Optimal nonparametric classification via radial distance

A. Okuno[1,2], R. Cao[3,2], K. Nakagawa[4], H. Shimodaira[3,2]

[1]Institute of Statistical Mathematics [2]RIKEN AIP [3]Kyoto University
[4]Nomura Asset Management Co., Ltd.

Related Publications of Ours

In English:

(1) Akifumi Okuno and Hidetoshi Shimodaira. "Extrapolation Towards Imaginary 0-Nearest Neighbour and Its Improved Convergence Rate", Advances in Neural Information Processing Systems 33 (**NeurIPS 2020**), pages 21889-21899.

(2) Ruixing Cao*, Akifumi Okuno*, Kei Nakagawa and Hidetoshi Shimodaira. "Improving Nonparametric Classification via Local Radial Regression with an Application to Stock Prediction", 23 pages. **arXiv:2112.13951**. (*First co-author)

In Japanese (JSAI Annual Proceedings):

(3) Ruixing Cao, Takuma Tanaka, Akifumi Okuno, Hidetoshi Shimodaira. "A Study on Regression and Loss Functions for Multiscale k-Nearest Neighbour". 2021. 4pages.

(4) Takuma Tanaka, Akifumi Okuno, Hidetoshi Shimodaira. "Extreme Multi-Label Classification of Images via Multiscale k-Nearest Neighbour". 2021. 4pages.

We had no opportunity to make a presentation for the study conducted 2 years ago...

Background

Regression/classification

Let $(X, Y) \in \mathcal{X} \times \mathbb{R}$ be a pair of covariate and response variables, defined with

$$f(x) = \mathbb{E}(Y \mid X = x).$$

In the regression problem, we estimate the function f from i.i.d. observations $\{(x_i, y_i)\}_{i=1}^{n}$.

- $f_\theta(x) = \langle \theta_1, x \rangle + \theta_2$ (linear regression),
- $f_\theta(x) = \langle \theta_3, \sigma(\langle \theta_1, x \rangle + \theta_2) \rangle + \theta_4$ (neural network), and so forth.

Problem in this study:

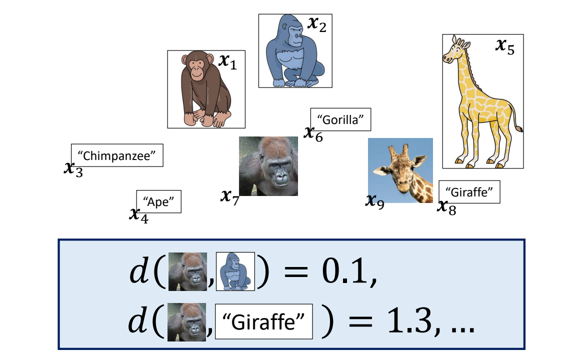
In real-world situations, several different forms of covariates are mixed. For instance,

$$\mathcal{X} = \bigcup_{q \in \mathbb{N}} \mathbb{R}^q;$$

typical regression functions cannot be applied to both $x_1 \in \mathbb{R}^{q_1}$ and $x_2 \in \mathbb{R}^{q_2}$ simultaneously.

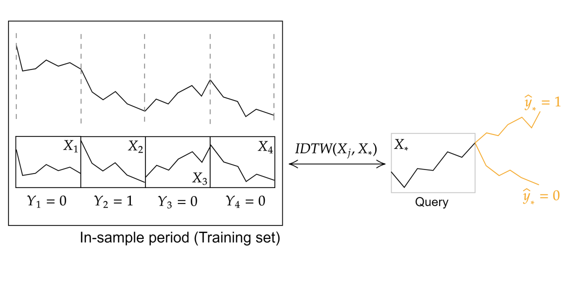
Examples

(Example 1) Mixture of pictures/drawings/texts



- $\mathbf{x}_i$: an image or a word,
- y_i: whether the object represents primates (霊長類 in Japanese) or not.

(Example 2) Time series of different lengths



- $\mathbf{x}_i$: stock price record in ith month ($\in \mathbb{R}^{\text{“days in } i\text{th month”}}$),
- y_i: whether the price increases ($y_i = 1$) or decreases ($y_i = 0$).

Distance-based approaches

Let $\mathcal{X} = \mathbb{R}^q$, $h > 0$ and consider a kernel smoother

$$\hat{f}_h^{(\mathrm{KS})}(\mathbf{x}) = \frac{1}{|\mathcal{N}_h(\mathbf{x})|} \sum_{j \in \mathcal{N}_h(\mathbf{x})} y_j, \quad \mathcal{N}_h(\mathbf{x}) := \{i \mid d(\mathbf{x}, \mathbf{x}_i) \leq h\}.$$

Then, the distance d can be replaced with other discrepancy functions formally. For instance, if $\{\mathbf{x}_i\}$ represents the time-series of different lengths, we may employ

$$d(\mathbf{x}, \mathbf{x}') := \mathrm{DynamicTimeWarping}(\mathbf{x}, \mathbf{x}').$$

👍 Distance-based approaches are (formally) widely-applicable.

Kernel smoother with $k = |\mathcal{N}_h(\mathbf{x})|$ is called *k-nearest neighbour (k-NN)* estimator.

(Higher-order) asymptotic bias and its correction

Consider the simple case $\mathcal{X} = \mathbb{R}^q$. While the kernel smoother and k-NN estimators are *consistent*, i.e.,

$$\hat{f} \to^p f,$$

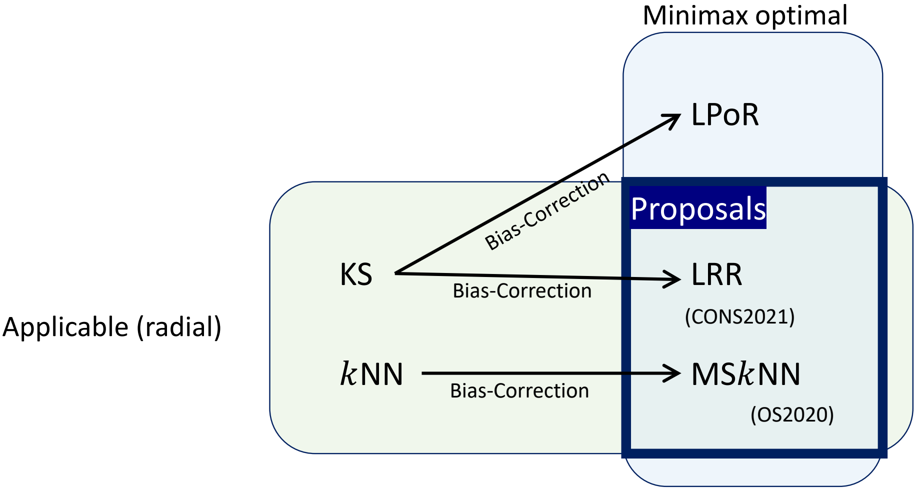
they are *not minimax optimal* if f is highly-smooth. Conventional local polynomial (LPoR) estimator corrects the asymptotic bias.

- KS and kNN: 👍 *widely-applicable*, 👎 *not optimal*.
- LPoR: 👎 *not widely-applicable*, 👍 *optimal*.

Problem: can we correct the asymptotic bias while holding the applicability?

Proposal

Overview

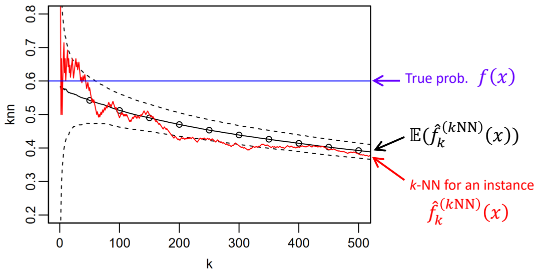

Minimax optimal
LPoR
Proposals
Bias-Correction
KS
LRR
Bias-Correction
(CONS2021)
Applicable (radial)
kNN
MSkNN
Bias-Correction
(OS2020)

Multiscale k-NN (Okuno and Shimodaira, NeurIPS2020)

k-NN estimator has larger bias as k increases.

Idea

Extrapolating k-NN estimators from $k = k_1, k_2, \ldots, k_V$ to $r = 0$ (via $r_k := d(\mathbf{x}, \mathbf{x}_{(k)})$) yields imaginary 0-NN estimator, which is also called *multiscale k-NN (MSkNN) estimator*.



👍 *minimax optimal* and *widely-applicable*.

Local Radial Regression (Cao, Okuno, Nakagawa and Shimodaira)

We define a *local radial regression (LRR)*:

$$\hat{f}^{(\mathrm{LRR})}(\mathbf{x}) = \hat{\tau}(0), \quad \hat{\tau} := \mathop{\arg\min}_{\tau \in \mathcal{P}(1,q)} \sum_{i=1}^{n} w(r_i)\{Y_i - \tau(r_i)\}^2,$$

equipped with the radial distance $r_i = d(\mathbf{x}, \mathbf{x}_i)$, decreasing non-negative function w and a polynomial function τ to be trained.

👍 *minimax optimal* and *widely-applicable*.

Theory

We prove the convergence rate for the plug-in type classifier

$$\hat{g}(x) := \mathbb{1}(\hat{f}(x) \geq 1/2) \in \{0, 1\}.$$

Going through a very bothersome calculation to prove the optimality, we have:

	Abbrev.	Application	Higher-order opt.
Multi-layer perceptron	MLP/NN	limited	No
Kernel smoother	KS	wide	No
k-Nearest neighbour	kNN	wide	No
Local polynomial regression	LPoR	limited	Optimal
Multiscale kNN (OS2020)	MSkNN	wide	Optimal
Local radial regression (CONS2021)	LRR	wide	Optimal

cf. optimal rate is $\mathcal{E}(\hat{g}_n) \asymp n^{-2\beta/(2\beta+d)}$ when assuming β-Hölder condition.

Proof overview

The optimality is shown with $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2$. For multiscale k-NN, the k-NN estimator satisfies

$$\hat{f}_k(\mathbf{x}_*) \approx \mathbb{P}(Y = 1 \mid \|X - \mathbf{x}_*\|_2 \le r)$$
$$= f(X_*) + \sum_{c=1}^{\lfloor \beta/2 \rfloor} \underbrace{b_c^*(\mathbf{x}_*) r^{2c}}_{(\star)} + \delta_{\beta,r}(\mathbf{x}_*), \quad |\delta_{\beta,r}(\mathbf{x}_*)| \lesssim r^{\beta},$$

for $r = r(k) := \|\mathbf{x}_* - \mathbf{x}_{(k)}\|_2$ and large $k \in \mathbb{N}$. Therefore, multiscale k-NN estimator removes the higher-order bias term $(\star)$ by the extrapolation (via regression). $\square$

Experiments

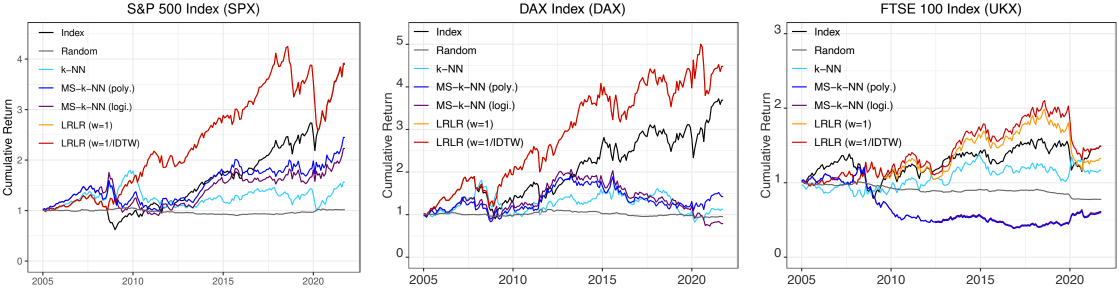
Table: n: sample size, d: dimension, m: ♯categories.
Sample average and the standard deviation for the prediction accuracy are computed on 10 times experiments. Best scores are **bolded**, and second best scores are <u>underlined</u>.

Dataset	n	d	m	kNN			**MSkNN**	
				$w_i = 1/k$	$w_i \geq 0$	$w_i \in \mathbb{R}$	via $r(k)$	via $\log k$
Iris	150	4	3	0.83 ± 0.04	0.92 ± 0.05	0.92 ± 0.04	$\underline{0.93} \pm 0.04$	$\mathbf{0.96} \pm 0.04$
Glass identification	213	9	6	0.58 ± 0.06	$\underline{0.64} \pm 0.06$	$\mathbf{0.67} \pm 0.05$	$\underline{0.64} \pm 0.05$	$\underline{0.64} \pm 0.05$
Ecoli	335	7	8	0.80 ± 0.03	$\mathbf{0.85} \pm 0.03$	$\underline{0.84} \pm 0.02$	$\mathbf{0.85} \pm 0.02$	$\underline{0.84} \pm 0.02$
Diabetes	768	8	2	$\mathbf{0.75} \pm 0.03$	$\underline{0.74} \pm 0.03$	0.70 ± 0.04	$\mathbf{0.75} \pm 0.03$	0.71 ± 0.03
Biodegradation	1054	41	2	$\underline{0.84} \pm 0.02$	$\mathbf{0.86} \pm 0.03$	0.79 ± 0.02	$\mathbf{0.86} \pm 0.02$	0.80 ± 0.02
Banknote	1371	4	2	0.95 ± 0.01	$\underline{0.98} \pm 0.01$	0.97 ± 0.01	$\underline{0.98} \pm 0.01$	$\mathbf{0.99} \pm 0.00$
Yeast	1484	8	10	$\underline{0.57} \pm 0.02$	$\mathbf{0.58} \pm 0.02$	0.54 ± 0.03	$\mathbf{0.58} \pm 0.02$	0.54 ± 0.02
Wireless localization	2000	7	4	$\underline{0.97} \pm 0.00$	$\mathbf{0.98} \pm 0.00$	$\mathbf{0.98} \pm 0.01$	$\mathbf{0.98} \pm 0.00$	$\mathbf{0.98} \pm 0.01$
Spambase	4600	57	2	$\underline{0.90} \pm 0.01$	$\mathbf{0.91} \pm 0.00$	0.86 ± 0.01	$\mathbf{0.91} \pm 0.00$	0.87 ± 0.01
Robot navigation	5455	24	4	0.81 ± 0.01	$\mathbf{0.86} \pm 0.01$	0.81 ± 0.01	$\underline{0.84} \pm 0.01$	$\underline{0.84} \pm 0.01$
Page blocks	5473	10	5	$\underline{0.95} \pm 0.01$	$\underline{0.95} \pm 0.01$	$\mathbf{0.96} \pm 0.01$	$\mathbf{0.96} \pm 0.01$	$\mathbf{0.96} \pm 0.01$
MAGIC	19020	10	2	0.82 ± 0.00	0.82 ± 0.00	$\mathbf{0.84} \pm 0.01$	$\underline{0.83} \pm 0.00$	$\underline{0.83} \pm 0.00$
Avila	20867	10	12	0.63 ± 0.01	0.68 ± 0.01	$\mathbf{0.70} \pm 0.01$	$\underline{0.69} \pm 0.00$	$\mathbf{0.70} \pm 0.01$

Application to Stock Prediction

MSkNN and LRR are applied to the stock prediction problem of S&P500, S&P500/TSX, EURO, ...

- $\mathbf{x}_i$: stock price record in ith month ($\in \mathbb{R}^{\text{"days in } i\text{th month"}}$),
- y_i: whether the price increases ($y_i = 1$) or decreases ($y_i = 0$).



Application to Stock Prediction

Table: Predictive classification accuracy. A higher score is better: the best and the second-best are bolded and underlined, respectively.

	S&P 500	S&P/TSX	EURO.	FTSE.	DAX	CAC.	TOPIX	Hang Seng
random	0.492	0.495	0.498	0.482	0.492	0.490	0.493	0.486
k-NN	0.574	<u>0.594</u>	<u>0.510</u>	<u>0.500</u>	0.530	<u>0.525</u>	<u>0.500</u>	<u>0.564</u>
MSkNN	<u>0.604</u>	0.559	**0.525**	0.485	<u>0.545</u>	0.495	**0.515**	0.530
LRLR	**0.649**	**0.609**	0.505	**0.574**	**0.609**	**0.550**	0.465	**0.574**

Conclusion

Conclusion

	Abbrev.	Applicability	Optimality
Multi-layer perceptron	MLP/NN	limited	No
Kernel smoother	KS	Good	No
k-Nearest neighbour	kNN	Good	No
Local polynomial regression	LPoR	Less	Optimal
Multiscale kNN (OS2020)	MSkNN	Good	Optimal
Local radial regression (CONS2021)	LRR	Good	Optimal

- We proposed a *widely-applicable* and *optimal* MSkNN and LRR estimators.
- This study was mainly based on:
  (1) Okuno and Shimodaira (NeurIPS2020)
  (2) Cao, Okuno, Nakagawa, and Shimodaira (arXiv:2112.13951)

- Contact Info.: `okuno@ism.ac.jp`