess risk by $R_{\text{reg}}(\bar{p}) - R_{\text{reg}}(p)$. The following result bounds the regression excess risk using classification excess risk.

Theorem 2 (Nonparametric regression via weighted classifications)**.** *Suppose that the regression function $p(\boldsymbol{X})$ is α-globally regular with $\alpha \in [0,1]$. Let $\bar{p}(\boldsymbol{X})$ the step function constructed from weighted classifiers as in (5). Then, there exists a constant $c_1, c_2 > 0$ such that*

$$\begin{aligned} R_{\text{reg}}(\bar{p}) - R_{\text{reg}}(p) &\leq 4\mathbb{E}_{\boldsymbol{X}} \left| \bar{p}(\boldsymbol{X}) - p(\boldsymbol{X}) \right| \\ &\leq \frac{2}{H} + \frac{4}{H} \sum_{\pi \in \Pi} \left\{ d_\pi^\alpha(S_{\text{bayes}}(\pi), \bar{S}(\pi)) + H d_\pi(S_{\text{bayes}}(\pi), \bar{S}(\pi)) \right\}. \end{aligned} \tag{9}$$

for all resolution parameter $H = |\Pi| \in \mathbb{N}_+$.

Theorem 2 shows the key role of $\bar{p}(\boldsymbol{X})$ in bridging regression and classification. The results suggest that the estimation of $p(\boldsymbol{X})$ can be reduced to estimation of $\bar{p}(\boldsymbol{X})$, or equivalently, to a sequence of weighted classifications $\{\bar{S}(\pi)\}_{\pi \in \Pi}$. The regression excess risk bound (9) has two terms. The first term is the bias due to the step function approximation to the regression function. The second term is the excess risk for classification optimized over $\mathcal{S}$ compared to that over all Borel sets, representing the capability of the candidate classifiers $\mathcal{S}$. In the case of unknown population risk $R_\pi(\cdot)$, the second term should be plugged in using the empirical risk, which results in an additional variance term due to finite sample size. The resolution parameter H shall be chosen to balances the bias-variance tradeoff.

Estimating $\bar{p}(\boldsymbol{X})$ as a surrogate of $p(\boldsymbol{X})$ provides several benefits. From a computational perspective, $\bar{p}(\boldsymbol{X})$ is a finite combination of weighted classifiers, which are easier to solve than a direct regression. From the statistical perspective, the function $\bar{p}(\boldsymbol{X})$ provides a valid approximation to $p(\boldsymbol{X})$ even when $p(\boldsymbol{X})$ is non-smooth and irregular. In particular, the estimation accuracy relies little on the local neighborhood of $\boldsymbol{X}$ but rather on the local neighborhood of $p(\boldsymbol{X})$. Note that the former is a d_1d_2-dimensional random variable whereas the later is a $[0,1]$-valued univariate random variable. The shifted focus of local structure to the range space is especially appealing for matrix-valued predictors, since the predictor space is high dimensional and often barely explored by small sample size data.

3.2 Sparse and low-rank function boundaries

We describe the choice of candidate classifiers $\mathcal{S}$ in (6). We rewrite the optimization (6) as the minimization over continuous-valued functions,

$$\bar{S}(\pi) = \{\boldsymbol{X}\colon \bar{f}(\boldsymbol{X}) \geq 0\}, \quad \text{with} \quad \bar{f}(\boldsymbol{X}) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[w_\pi(y)\mathbb{1}(y \neq \text{sign} f(\boldsymbol{X}))\right], \tag{10}$$

where $\bar{f} \in \mathcal{F}$ is a continuous-valued function from $\mathbb{R}^{d_1 \times d_2}$, and its sign induces the classifier $\bar{S}(\pi) \in \mathcal{S}$. The choice of $\mathcal{S}$ thus reduces to the choice of function family $\mathcal{F}$. A desirable $\mathcal{F}$ should balance the prediction and interpretability; i.e., $\mathcal{F}$ should be flexible enough for accurate prediction while being simple enough for high interpretability.

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

$$\mathcal{F}(r, s_1, s_2) = \left\{f\colon \boldsymbol{X} \mapsto \langle \boldsymbol{X}, \boldsymbol{B}\rangle + b \;\middle|\; \text{rank}(\boldsymbol{B}) \leq r,\ \text{supp}(\boldsymbol{B}) \leq (s_1, s_2),\ \boldsymbol{B} \in \mathbb{R}^{d_1 \times d_2},\ b \in \mathbb{R}\right\}, \tag{11}$$

where $\text{rank}(\boldsymbol{B})$ denotes the rank of the coefficient matrix, and $\text{supp}(\boldsymbol{B})$ denotes the two-way sparsity parameter 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. For the theory, we assume that (r, s_1, s_2) are known; the adaptation to unknown (r, s_1, s_2) in practice is described in Section 6. Combining formulations (5), (10) and (11) yields our (population version) "learning deduction" approach to nonparametric matrix regression.

The low-rank two-way sparse classifier (11) enables efficient variable selection in high-dimensional matrices, thereby achieving high interpretability in prediction. In the brain network analysis, scientists are interested in identifying important nodes attached to at least one active edges with non-zero effects. Classical entrywise sparsity essentially treats $\boldsymbol{X}$ as "a bag of non-ordered edges", and loses the two-way paring information among entries. In contrast, our two-way sparsity efficiently identifies the underlying active nodes by making use of matrix structure in the predictors.

It is worthy noting that the linearity in the classifiers $\mathcal{F}$ does not preclude the global nonlinearity in the regression function $p(\boldsymbol{X})$ or its variant $\bar{p}(\boldsymbol{X})$. As shown in the following examples, many nonlinear regression functions in existing literature are special cases of our representation (10) with (11), in the sense that the second term in the approximation (9) becomes precisely zero.

Example 1 (Single index models (Alquier and Biau, 2013; Ganti et al., 2017))**.** Suppose the true regression function can be expressed as $p(\boldsymbol{X}) = g(\langle \boldsymbol{X}, \boldsymbol{B}\rangle)$, where $g(\cdot)\colon \mathbb{R} \to [0,1]$ is an arbitrary monotonic function, and $\boldsymbol{B}$ is a low-rank two-way sparse matrix. Then, for every $\pi \in (0,1)$, there exists $f \in \mathcal{F}(r, s_1, s_2)$, such that $\text{sign}(p(\boldsymbol{X}) - \pi) = \text{sign} f(\boldsymbol{X})$. Our method generalizes single index model to high dimensional matrices by joint learning matrix coefficient $\boldsymbol{B}$ and nonlinear function g.

Common link functions, such as logistic function $g(t) = (1+\exp(-t))^{-1}$, arctangent function $g(t) = \frac{1}{\pi}\arctan(t)+\frac{1}{2}$, truncated rectified linear unit (ReLU) function $g(t) = t\mathbb{1}(t \in [0,1])+\mathbb{1}(t > 1)$, and any arbitrary inverse cumulative distribution functions are included in our functional class. In particular, our model generalizes the matrix regressions (Zhou and Li, 2014; Guha and Rodriguez, 2020; Relión et al., 2019) to the case with unknown link functions.

Our function class also incorporates models from matrix linear discriminant analysis (LDA) (Hu et al., 2020).

Example 2 (Multivariate normal mixtures (Hu et al., 2020))**.** Suppose the matrix-valued predictor $\boldsymbol{X}$ follows a Gaussian mixture distribution, $\boldsymbol{X}|\{y=-1\} = \boldsymbol{B}_1 + \boldsymbol{E}_1$ and $\boldsymbol{X}|\{y=1\} = \boldsymbol{B}_2 + \boldsymbol{E}_2$, where $(\boldsymbol{B}_1 - \boldsymbol{B}_2)$ is a low-rank two-way sparse matrix, and $\boldsymbol{E}_1, \boldsymbol{E}_2$ are two mutually independent noise matrices with i.i.d. $N(0,1)$ entries. Then, for every $\pi \in (0,1)$, $\text{sign}(p(\boldsymbol{X})-\pi) = \text{sign}f(\boldsymbol{X})$ for some $f \in \mathcal{F}(r, s_1, s_2)$. More generally, we have established the characterization by extending two classes of $\boldsymbol{X}$ to a series of $\boldsymbol{X} = \boldsymbol{X}(\pi)$ over a continuous spectrum of $\pi \in (0,1)$ (results not shown).

In principle, more complicated classifiers, such as neural network, decision trees, and boosting, can also be brought to bear on the level set construction (10). The ability to import and adapt existing classification methods is one advantage of the proposed learning reduction framework. We find that, in our motivating brain network analysis, the low-rank two-way sparse classifiers (11) provide the benefit of interpretable predictions (see Section 7).

4 Estimation

In previous sections we have established the population properties from classification to regression (Figure 1b). In this section we address the empirical learning problems when the true distribution $\mathbb{P}_{\boldsymbol{X},y}$ is unknown. The objective function in the earlier optimization now becomes empirical (stochastic) risks calculated from high dimensional low sample size training data $(X_i, y_i)_{i=1}^n$.

4.1 Large-margin learning with high dimensional matrices

When the distribution $\mathbb{P}_{\boldsymbol{X},y}$ is unknown, we propose the regression function estimate $\hat{p}(\cdot)\colon \mathbb{R}^{d_1\times d_2} \to [0,1]$ by

$$\hat{p}(\boldsymbol{X}) = \frac{1}{2H}\sum_{\pi\in\Pi}\text{sign}(\boldsymbol{X} \in \hat{S}(\pi)) + \frac{1}{2}, \quad \text{for all } \boldsymbol{X} \in \mathbb{R}^{d_1\times d_2}. \tag{12}$$

Here, for every $\pi \in \Pi$, the set $\hat{S}(\pi) \subset \mathbb{R}^{d_1\times d_2}$ is the estimated classifier from empirical surrogate risk minimization,

$$\hat{S}(\pi) = \{\boldsymbol{X} \colon \hat{f}_\pi(\boldsymbol{X}) \geq 0\}, \quad \text{with} \quad \hat{f}_\pi = \min_{f \in \mathcal{F}(r,s_1,s_2)} \left\{ \frac{1}{n} \sum_{i=1}^{n} w_\pi(y_i) \ell\left(y_i f(\boldsymbol{X}_i)\right) + \lambda \|f\|_F^2 \right\}, \tag{13}$$

where $w_\pi(y) = 1 - \pi$ if $y = 1$ and $w_\pi(y) = \pi$ if $y = -1$ is the label-dependent weight; $\ell(z) \colon \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$ is the surrogate classification loss defined as a function of margin $z = yf(\boldsymbol{X})$; $\lambda > 0$ is the penalty parameter; and we define the penalization term $\|f\|_F = \|\boldsymbol{B}\|_F$, with $\boldsymbol{B}$ being the coefficient matrix associated with $f \in \mathcal{F}(r, s_1, s_2)$. 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 $\ell(z) = 2\min(1, (1 - z)_+)$ with $z_+ = \max(z, 0)$. We choose hinge loss for parsimony; our framework applies equally to other common large-margin losses (Bartlett et al., 2006).

The estimation (13) generalizes the population formulation (10) from three aspects. First, the population expectation in (10) 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 as we will describe in the next paragraph. The resulting sieve estimate enjoys numerical stability and statistical accuracy. In practice, we choose λ in a data-adaptive fashion via cross validation. Third, we replace the binary loss in (10) 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.

4.2 Alternating optimization for structural risk minimization

We describe the optimization algorithm for solving matrix classification and regression. We focus on the general π-weighted classification (13) because both classification and regression naturally follow by setting $\pi = \frac{1}{2}$, and $\pi \in \{\frac{1}{H}, \ldots, \frac{1-H}{H}\}$, respectively. For brevity, we assume the intercept in the function class (11) is zero, and use $\mathcal{F}(r, s_1, s_2)$ to denote the set of matrices satisfying $\text{rank}(\boldsymbol{B}) \leq r$ and $\text{supp}(\boldsymbol{B}) \leq (s_1, s_2)$. The estimation problem (13) is formulated as an optimization over matrix space,

$$\min_{\boldsymbol{B} \in \mathcal{F}(r,s_1,s_2)} L(\boldsymbol{B}), \quad \text{where } L(\boldsymbol{B}) = \frac{1}{n} \sum_{i=1}^{n} w_\pi(y_i) \ell(y_i \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle) + \lambda \|\boldsymbol{B}\|_F^2, \tag{14}$$

where the objective function can be either convex (such as hinge loss, logistic loss) or non-convex (ψ-loss). The optimization (14) has a non-convex feasible region because of the low-rank and sparse constraint.

We propose an alternating direction method of multipliers (ADMM) approach to solve problem of this type. ADMM introduces a dual variable and an additional feasibility constraint to perform coordinate descent in the corresponding augmented Lagrangian function. The augmented ADMM objective in our context is given by

$$L(\boldsymbol{B},\boldsymbol{S},\boldsymbol{\Lambda},\rho)=\frac{1}{n}\sum_{i=1}^{n}w_\pi(y_i)\ell(y_i\langle\boldsymbol{X}_i,\boldsymbol{B}\rangle)+\lambda\|\boldsymbol{B}\|_F^2+\rho\|\boldsymbol{B}-\boldsymbol{S}\|_F^2+\langle\boldsymbol{\Lambda},\boldsymbol{B}-\boldsymbol{S}\rangle, \tag{15}$$

and $\boldsymbol{B}\in\mathbb{R}^{d_1\times d_2}$ is the unconstrained primal variable, $\boldsymbol{S}\in\mathcal{F}(r,s_1,s_2)$ is the constrained dual variable, $\Lambda\in\mathbb{R}^{d_1\times d_2}$ is the Lagrangian multiplier, and $\rho>0$ is the step-size parameter. Note that formulation (15) has moved the non-convexity from the first two terms in $\boldsymbol{B}$ to the last two simpler terms in $\boldsymbol{S}$. This separability of ADMM makes the optimization efficient for a wide range of loss functions and constraints.

We optimize the ADMM objective (15) via coordinate descent, by iteratively update one variable at a time while holding others fixed. Each update in the ADMM reduces to a simpler problem and can be efficiently solved by standard algorithms. Specifically, given variables $(\boldsymbol{S},\boldsymbol{\Lambda},\rho)$ and $\bar{\boldsymbol{S}}\overset{\text{def}}{=}\frac{1}{2(\rho+\lambda)}(2\rho\boldsymbol{S}-\boldsymbol{\Lambda})$, the objective with respect to $\boldsymbol{B}$ is

$$L(\boldsymbol{B}|\boldsymbol{S},\boldsymbol{\Lambda},\rho)=\frac{1}{n}\sum_{i=1}^{n}w_\pi(y_i)\ell(y_i\langle\boldsymbol{X}_i,\boldsymbol{B}\rangle)+(\lambda+\rho)\|\boldsymbol{B}-\bar{\boldsymbol{S}}\|_F^2. \tag{16}$$

This unconstrained optimization is a usual vector-based classification with ridge penalty and an offset $\bar{\boldsymbol{S}}$. Therefore, various loss functions and fast software can be adopted into (16) such as weighted SVM, logistic, and ψ-learning. Similarly, given $(\boldsymbol{B},\boldsymbol{\Lambda},\rho)$ and $\bar{\boldsymbol{B}}\overset{\text{def}}{=}\frac{1}{2\rho}(2\rho\boldsymbol{B}+\boldsymbol{\Lambda})$, the objective with respect to $\boldsymbol{S}$ is

$$L(\boldsymbol{S}|\boldsymbol{B},\boldsymbol{\Lambda},\rho)=\|\boldsymbol{S}-\bar{\boldsymbol{B}}\|_F^2,\quad\text{subject to }\boldsymbol{S}\in\mathcal{F}(r,s_1,s_2). \tag{17}$$

This formulation is equivalent to the best sparse and low rank approximation, in the least-square sense, to the matrix $\boldsymbol{B}$. Compared to the original objective (14), the F-norm based objective makes the optimization easier to handle. A number of algorithms have been designated to approximately solve this problem, including sparse PCA, sparse SVD, and projection pursuit. We use the recently-developed double projection algorithm for (17) which has provably better performance than convex alternatives in high dimensional regimes (Yang et al., 2016). Finally, the Lagrangian multiplier $\boldsymbol{\Lambda}$ is updated by standard scheme $\boldsymbol{\Lambda}\leftarrow\boldsymbol{\Lambda}+2\rho(\boldsymbol{B}-\boldsymbol{S})$. These steps are performed until the algorithm convergence within tolerance. The value ρ controls the closeness between dual and primal variables. We initialize ρ from 0.1 and increases its value geometrically throughout iterations. In practice, we observed this scheme gives a good balance between the variable feasibility and convergence speed, although other self-tuning methods are also possible (Parikh and Boyd, 2014). Algorithm 1 gives the full description for the π-classification.

We develop the nonparametric matrix regression in a similar way with little modification. The nonparametric regression (12) estimates π-classifiers at a sequence of weights $\pi\in\{\frac{1}{H},\ldots,\frac{H-1}{H}\}$. In principal, one can optimize all classifiers jointly subject to nestedness constraints among sequential level sets. However, this strategy would lead to increased computational burden, and the gain in

Algorithm 1: Matrix classification and level-set estimation via ADMM

```
Input: Data $\{(\boldsymbol{X}_i, y_i) \in \mathbb{R}^{d_1 \times d_2} \times \{-1, 1\} \colon i \in [n]\}$, rank $r$, support $(s_1, s_2)$, ridge parameter $\lambda$, the target level $\pi \in \Pi$.
Initialize: primal variable $\boldsymbol{B}$, dual variable $\boldsymbol{S}$, Lagrangian multiplier $\boldsymbol{\Lambda} = \boldsymbol{0}$, step size $\rho = 0.1$.
Objective: $L(\boldsymbol{B}, \boldsymbol{S}, \boldsymbol{\Lambda}, \rho) = \frac{1}{n} \sum_{i=1}^{n} w_\pi(y_i) \ell(y_i \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle) + \lambda \|\boldsymbol{B}\|_F^2 + \rho \|\boldsymbol{B} - \boldsymbol{S}\|_F^2 + \langle \boldsymbol{\Lambda}, \boldsymbol{B} - \boldsymbol{S} \rangle$.
Do until converges
  Update $\boldsymbol{B}$ fixing $(\boldsymbol{S}, \boldsymbol{\Lambda}, \rho)$: $\boldsymbol{B} \leftarrow \arg\min_{\boldsymbol{B}} L(\boldsymbol{B} | \boldsymbol{S}, \boldsymbol{\Lambda}, \rho)$, where
  $L(\boldsymbol{B} | \boldsymbol{S}, \boldsymbol{\Lambda}, \rho) = \frac{1}{n} \sum_{i=1}^{n} w_\pi(y_i) \ell(y_i \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle) + (\lambda + \rho) \|\boldsymbol{B} - \bar{\boldsymbol{S}}\|_F^2$ and $\bar{\boldsymbol{S}} \overset{\text{def}}{=} \frac{1}{2(\rho+\lambda)} (2\rho \boldsymbol{S} - \boldsymbol{\Lambda})$.
  Update $\boldsymbol{S}$ fixing $(\boldsymbol{B}, \boldsymbol{\Lambda}, \rho)$: $\boldsymbol{S} \leftarrow \arg\min_{\boldsymbol{S} \in \mathcal{F}(r, s_1, s_2)} L(\boldsymbol{S} | \boldsymbol{B}, \boldsymbol{\Lambda}, \rho)$ subject to $\boldsymbol{S} \in \mathcal{F}(r, s_1, s_2)$,
  where $L(\boldsymbol{S} | \boldsymbol{B}, \boldsymbol{\Lambda}, \rho) = \|\boldsymbol{S} - \bar{\boldsymbol{B}}\|_F^2$.
  Update $\boldsymbol{\Lambda} \leftarrow \boldsymbol{\Lambda} + 2\rho(\boldsymbol{B} - \boldsymbol{S})$.
  Update $\rho \leftarrow 1.1\rho$.
Output: Estimated $\pi$-level set $\hat{S}(\pi) = \{\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2} \colon \hat{f}(\boldsymbol{X}) \geq 0\}$.
```

accuracy is often little with moderate sample size. We choose to use parallel processing to obtain π-classifiers separately to speed up the computation. The procedure is summarized in Algorithm 2. The software for both matrix classification and regression will be available at CRAN.

Algorithm 2: Nonparamatrix matrix regression

```
Input: $(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_n, y_m)$, rank $r$,pre-specified kernels $K_1, K_2$, and smooth parameter $H$.
Output: Level sets $\hat{S}(\pi_h)$ for $h = 1, \ldots H$ and regression function $\hat{p}(\boldsymbol{X})$.
Initialize: $\pi_h = (h-1)/H$ for $h = 1, \ldots, H+1$
For $h = 1, \ldots, H+1$:
   Level set $\hat{S}(\pi_h)$ estimation:
      Train weighted margin classifier $\hat{f}_{\pi_h}$ from (13) based on Algorithm 1.
      $\hat{S}(\pi_h) = \{\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2} : \text{sign}(\hat{f}_{\pi_h}(\boldsymbol{X})) = 1\}$.
Regression $\hat{p}(\boldsymbol{X})$ estimation:
   $\hat{p}(\boldsymbol{X}) = \sum_{h=1}^{H} \frac{1}{H} \mathbb{1}\left\{\boldsymbol{X} \notin \hat{S}\left(\pi_h\right)\right\}$.
```

5 Statistical learning theory

In this section we establish excess risk bounds for the weighted classification (13) and for the matrix regression (12). Our learning reduction approach successfully bridges these two tasks based on Vapnik's maxim and achieves theoretical guarantee for both problems.

The following assumption quantifies the representation capability of candidate classifiers $\mathcal{F}(r, s_1, s_2)$. For simplicity of notation, we assume $d_1 = d_2 = d$ and $\|\boldsymbol{X}\|_F \leq 1$ with probability 1.

Assumption 1 (Approximation error). Let $f_{\text{bayes},\pi}(\boldsymbol{X}) = \text{sign}(p(\boldsymbol{X}) - \pi)$ be the Bayes classifier corresponding to the π-level set. Assume 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., $R_{\ell,\pi}(f_n^*) - R_{\ell,\pi}(f_{\text{bayes},\pi}) \leq a_n$ for some sequence $a_n \to 0$ as $n, d \to \infty$. Here $R_{\ell,\pi}(f) = \mathbb{E}\left[w_\pi(y) \ell(y f(\boldsymbol{X}))\right]$ denotes the population surrogate risk as the counterpart of the empirical surrogate risk in (13). Let $J_n = \|f_n^*\|_F^2$, and we allow J_n to grow with n.

We now provide the accuracy guarantee for the level set estimation (13). We consider the high dimensional regime as both the sample size n and matrix dimension d grow, while treating (r, s_1, s_2) as fixed constants. The result demonstrates the statistical consistency of our classifier even when the matrix dimension far exceeds the sample size n.

Theorem 3 (Accuracy for matrix classification)**.** *Fix a level $\pi \in (0,1)$. Consider the problem of π-level set estimation for a (π,α)-locally regular function $p(\boldsymbol{X})$ with $\alpha \in [0,1]$. Suppose Assumption 1 holds, and let $\hat{f}_\pi$ be the level set estimate in (13) with penalty parameter $\lambda \asymp \left(\frac{r(s_1+s_2)\log d}{nJ_n}\right)^{1/(2-\alpha)}$ Then, with high probability, the classification excess risk is bounded by*

$$R_\pi(\hat{f}_\pi) - R_\pi(f_{\text{bayes},\pi}) \lesssim \max\left\{ \left(\frac{r(s_1+s_2)\log d}{n}\right)^{1/(2-\alpha)},\ a_n \right\}, \tag{18}$$

where $R_\pi(f) = \mathbb{E}[w_\pi(y)\mathbb{1}(y \neq \text{sign} f(\boldsymbol{X}))]$ denotes the weighted classification risk. Notice that the usual classification corresponds to $\pi = 1/2$.

Theorem 3 reveals the weak dependence on matrix dimension of our estimates. Consider the case when the statistical error (first term) dominates the approximation error (second term). Then, the bound (18) 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 sample complexity shows the advantage of proposed low-rank two-way sparse structural models. Furthermore, we find that classification (18) reaches a fast rate $1/n$ when $\alpha = 1$, and in general the risk has rate no slower than $1/\sqrt{n}$. This observation extends the asymptotic for usual vector-based classification (Tsybakov et al., 2004; Shen and Wang, 2006; Audibert et al., 2007).

We now reach the main results in this section for our nonparametric matrix regression.

Theorem 4 (Accuracy for nonparametric matrix regression)**.** *Let the regression function $p(\boldsymbol{X})$ be α-globally regular with $\alpha \in [0,1]$. Consider the same setup as in Theorem 3. Furthermore, assume Assumption 1 holds for all $\pi \in \Pi \setminus \mathcal{N}$. Then with high probability, the estimate (12) is bounded by*

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

Theorem 4 demonstrates the high dimensional convergence of our nonparametric matrix regression. Our results reveal three sources of errors: the statistical error in classification due to finite sample size, the approximation error due to the capability of candidate classifiers $\mathcal{F}(r, s_1, s_2)$, and an additional approximation error due to learning reduction from classification to regression. The resolution parameter H controls the bias-variance tradeoff.

Corollary 1 (High-dimensional consistency)**.** *Consider the same set-up as in Theorem 4. 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, with high probability,*

$$\mathbb{E}|\hat{p}(\boldsymbol{X}) - p(\boldsymbol{X})| \lesssim \left(\frac{r(s_1+s_2)\log d}{n}\right)^{\min(1/2,\alpha)/(2-\alpha)} \quad \textit{as } d, n \to \infty \textit{ while } d = o(e^n). \tag{19}$$

We apply the convergence rate in Theorem 4 to two specific learning examples.

Example 3 (Piece-wise constant model)**.** Consider piece-wise constant probability function $p(\boldsymbol{X}) = \sum_{t=1}^{T} c_t \mathbb{1}(\langle \boldsymbol{X}, \boldsymbol{B}_t \rangle = 0)$ with nonequal $c_1 < c_2 < \cdots < c_T$. In particular, $T = 1, \boldsymbol{B}_1 = \boldsymbol{0}$ reduces the constant model $p(\boldsymbol{X}) \equiv c$. We have $\alpha = 1$ in both cases. Theorem 4 gives an convergence rate $\mathcal{O}(n^{-1/2})$ by setting $H \asymp n^{-1/2}$. This rate achieves minimax optimality as in parametric models.

Example 4 (Single index model)**.** Consider the parametric model $p = g \circ f$ as in Example 1. For common links such as $g(t) = t$ and logistic $g(t) = (1 + \exp(t))^{-1}$, we have $\alpha = 1/2$. Choosing $H \asymp n^{1/3}$ yields the convergence rate $\mathcal{O}(n^{-1/3})$.

We conclude this section by comparing regression and classification. The regression bound (19) reaches the fastest rate $1/\sqrt{n}$ when $\alpha = 1$. This error rate is generally slower than the corresponding classification rate in (18). The fact confirms our earlier premise that classification is an easier problem than regression. Our level set approach successfully bridges theses two tasks and achieves theoretical guarantee for both problems. We expect this general principle may also benefit other settings beyond matrix learning tasks.

6 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.

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

We examine the prediction accuracy of our proposed method using multiple index models. The multiple index model extends the single index model (see Example 1) by allowing a multi-variate latent response $(z_1, z_2) = (\langle \boldsymbol{B}_1, \boldsymbol{X} \rangle, \langle \boldsymbol{B}_2, \boldsymbol{X} \rangle)$. We simulate random matrices $\boldsymbol{X} \in \mathbb{R}^{d \times d}$ with i.i.d. Uniform[0,1] entries, and draw $\boldsymbol{B}_1, \boldsymbol{B}_2$ from $\mathcal{F}(r, s, s)$. The response label is simulated from $y \sim$ Ber$(p(\boldsymbol{X}))$, where the regression function $p(\boldsymbol{X})$ is generated from the following chain scheme,

$$\boldsymbol{X} \to (z_1, z_2) \to (\bar{z}_1, \bar{z}_2) \to p(\boldsymbol{X}) = \begin{cases} g(\bar{z}_1), & \text{if } \bar{z}_1 > 0, \\ g(\bar{z}_2), & \text{otherwise.} \end{cases}$$

We set Φ_i = empirical cumulative distribution function (CDF) of z_i for $i = 1, 2$; Φ = CDF of standard normal; $g(z) = (1 + \exp(z))^{-1}$; matrix dimension $d = 20, 30, \ldots, 60$; and training sample size $n = 100, 150, \ldots, 400$. The construction of p amounts to a high nonlinearity from $\boldsymbol{X}$ to $p(\boldsymbol{X})$. Unlike parametric methods, the functional form of p is set unknown to the algorithm.

The first experiment assesses the impact of sample size to classification. We fix the matrix dimension $d = 30$ and let the sample size n increase. Figure 2a plots the resulting density of $p(\boldsymbol{X})$ induced from the nonlinear function p and distribution of $\boldsymbol{X}$. The classification error is measured by the excess risk $R(\hat{S}_{\text{bayes}}) - R(S_{\text{bayes}})$ evaluated on test data. As seen from Figure 2b, the classification error decays polynomially in sample size, which is consistent with our theoretical results. We find that a higher rank or a higher number of active nodes 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 2c plots the classification error versus the dimension d for each of three model settings $(r, s) = (2, 2), (2, 3)$ and $(4, 4)$. We find that the error increases slowly with matrix dimension, and the growth appears well controlled by the log rate. Note that, as the dimension increases, the number of active nodes remain unchanged, but the combinatoric complexity increases in the model space. The error seems unavoidable because of the price one needs to pay for not knowing the positions of the s active nodes. The ability to effectively control massive noisy features highlights the benefit of our method in high dimensions.

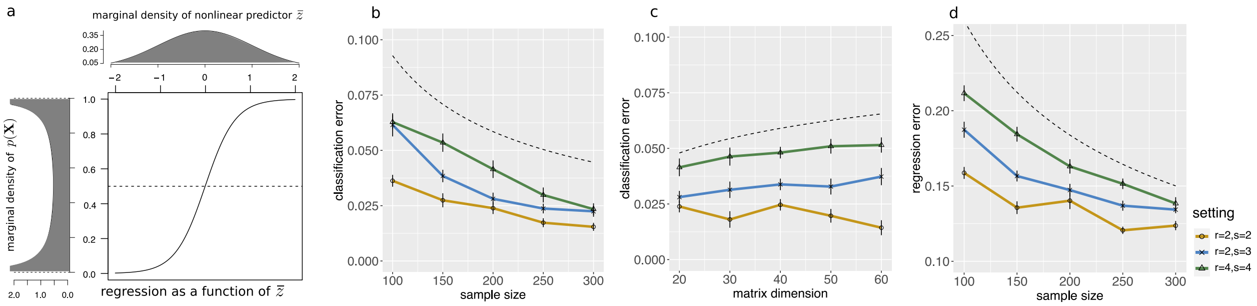

Figure 2: Finite sample accuracy of matrix classification and regression. (a) simulation setup. (b) classification error with sample size when $d = 30$. (c) classification error with matrix dimension when $n = 200$. (d) regression error with sample size. The dashed line in panels (b)-(d) represent theoretical rates $\mathcal{O}(n^{-2/3})$, $\mathcal{O}(\log d)$, and $\mathcal{O}(n^{-1/2})$, respectively. The reported statistics are averaged across 30 simulation replicates, with standard error given in the error bar.

The third experiment investigates similar aspects as before but to the task of matrix regression. We set the smoothing parameter $H = 20$ and aggregated the multiple level-sets as in our proposal (12). Figure 2d shows the regression error measured by L-1 risk, $\mathbb{E}|\hat{p}(\boldsymbol{X}) - p(\boldsymbol{X})|$, evaluated on test data. Again, we see 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. This scenario of high feature dimension low sample size is prevalent in brain network analysis. Nevertheless, our nonparametric method consistently learns the function p from limited data without the need to specify a priori functional form.

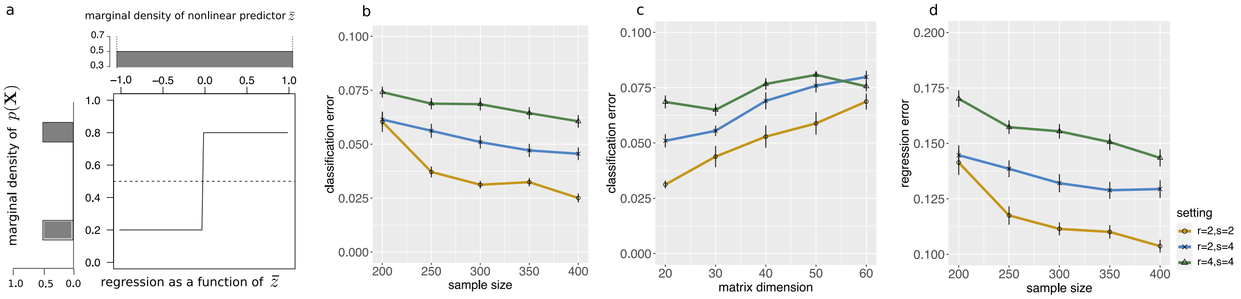

Figure 3: Finite sample accuracy under a different setting. (a) simulation setup. (b) classification error with sample size when $d = 30$. (c) classification error with matrix dimension when $n = 200$. (d) regression error with sample size.

The fourth experiment investigates the impact of target function to the prediction accuracy. We have shown that the probabilistic behavior of the random variable $p(\boldsymbol{X})$ plays a key role in our learning theory (see Section 3). Here we assess the empirical performance by repeating all the above experiments using a variety of $p(\boldsymbol{X})$. For space consideration, only one representative example is presented in the main texts, and the rest in the appendix. Figure 3a shows a model setting that falls on the other end of the spectrum. The random variable $p(\boldsymbol{X})$ concentrates at two mass points $\pi = 0.2$ and 0.8. This makes the π-level set estimation challenging around $\pi = 0.2$ and 0.8, because of the nonidentifiability in the weighted classification. Interestingly, we find that our method maintains good performance on classification at $\pi = 0.5$ (Figures 3b-c) and the overall regression (Figure 3d). One possible reason of this robustness is that we aggregate in total $(H-1)$ classifiers from $\Pi = \{\frac{1}{H}, \frac{2}{H}, \ldots, \frac{H-1}{H}\}$, each of which incurs at most $1/H$ error to the function estimation. Therefore, the estimation is robust against off-target level sets, as long as the majority are accurate.

6.2 Comparison with other methods

Next, we compare our method with several popular alternative methods:

- Unstructured regular lasso (**Lasso**). We vectorize the network predictor into a high-dimensional feature and then use elastic net (Friedman et al., 2010) with logistic loss to fit the vector-valued predictors.

- Parametric regression for network predictors with group lasso (**LogisticM**, Relión et al. (2019)). The original proposal is designated for network classification, and we adopt the fitted value from logistic loss as the probability estimation.

- Convolutional Neural Network (**CNN**) with two hidden layers implemented in Keras (Chollet et al., 2015). We apply 64 filters with 3 × 3 convolutional kernels to the matrix-valued predictor, followed by a pooling layer with size 5 × 5. The resulting features (neurons) are fed to a fully connected layer of neural network with ReLU activation.

- **Non**parametric **MA**trix **R**egression (**NonMAR**). This is our method that uses level set approaches to estimate the regression function for matrix-valued predictors.

We choose a range of representative methods and investigate the benefit of each approach. The **Lasso** serves as a baseline to assess the gain of matrix-valued predictors over vector-valued predictors. The methods **CNN** and **NonMAR** are nonparametric approaches and **LogisticM** is a parametric solution for matrix based prediction.

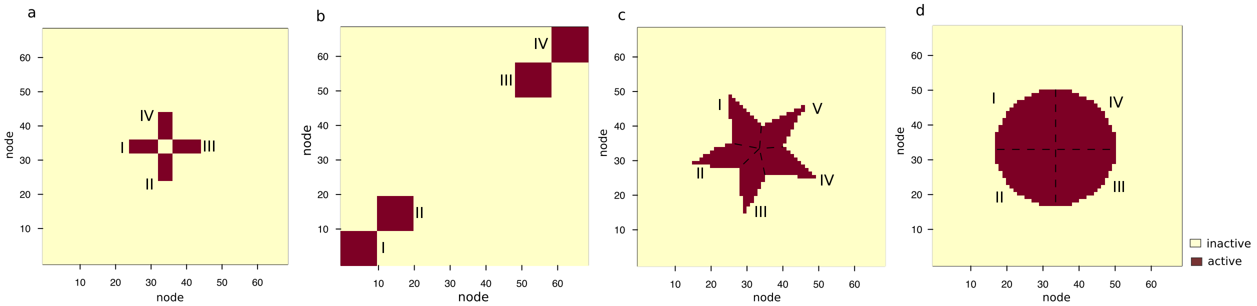

Figure 4: Four active pattern in simulations. The active region is divided into four or five subregions (denoted I, II, ...), each of which has its own edge connectivity signal $g_{pq}(\pi)$.

For fair comparison, we adopt similar simulation setup as in Relión et al. (2019), except that we add more challenging network patterns in order to assess model misspecification. We simulate the data $(\boldsymbol{X}_i, y_i)_{i\in[n]}$ from latent variable model $(\boldsymbol{X}, y)|\pi$ based on the following scheme,

$$\pi \sim_{\text{i.i.d.}} \text{Uniform}[0,1] \xrightarrow{\text{conditional on } \pi} \begin{cases} y \sim \text{Ber}(\pi),\ \ y \perp \boldsymbol{X}|\pi, \\ \boldsymbol{X} = [\![\boldsymbol{X}_{pq}]\!],\ \text{ where } \boldsymbol{X}_{pq} \sim_{\text{i.i.d.}} \mathcal{N}(g_{pq}(\pi)\mathbb{1}(\text{edge } (p,q) \text{ is active}), \sigma^2). \end{cases}$$

The edge connectivity signal, $g_{pq}(\pi)$, varies depending on the response probability π and location of $(p,q) \in [d]^2$. Figure 4 illustrates the active pattern which specifies the locations of active edges. The active region is further divided into several subregions, each of which has its own signal function $g_{pq}(\cdot)\colon [0,1] \to \mathbb{R}$. The function form of $g_{pq}(\cdot)$ is randomly drawn from a pre-specified library consisting of common functions such as $g(z) = \log(5z+1), 3\tan(z), 6z^2, \ldots$. We set $d = 68$, a training size $n = 160$, and a test size 80.

Our simulation reflects the challenging heterogeneity commonly arisen in brain network analysis. The sample consists of a mixture of individual groups with varying disease propensity, and the network patterns vary from one group to another. Active brain regions are supported on a submatrix with typically unknown rank. In the noiseless case, the cross and block patterns are low-rank ($r = 3$ and 5, respectively), whereas the star and circle patterns are nearly full-rank (numerical rank $r \approx 30$ on the supported submatrix). In our simulation with noisy observation, we select the rank and sparsity parameters (r, s) by 5-fold cross validation within training data. The hyperparameters for the other three methods are selected by either default setting (**LogisticM**) or cross validation (**Lasso**, **CNN**, **NonMAR**). We provide each algorithm the labeled networks as inputs after randomly permuting the node indices in the network. Because the software for **LogisticM** supports symmetric matrices only, we provide the algorithm $\frac{1}{2}(\boldsymbol{X} + \boldsymbol{X}^T)$.

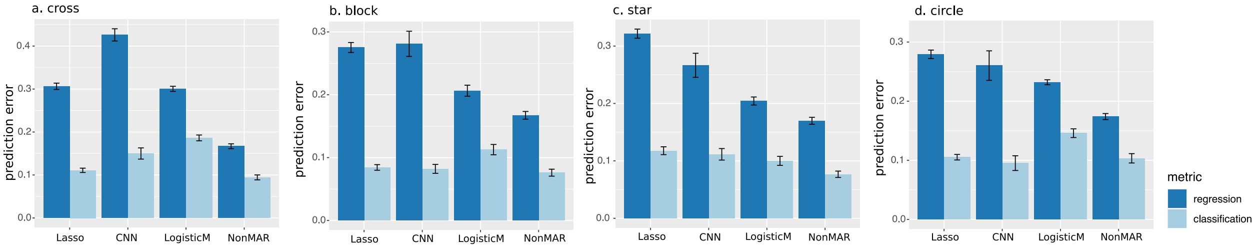

Figure 5: Performance comparison between various methods under four different active patterns.

Figure 5 compares the out-of-sample prediction between different methods. We focus on the regression problem and assess the performance using test data. We find that **NonMAR** consistently outperforms others, and the reduction in error is substantial. For example, the relative reduction using **NonMAR** over the next best approach, **LogisticM**, is over 20% for patterns a and d, and over 15% for patterns b and c. The results show the benefit of our nonparametric approaches 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 row/column indices. Although local similarity is an appropriate model for common imaging analysis, the row/column indices are meaningless for networks. Indeed, adjacency matrices differ by row/column permutation represent the same network, and methods that are index-invariant (**LogisticM** and **NonMAR**) show better performance. Our simulations cover a reasonably broad range of network models with rich structure. 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 pattern where the **CNN** has a lower classification 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 **NonMAR** achieves stable performance in spite of its simplicity. Interestingly, we find that the benefit of our method is more substantial in regression than in classification. One possible reason is that classification is an easier problem and less sensitive to various approaches. The result also suggests that the main reason of our method's superior regression performance may be attributable to level set aggregations.

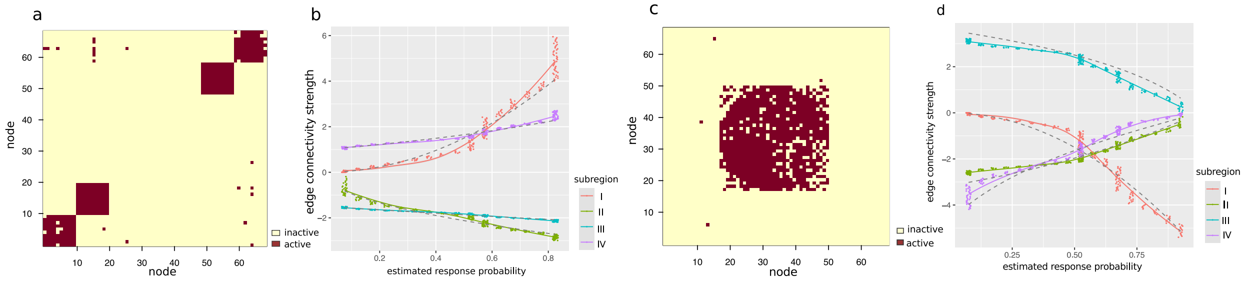

Figure 6: Example outputs returned by **NonMAR**. Panels (a) and (c) plot the top edges selected by our method. Panels (b) and (d) are scatter plots of the edge connectivity strength (averaged by subregion) versus the estimated response probability. The ground truth function is depicted in dashed curve.

We provide illustrate examples to show the outputs returned by **NonMAR**. Figures 6a and c plot the top edges selected by **NonMAR** based on the moving averages of feature weights $(\hat{\boldsymbol{B}}_\pi)_{\pi\in[0,1]}$ with a window size $\Delta\pi = 0.2$. The selected region agrees well with the ground truth (Figures 5a and c). We also investigate the relationship between edge connectivity for individual i and the estimated response probability $\hat{\pi}_i$. The trajectory of the edge connectivity accurately resembles the ground truth function in each subregion. The results demonstrate that our method is able to recover the right "sorting" of individuals with respect to the response probability on a continuous spectrum. The successful recovery of complicated unknown functions makes our method **NonMAR** appealing in applications.

7 Application to human brain connectome data

We apply our method to brain network data from Human Connectome Project (HCP). We analyze the Variable Short Penn Line Orientation Test (VSPLOT) score which measures the individual's visuospatial processing ability. We preprocess the data as in Wang et al. (2019), and analyze $n = 212$ individuals whose VSPLOT scores are either high ($y = 1$) or low ($y = -1$). Each individual's brain network is represented by a 68-by-68 binary adjacency matrix $\boldsymbol{X} \in \{0,1\}^{68\times 68}$, with the entries encoding the presence or absence of fiber connections between the 68 brain nodes. We adjust age and gender as additional covariates in the prediction, and use a random 60-20-20 split of the data for training, validation, and testing.

We compare our performance to other methods using the same procedure as in the previous section. Figure 7a shows that our method achieves high regression accuracy, measured by area under receiver operating characteristic (AUC). As common in high-dimensional settings, we observe that models with optimal cross-validation accuracy tend to include many noise variables. A useful heuristic is the so-called "one-standard-error rule" (Hastie et al., 2015), in which one selects the most parsimonious model with cross-validation accuracy within one standard error of the best. We apply this rule and report the results as **NonMAR-p**. It is remarkable to see that **NonMAR-p** results in 12% reduction a

Method	AUC	% of Active Nodes
NonMAR-p	**0.73 (0.03)**	88.2
NonMAR	**0.77 (0.04)**	97.3
LogisticM	0.72 (0.02)	100.0
Lasso	0.68 (0.01)	89.7
CNN	0.67 (0.03)	-

Note: CNN dost not report summary statistics for node selection. Numbers in parentheses are standard errors over 5-fold cross validations.

b

Rank	Node	Node	*p*-value*
1	r-inferiortemp**	r-middletemp	0.01
2	r-pars	r-supra	3e-5
3	r-posterior	l-precentral	0.01
4	l-caudal	l-isthmus	2e-5

* calculated from two sample test based on two label groups.
** Node names are shown in abbreviations, with "r" and "l" indicating the right and left hemisphere, respectively.

Figure 7: HCP analysis results based on our method **NonMAR**. (a) Comparison of prediction accuracy. (b) Top edges selected by our method **NonMAR-p**.

of active nodes but still achieves excellent accuracy (AUC = 0.73).

Figure 7b lists the top brain edges identified by our method. Edges are ranked by their maximal values in the feature weights $(\hat{\boldsymbol{B}}_\pi)_{\pi\in[0,1]}$ via moving averaging. We find that the top edges involve connections between frontal and occipital regions in the right hemisphere (Figure 8a). This seems consistent with recent evidence of dysfunction in right posterior regions for deficits in visuospatial processing (Wang et al., 2019). We also find nonlinear relationship between edge connection strength and response probability. In Figure 8b, the connection (r-parstriangularis, r-supramarginal) grows slowly when π is low but fast when π is high. In contrary, the connection (r-posteriorcingulate, r-precentral) grows fast initially and then reaches a plateau as π increaases. The detected pattern reveals the heterogeneous changes in brain connectivities with respect to visuospatial processing ability.

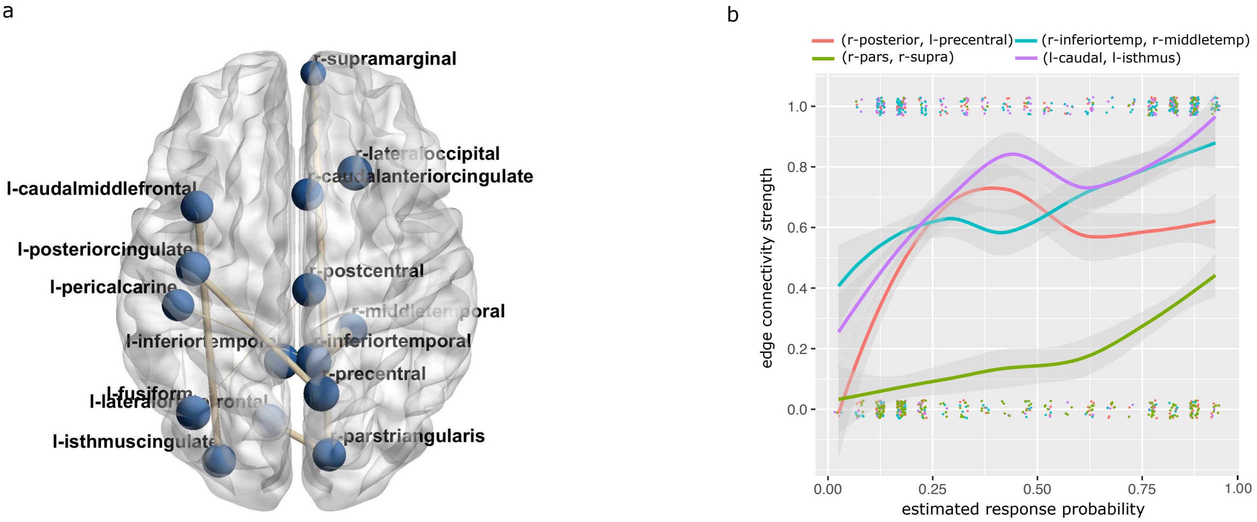

Figure 8: HCP analysis results. (a) Top edges overlaid on brain template. (b) Edge connectivity strength versus estimated response probability. Colored curves represent the moving averages of connectivity strengths, gray bands represent one standard error, and jitter points represent the raw connectivity values (0 or 1).

8 Conclusion

We have developed the learning framework for the relationship between a binary label response and a high-dimensional matrix-valued predictor. Our method respects the matrix structure of the predictors and provide interpretable prediction via a nonparametric approach. The theoretical and numerical results demonstrate the competitive performance of our method. 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 such as matrix completion and denoising problem warrants future research.

References

Alquier, P. and G. Biau (2013). Sparse single-index model. *Journal of Machine Learning Research 14*(Jan), 243–280.

Audibert, J.-Y., A. B. Tsybakov, et al. (2007). Fast learning rates for plug-in classifiers. *The Annals of statistics 35*(2), 608–633.

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.

Fan, J., W. Gong, and Z. Zhu (2019). Generalized high-dimensional trace regression via nuclear norm regularization. *Journal of Econometrics 212*(1), 177–202.

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

Ganti, R., N. Rao, L. Balzano, R. Willett, and R. Nowak (2017). On learning high dimensional structured single index models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1898–1904. AAAI Press.

Guha, S. and A. Rodriguez (2020). Bayesian regression with undirected network predictors with an application to brain connectome data. *Journal of the American Statistical Association*, 1–34.

Hastie, T., R. Tibshirani, and M. Wainwright (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

Hu, W., W. Shen, H. Zhou, and D. Kong (2020). Matrix linear discriminant analysis. *Technometrics 62*(2), 196–205.

Parikh, N. and S. Boyd (2014). Proximal algorithms. *Foundations and Trends in Optimization 1*(3), 127–239.

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

Shen, X. and L. Wang (2006). Discussion of 2004 IMS Medallion Lecture: Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics 34*, 2677–2680.

Singh, A., C. Scott, R. Nowak, et al. (2009). Adaptive Hausdorff estimation of density level sets. *The Annals of Statistics 37*(5B), 2760–2782.

Tsybakov, A. B. et al. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics 32*(1), 135–166.

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

Wang, L., Z. Zhang, D. Dunson, et al. (2019). Common and individual structure of brain networks. *The Annals of Applied Statistics 13*(1), 85–112.

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

Yang, D., Z. Ma, and A. Buja (2016). Rate optimal denoising of simultaneously sparse and low rank matrices. *The Journal of Machine Learning Research 17*(1), 3163–3189.

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