Nonparametric learning with matrix-valued predictors in high dimensions

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

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

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

Problems & Existing methods

an

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 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 to function form of $p(\boldsymbol{X})$:

about

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

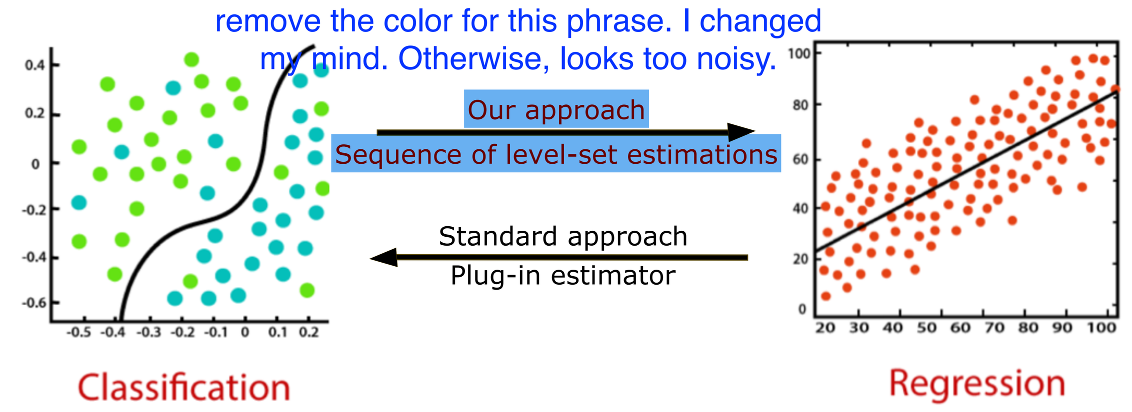
Existing methods :

- Classification: Decision tree, nearest neighbor, neural network, and support vector machine. However, most methods have focused on vector valued features. hyphen. Be consistent. either use hyphens for ``matrix-valued'', ``vector-valued''or do or use no hyphen. Do not mis
- 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.

extra space? a nonparametric ..

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

remove the color for this phrase. I changed my mind. Otherwise, looks too noisy.



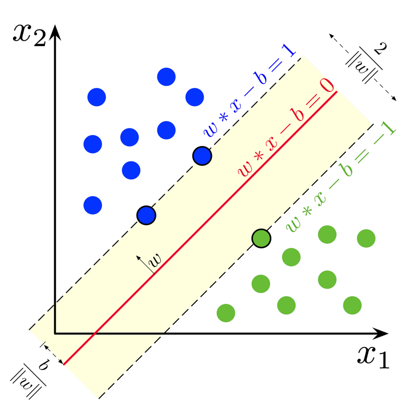
~~Methods:~~ Classification with matrix predictors

Remove ``Methods'' to fit in one line —> avoid orphan word in posters

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

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

- 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.



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), estimation method

highlight \omega_pi(y_i)

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

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

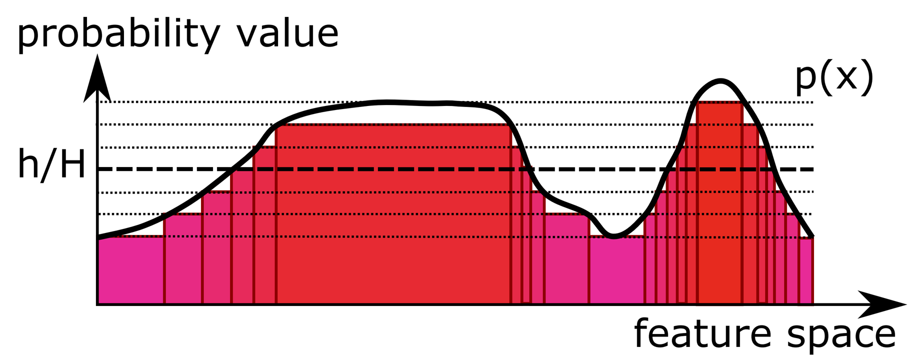
- The main idea is to estimate $p(\boldsymbol{X})$ through two steps of approximations:

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

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

Keep your original line `` where H is …''
Was not intended to delete the line. Editing in Preview was terrible..

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



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

decision region estimation reason: avoid orphan and widow in slide and posters. fine in paper

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

- 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 $\boldsymbol{U} \in \mathbb{R}^{d_1\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{d_2\times r}$.

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

Algorithm 1: Classification algorithm with matrix predictors

Input: $(\boldsymbol{X}_1, y_1), \cdots, (\boldsymbol{X}_n, y_n)$, 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 Is it pdf? looks blurry

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

not a section on its own.
Just plain words
Reference (left aligned, in bold):
[1]
[2]

Application: Probability Matrix Estimation

- Our method leads itself well to ~~nonparametric 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 $y_{ij} \overset{\text{ind.}}{\sim} \text{Ber}(p_{ij})$ for $(i,j) \in [d_1]\times[d_2]$. comma
- Training set: $\{(\boldsymbol{X}_{ij}, y_{ij}) : (i,j) \in [d_1]\times[d_2]\}$ where

Latex: do you use \colon or : Looks wired

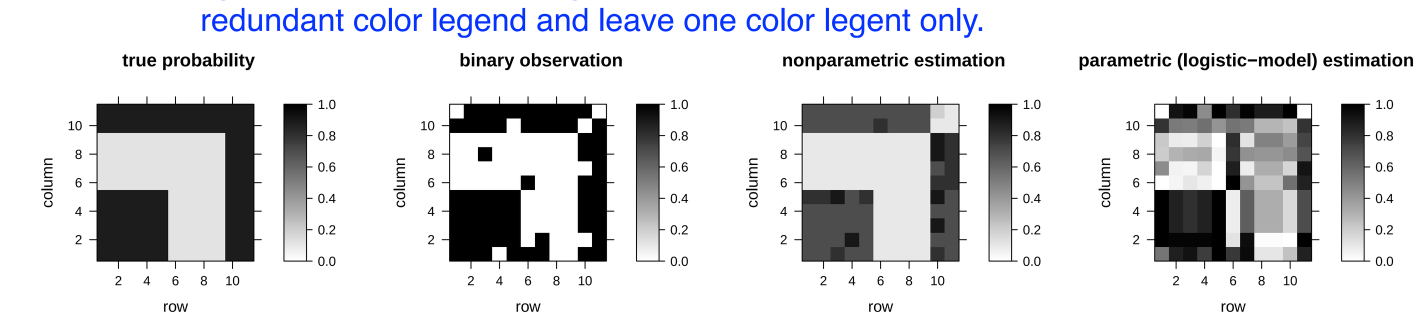
$$[\boldsymbol{X}_{ij}]_{pq} = \begin{cases} 1 & \text{if } (p,q) = (i,j). \\ 0 & \text{otherwise} \end{cases}$$

comma comma

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 $p_{ij} = \mathbb{P}(y_{ij} = 1|\boldsymbol{X}_{ij})$.

figure title too small. Enlarge the title. You could remove redundant color legend and leave one color legent only.



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

Theoretical results

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

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

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

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

hat p: \mathbb{R}^{d1\times d2} \to [0,1]

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

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

where α is a regularity parameter determined by the true probability. probability function.

If $\alpha > 1$ and $d_1 = d_2 = d$, we have In particular, then …

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

References

[1] Chanwoo Lee and Miaoyan Wang. "Tensor denoising and completion based on ordinal observations". ~~In:~~ *International conference on machine learning* (2020).

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

[3] 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.