

# Nonparametric learning with matrix-valued predictors in high dimensions

Hao Helen Zhang

Chanwoo Lee<sup>1</sup>, Lexin Li<sup>2</sup>, Helen Zhang<sup>3</sup>, and Miaoyan Wang<sup>1</sup>

<sup>1</sup>University of Wisconsin-Madison <sup>2</sup>University of California-Berkeley <sup>3</sup>University of Arizona

resize figure.  
Avoid stretching.  
try 2-by-2 layout



## Problems & Existing methods

**Problems:** Let  $\{(\mathbf{X}_i, y_i) \in \mathbb{R}^{d_1 \times d_2} \times \{-1, 1\} : i = 1, \dots, n\}$  denote an i.i.d. sample from unknown distribution  $\mathcal{X} \times \mathcal{Y}$ .

- Classification: How to efficiently classify high-dimensional matrices with limited sample size:

$n \ll d_1 d_2$  = dimension of feature space?

- Regression: How to robustly predict the label probability when little is known to function form of  $p(\mathbf{X})$ :

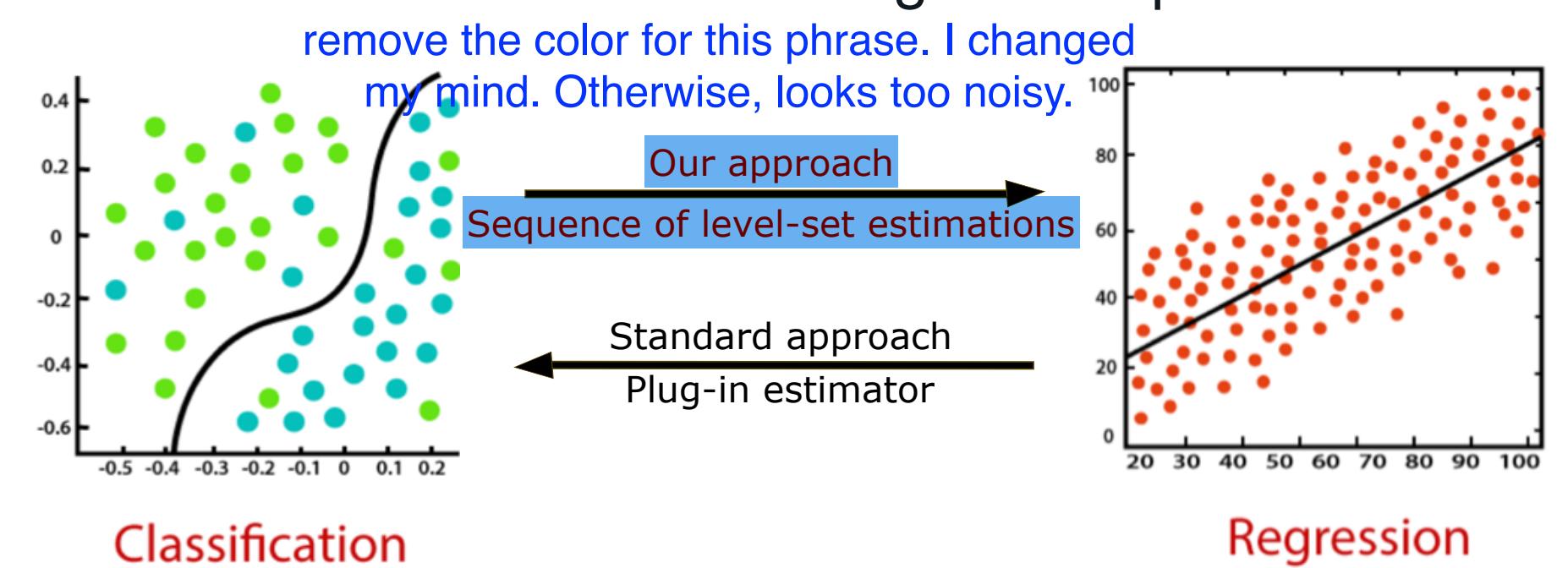
about  $p(\mathbf{X}) \stackrel{\text{def}}{=} \mathbb{P}(y=1|\mathbf{X})$ ?

### Existing methods :

- Classification: Decision tree, nearest neighbor, neural network, and support vector machine. However, most methods have focused on vector valued features. Be consistent. hyphen. either use hyphens for "matrix-valued", "vector-valued" or do or use no hyphen. Do not mix extra space? a nonparametric..

- Regression: Logistic regression and linear discriminant analysis. However, it is often difficult to justify the assumptions on the function form, especially when the feature space is high-dimensional.

**Goal:** We propose nonparametric learning approach with matrix-valued predictors. Unlike classical approaches, our approaches find classification rule first and address regression problem.



## Methods: Classification with matrix predictors

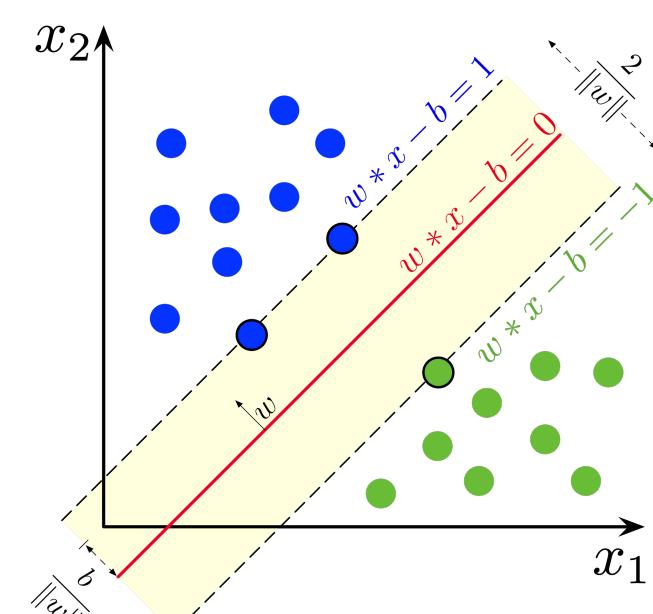
- Remove "Methods" to fit in one line → avoid orphan word in posters

- We develop a large-margin classifier for matrix predictors.

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n L(y_i f(\mathbf{X}_i)) + \lambda J(f), \quad (1)$$

- We set  $\mathcal{F} = \{f : f(\cdot) = \langle \mathbf{B}, \cdot \rangle \text{ where } \text{rank}(\mathbf{B}) \leq r, \|\mathbf{B}\|_F \leq C\}$ ,  $J(f) = \|\mathbf{B}\|_F^2$ , and we choose  $L(t)$  to be a large-margin loss, such as hinge loss, logistic loss, etc.

- We also develop nonlinear classifiers for matrix predictors using a new family of matrix-input kernels.



Large margin classifier for vector predictors  
(Picture source: Wiki).

## Methods: Regression function estimation with matrix predictors

- We propose a nonparametric functional estimation using a sequence of weighted classifiers from (1).

$$\hat{f}_\pi = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \omega_\pi(y_i) L(y_i f(\mathbf{X}_i)) + \lambda J(f),$$

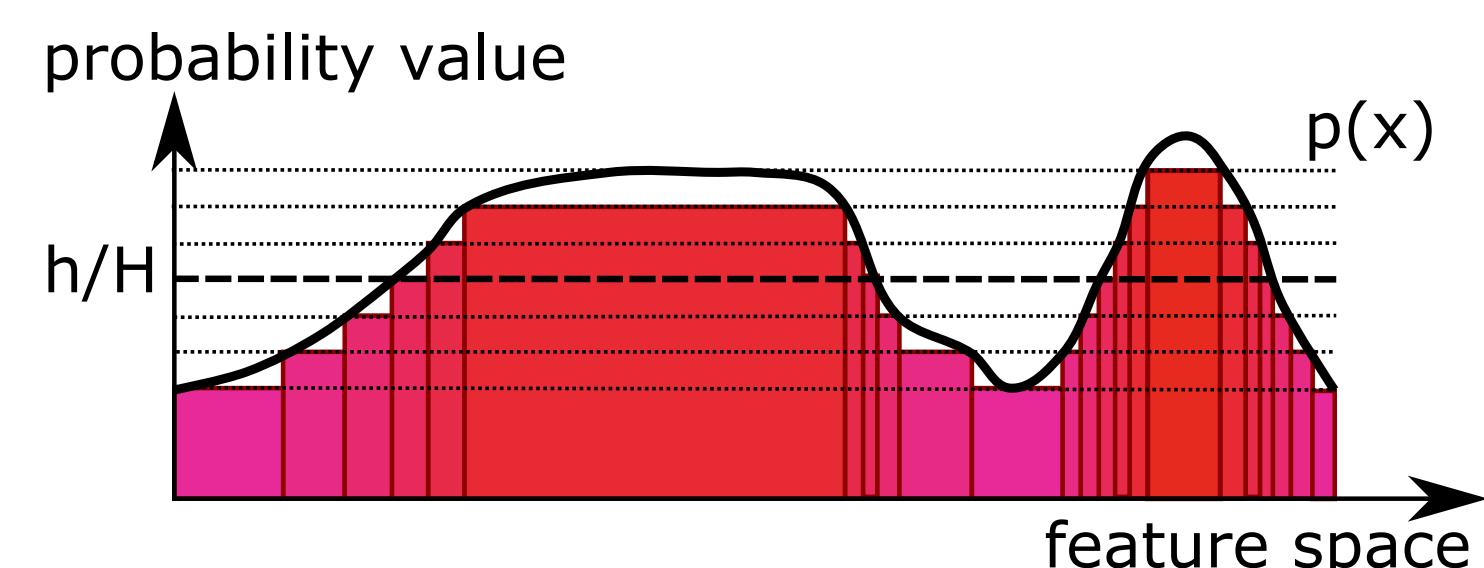
where  $\omega_\pi(y) = 1 - \pi$  if  $y = 1$  and  $\pi$  if  $y = -1$ .

- The main idea is to estimate  $p(\mathbf{X})$  through two steps of approximations:

$$p(\mathbf{X}) \stackrel{\text{Step 1}}{\approx} \frac{1}{H} \sum_{h \in [H]} \mathbb{1} \left\{ \mathbf{X} : p(\mathbf{X}) \leq \frac{h}{H} \right\}$$

$$\stackrel{\text{Step 2}}{\approx} \frac{1}{H} \sum_{h \in [H]} \mathbb{1} \left\{ \mathbf{X} : \underbrace{\text{sign} [\hat{f}_{\pi_h}(\mathbf{X})]}_{\text{Keep your original line `` where H is ...''}} = -1 \right\}.$$

- Step 1 is discretization of target function by level sets.



- Step 2 is to estimate decision region using a sequence of weighted classifiers.

$$\left\{ \mathbf{X} : \underbrace{\text{sign} [\hat{f}_\pi(\mathbf{X})]}_{\substack{\text{estimated decision region from classification}}} = -1 \right\} \xrightarrow{\text{in } p} \left\{ \mathbf{X} : \underbrace{\mathbb{P}(Y=1|\mathbf{X}) \leq \pi}_{\substack{\text{targeted level set}}} \right\},$$

- We provide accuracy guarantees for the above two steps by extending theories in [2] from vectors to high-dimensional matrix predictors.

## Algorithms

- We develop an alternating optimization to solve non-convex problem (1).

- We factor the coefficient matrix  $\mathbf{B} = \mathbf{U}\mathbf{V}^T$  where  $\mathbf{U} \in \mathbb{R}^{d_1 \times r}$  and  $\mathbf{V} \in \mathbb{R}^{d_2 \times r}$ .

either fit into one line, or add something to make it longer.

### Algorithm 1: Classification algorithm with matrix predictors

Input:  $(\mathbf{X}_1, y_1), \dots, (\mathbf{X}_n, y_n)$ , and prespecified rank  $r$

Initialize:  $(\mathbf{U}^{(0)}, \mathbf{V}^{(0)}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$

Do until converges

Is it pdf? looks blurry

Update  $\mathbf{U}$  fixing  $\mathbf{V}$ :

$$\mathbf{U} = \arg \min_{\mathbf{U}} \frac{1}{n} \sum_{i=1}^n (1 - \langle \mathbf{U}\mathbf{V}^T, \mathbf{X}_i \rangle)_+ + \lambda \|\mathbf{U}\mathbf{V}^T\|_F^2.$$

Update  $\mathbf{V}$  fixing  $\mathbf{U}$ :

$$\mathbf{V} = \arg \min_{\mathbf{V}} \frac{1}{n} \sum_{i=1}^n (1 - \langle \mathbf{U}\mathbf{V}^T, \mathbf{X}_i \rangle)_+ + \lambda \|\mathbf{U}\mathbf{V}^T\|_F^2.$$

Output:  $\hat{f}(\mathbf{X}) = \langle \mathbf{U}\mathbf{V}^T, \mathbf{X} \rangle$

Remove the section title "Reference"  
Just plain words  
Reference (left aligned, in bold):

You have chosen to capitalize the first word only. Be consistent.  
Fit the section title into one line?

## Application: Probability Matrix Estimation

- Our method leads itself well to nonparametric matrix estimation problems.

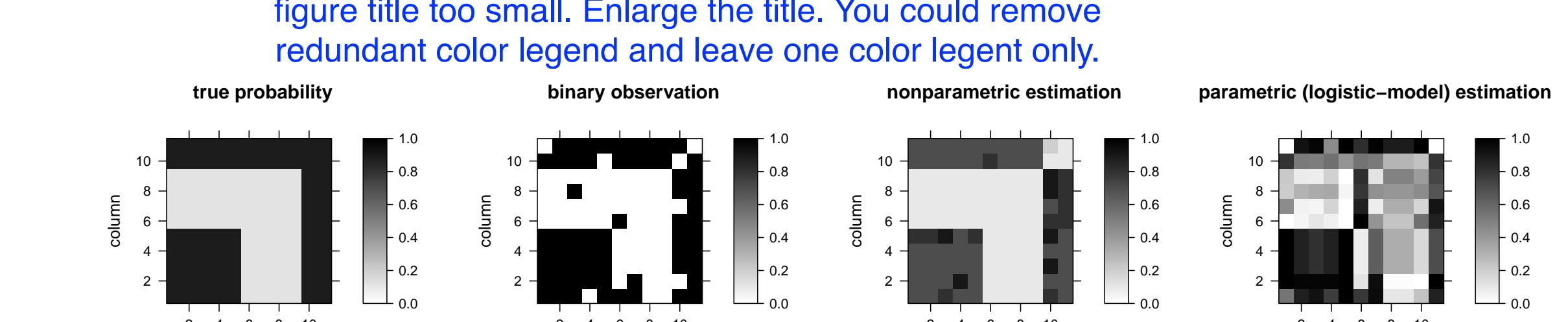
- Goal: Estimate the probability matrix  $\mathbf{P} = [p_{ij}] \in [0, 1]$  from binary observations  $\mathbf{Y} = [y_{ij}] \in \{0, 1\}$ , where  $y_{ij} \stackrel{\text{ind.}}{\sim} \text{Ber}(p_{ij})$  for  $(i, j) \in [d_1] \times [d_2]$ .

Latex: do you use colon or plain ?:  
• Training set:  $\{(\mathbf{X}_{ij}, y_{ij}) : (i, j) \in [d_1] \times [d_2]\}$  where

Use \entry{...} or [...]  $[X_{ij}]_{pq} = \begin{cases} 1 & \text{if } (p, q) = (i, j) \\ 0 & \text{otherwise} \end{cases}$

Choose one and be consistent with the format,  
is an indicator matrix with 1 in  $(i, j)$ -th position and 0's everywhere.  
should not be in math mode

- We apply our developed methods to estimate  $p_{ij} = \mathbb{P}(y_{ij} = 1 | \mathbf{X}_{ij})$ .



- Our nonparametric approach provides more robust matrix estimation than parametric approaches [1],[3]

## Theoretical results

is a set of i.i.d. Gaussian random matrices with bounded variance

**Theorem 1.** Assume that  $\{\mathbf{X}_i\}_{i=1}^n$  be set of i.i.d. Gaussian distribution with bounded variation. Then with high probability, remove "4".

$$\mathbb{P}[Y \neq \text{sign}(\hat{f}(\mathbf{X}))] - \mathbb{P}[Y \neq \text{sign}(f^*(\mathbf{X}))] \leq \frac{4C\sqrt{r(d_1 + d_2)}}{\sqrt{n}},$$

where  $f^*$  is the best predictor in  $\mathcal{F}$ .

**Theorem 2.** Let  $\hat{p}$  be an estimated probability function from our method. Under some assumptions, we have  
Depending on the space, you may consider

Under some assumptions → Under some assumptions on the function class  $\mathcal{F}$ , penalization parameter lambda, and smoothing parameter  $H$ ,  $\mathbb{E}\|\hat{p} - p\|_1 = \mathcal{O}\left(\left(\frac{\log(n/(d_1+d_2))}{n/(r(d_1+d_2))}\right)^{1/(2-\alpha \wedge 1)}\right)$ ,

If assumptions are not included in the poster, you would know them in your mind.

where  $\alpha$  is a regularity parameter determined by the true probability.  
probability function.

If  $\alpha > 1$  and  $d_1 = d_2 = d$ , we have

$$\mathbb{E}\|\hat{p} - p\|_1 = \mathcal{O}\left(\frac{\log(n/rd)}{(n/rd)}\right).$$

## References

- Chanwoo Lee and Miaoyan Wang. "Tensor denoising and completion based on ordinal observations". In: International conference on machine learning (2020).
- Junhui Wang, Xiaotong Shen, and Yufeng Liu. "Probability estimation for large-margin classifiers". In: Biometrika 95.1 (2008), pp. 149–167.
- Miaoyan Wang and Lexin Li. "Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality". In: The Journal of Machine Learning Research In press (2020).

Use APA citation format for conciseness

No quote "", no "In: ..."

Could also just abbreviation for publication venues, ICML, JMLR

reason: column 3 is longer than the other two columns.