

Nonparametric learning with matrix-valued predictors in high dimensions

Chanwoo Lee¹, Lexin Li², Helen Zhang³, and Miaoyan Wang¹

¹University of Wisconsin-Madison ²University of California-Berkeley ³University of Arizona



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:

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

1. revere order
 p(X) \stackrel{\text{def}}{=} P(...)
 2. non-bold font p
 3. "little is known about the function form of p(X)"
 many typos:
 square in F norm?
 space before "and"

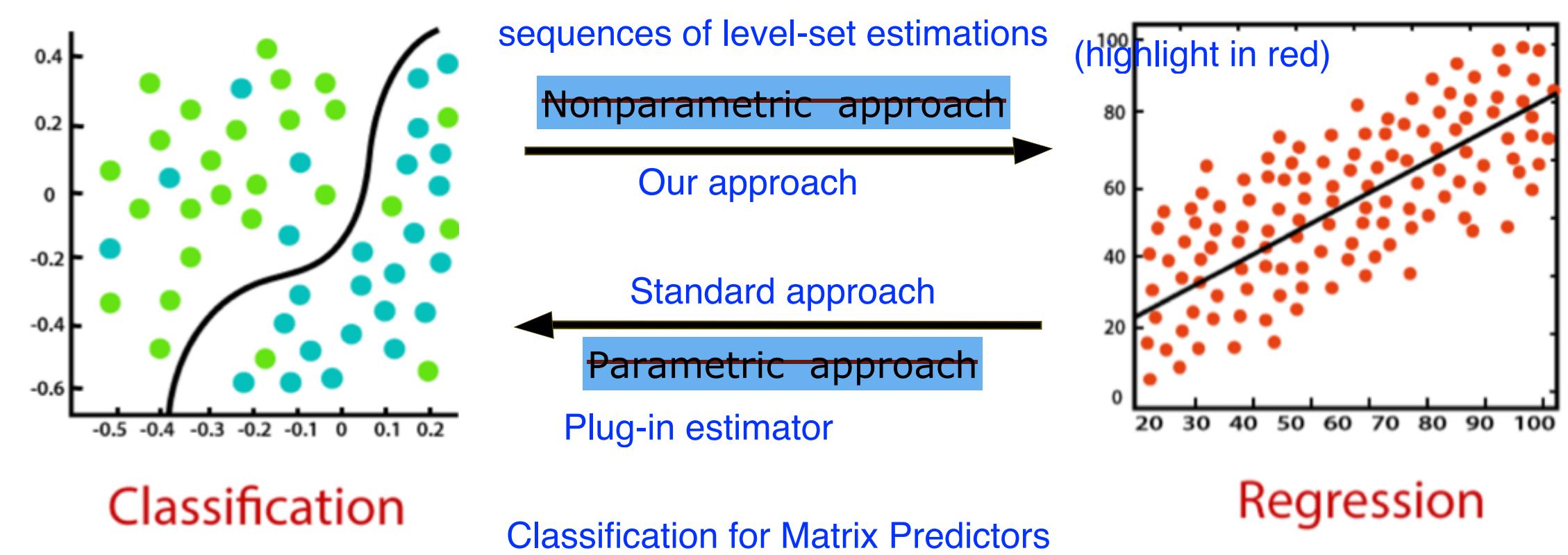
Existing methods :

- Classification: Decision tree, Nearest neighbor, Neural network, and Support vector machine. However, most of methods have focused on vector valued features.

non-capital

- Regression: Logistic regression and Linear discriminant analysis. However, it is often difficult to justify the assumptions made when features are matrices because of high-dimensionality.

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



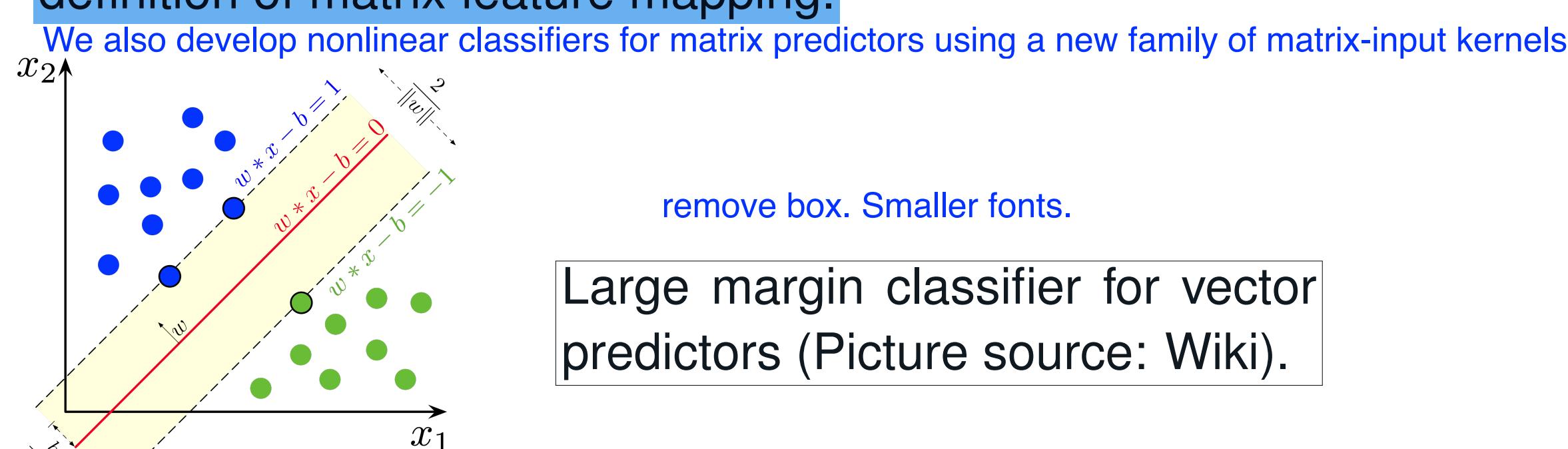
Methods: 1. Classification

- We develop a large-margin classifiers 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, $L(x) = (1 - x)_+$. we choose $L(t)$ to be a large-margin loss, such as hinge loss, logistic loss, etc.

- We can extend linear classifiers to nonlinear classifiers with a new definition of matrix feature mapping.



Methods: Regression Function Estimation with Matrix-valued Predictors

Methods: 2. Regression

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

- We consider weighted loss function from (1),

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

sum over i
highlight omega_pi in red
reserve the order for these two bullet points

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

- We estimate $p(\mathbf{X})$ through two steps of approximations:

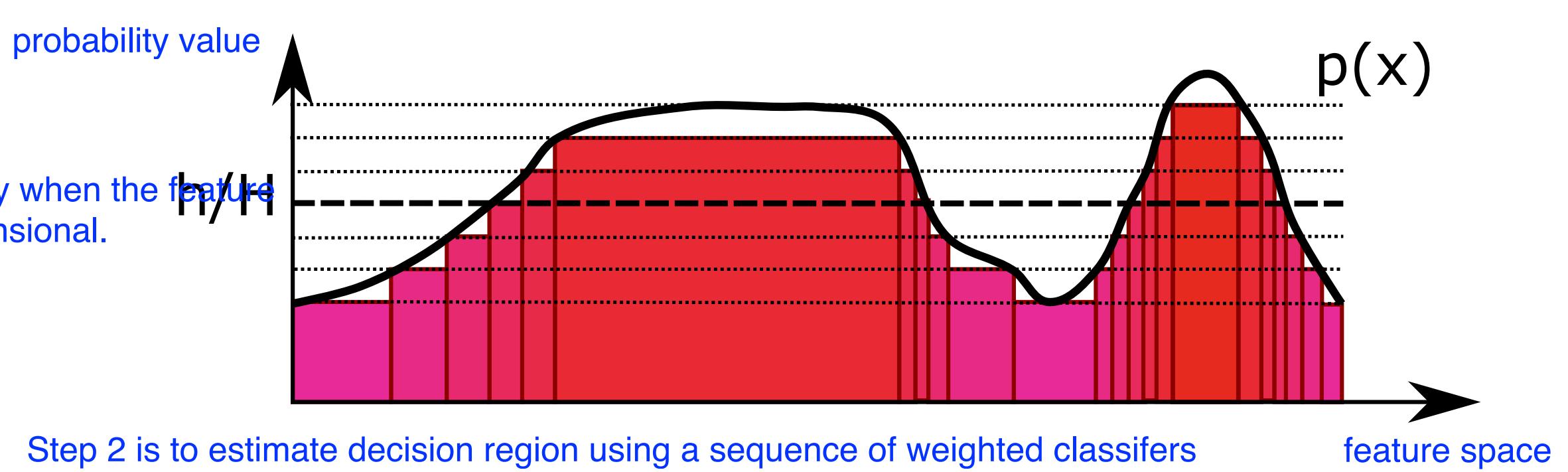
The main idea is to estimate $p(\mathbf{X})$

$$\begin{aligned} p(\mathbf{X}) &\approx \underset{\text{Step 1}}{\frac{1}{H} \sum_{h \in [H]} \mathbb{1} \left\{ \mathbf{X} : p(\mathbf{X}) \leq \frac{h}{H} \right\}} \\ &\underset{\text{Step 2}}{\approx} \frac{1}{H} \sum_{h \in [H]} \mathbb{1} \left\{ \mathbf{X} : \text{sign} \left[\hat{f}_{\frac{h}{H}}(\mathbf{X}) \right] = -1 \right\}, \end{aligned}$$

Step 1
Step 2
should not be in bold font

where $H \in \mathbb{N}_+ \rightarrow \infty$ is the smoothing parameter.

- 1st step is discretization of target function by level sets



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

- 2nd step is from

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

targeted level set

which is verified in [1].

Bullet point: We provide accuracy guarantees for the above two steps by extending theories in [1] from vectors to high-dimensional matrix predictors.

Application: Regression

Application: Probability Matrix Estimation

- Binary matrix probability estimation.

Our method lends itself well to nonparametric matrix estimation problems.

- Goal: estimate the probability matrix $\mathbf{P} \in [0, 1]^{d_1 \times d_2}$ from binary observations

$\mathbf{Y} = \{0, 1\}^{d_1 \times d_2}$ such that $y_{ij} = \text{Bernoulli}(p_{ij})$. $P = [p_{ij}] \in [0, 1]^{d_1 \times d_2}$

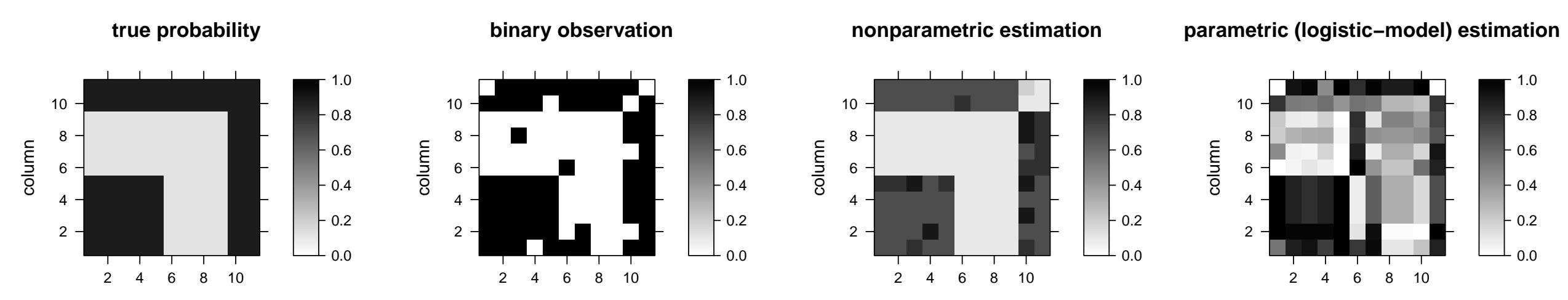
- Procedure: 1. Generate training set: $\{(\mathbf{X}_{ij}, y_{ij}) : (i, j) \in [d_1] \times [d_2]\}$ where

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

is an indicator matrix with 1 in the (i,j)-th position and 0's everywhere.

- 2. Estimate $P_{ij} = \mathbb{P}(y_{ij} = 1 | \mathbf{X}_{ij})$.

Bullet point: We apply our developed methods to estimate ...



Bullet point: Our nonparametric approach provides a more robust matrix estimation than parametric approaches [1] (cite your earlier ICML paper and my JMLR paper)

Algorithms

- We factor the coefficient matrix $\mathbf{B} = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U} \in \mathbb{R}^{d_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{d_2 \times r}$.
- We take alternating optimization approach to solve non-convex problem. (1). We develop an alternating optimization to solve non-convex problem (1)

Algorithm 1: Classification algorithm

Input: $(\mathbf{X}_1, y_1), \dots, (\mathbf{X}_n, y_n)$, and prespecified rank r Inconsistent notation: m or n?
 Initialize: $(\mathbf{U}^{(0)}, \mathbf{V}^{(0)}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$
 Do until converges
 Update \mathbf{U} fixing \mathbf{V} :
 $\mathbf{U} = \arg \min_{\mathbf{U}} \frac{1}{n} \sum_{i=1}^n (1 - \langle \mathbf{U}\mathbf{V}^T, \mathbf{X}_i \rangle)_+ + \lambda \|\mathbf{U}\mathbf{V}^T\|_F$.
 Update \mathbf{V} fixing \mathbf{U} :
 $\mathbf{V} = \arg \min_{\mathbf{V}} \frac{1}{n} \sum_{i=1}^n (1 - \langle \mathbf{U}\mathbf{V}^T, \mathbf{X}_i \rangle)_+ + \lambda \|\mathbf{U}\mathbf{V}^T\|_F$.
 Output: $\hat{f}(\mathbf{X}) = \langle \mathbf{U}\mathbf{V}^T, \mathbf{X} \rangle$

many typos:
 square in F norm?
 space before "and"
 Classification algorithm
 → Classification algorithm with matrix predictors

Theoretical results

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,

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

flip the order

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

Theorem 2. Denote \hat{p} as an estimated probability function from our method. Under some assumptions, we have

$$\mathbb{E}\|\hat{p} - p\|_1 = \mathcal{O} \left(\left(\frac{\log(n/r(d_1 + d_2))}{(n/r(d_1 + d_2))} \right)^{1/(2-\alpha)} \right),$$

a regularity parameter determined by the true probability

where α is a constant determined by true probability. If $\alpha > 1$ and $d_1 = d_2 = d$, we have

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

Highlight this only. No need to highlight earlier line.

References

- [1] Junhui Wang, Xiaotong Shen, and Yufeng Liu. "Probability estimation for large-margin classifiers". In: *Biometrika* 95.1 (2008), pp. 149–167.