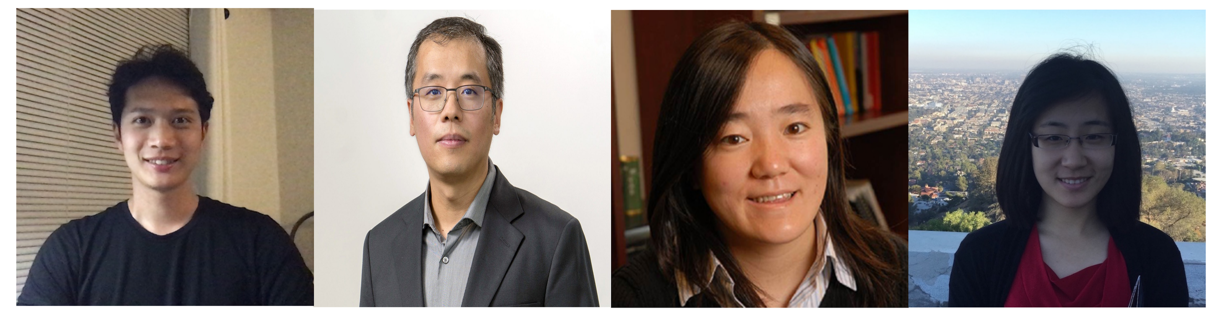
Nonparametric learning with matrix-valued predictors in high dimensions

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

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



"Problem" & Existing methods

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

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

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

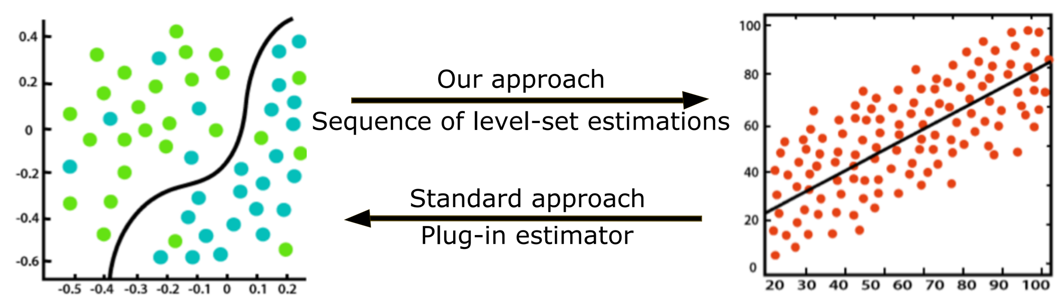
- Regression: How to robustly predict the label probability when little is known about the function form of $p(\boldsymbol{X})$:

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

Existing methods:

- Classification: Decision tree, nearest neighbor, neural network, and support vector machine. However, most methods have focused on vector-valued features.
- 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 a nonparametric learning approach with matrix-valued predictors. Unlike classical approaches, our approach uses classification rule to address regression problem.



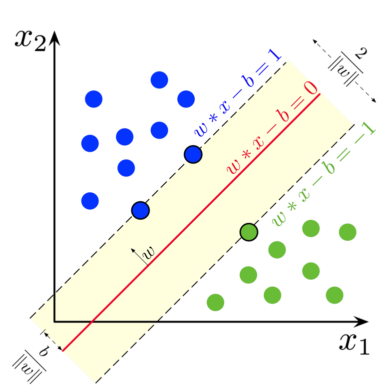
Classification Regression

Classification with matrix predictors

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

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

- We set $\mathcal{F} = \{f : f(\cdot) = \langle \boldsymbol{B}, \cdot\rangle$ where $\text{rank}(\boldsymbol{B}) \le r, \|\boldsymbol{B}\|_F \le C\}$, $J(f) = \|\boldsymbol{B}\|_F^2$, 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.



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

Regression with matrix predictors

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

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

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

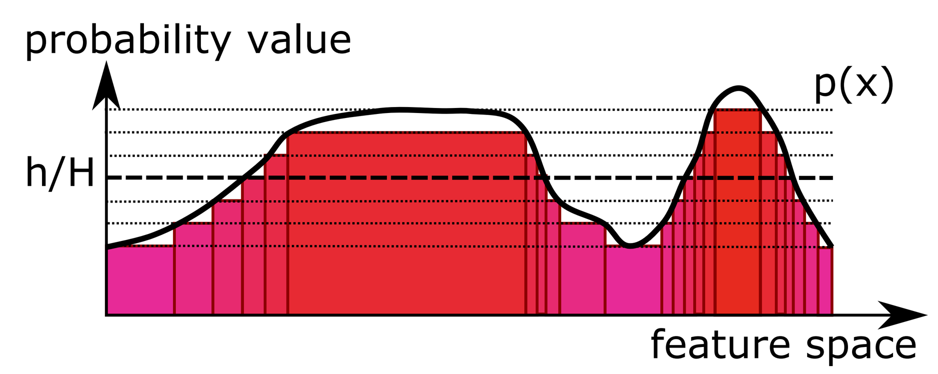
- The main idea is to estimate $p(\boldsymbol{X})$ through two steps of approximations:

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

$$\stackrel{\text{Step 2}}{\approx} \frac{1}{H}\sum_{h\in[H]} \mathbb{1}\left\{\boldsymbol{X}: \text{sign}\left[\hat{f}_{\frac{h}{H}}(\boldsymbol{X})\right] = -1\right\},$$

where H is a smooth parameter.

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



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

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

- 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 $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{V}^T$ where $(\boldsymbol{U},\boldsymbol{V}) \in \mathbb{R}^{d_1\times r}\times\mathbb{R}^{d_2\times r}$.

Algorithm 1: Classification algorithm with matrix predictors

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

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

Do until converges

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

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

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

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

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

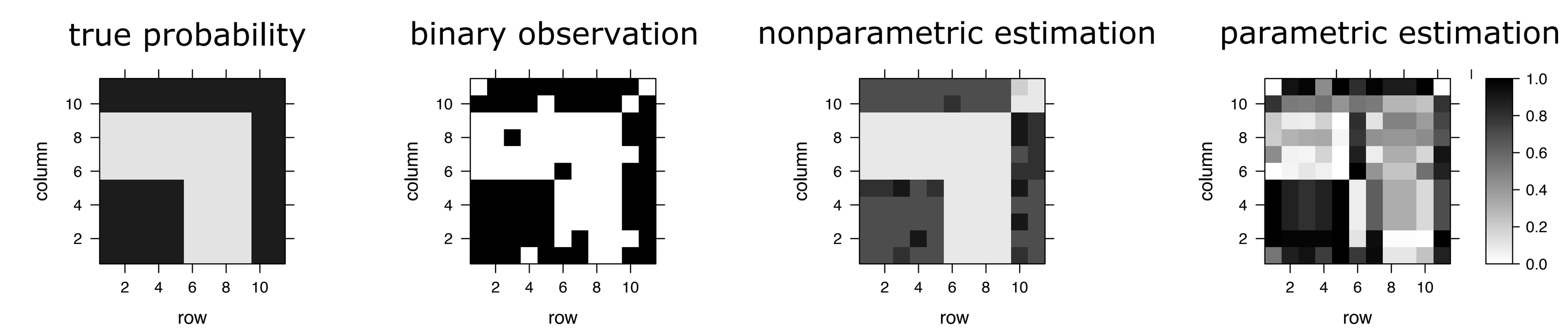
Application: Probability matrix estimation

- Our method leads itself well to probability matrix estimation problems.
- Goal: Estimate the probability matrix $\boldsymbol{P} = [\![p_{ij}]\!] \in [0,1]$ from binary observations $\boldsymbol{Y} = [\![y_{ij}]\!] \in \{0,1\}$, where $y_{ij} \stackrel{\text{ind.}}{\sim} \text{Ber}(p_{ij})$ for $(i,j)\in[d_1]\times[d_2]$.
- Training set: $\{(\boldsymbol{X}_{ij}, y_{ij}) : (i,j)\in[d_1]\times[d_2]\}$ where

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

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

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



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

Theoretical results

Theorem 1. *Assume that $\{\boldsymbol{X}_i\}_{i=1}^n$ is a set of i.i.d. Gaussian random matrices with bounded variance. Then with high probability,*

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

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

Theorem 2. *Let $\hat{p}: \mathbb{R}^{d_1\times d_2} \to [0,1]$ be the estimated probability function from our method. Under some assumptions on the function class $\mathcal{F}$, the penalty parameter λ, and the smoothing parameter H, we have*

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

where α is a regularity parameter determined by the true probability function. If $\alpha \ge 1$ 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

[1] Lee, C. and Wang, M. (2020). Tensor denoising and completion based on ordinal observations. *International conference on machine learning.*

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

[3] Wang, M. and Li, L. (2020). Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. *JMLR*, In press.