Robust modal regression with direct gradient approximation of modal regression risk

Hiroaki Sasaki[1], **Tomoya Sakai**[2], **Takafumi Kanamori**[3,4]
[1]Future University Hakodate, Hokkaido, Japan [2]NEC Corporation, Tokyo, Japan
[3]Tokyo Institute of Technology, Tokyo, Japan [4]RIKEN AIP, Tokyo, Japan

Abstract

Modal regression is aimed at estimating the global mode (i.e., global maximum) of the conditional density function of the output variable given input variables, and has led to regression methods robust against a wide-range of noises. A typical approach for modal regression takes a two-step approach of firstly approximating the modal regression risk (MRR) and of secondly maximizing the approximated MRR with some gradient method. However, this two-step approach can be suboptimal in gradient-based maximization methods because a good MRR approximator does not necessarily give a good gradient approximator of MRR. In this paper, we take a novel approach of *directly* approximating the gradient of MRR in modal regression. Based on the direct approach, we first propose a modal regression method with reproducing kernels where a new update rule to estimate the conditional mode is derived based on a fixed-point method. Then, the derived update rule is theoretically investigated. Furthermore, since our direct approach is compatible with recent sophisticated stochastic gradient methods (e.g., Adam), another modal regression method is also proposed based on neural networks. Finally, the superior performance of the proposed methods is demonstrated on various artificial and benchmark datasets.

1 Introduction

The goal of modal regression is to estimate the global mode (i.e., global maximum) of the conditional density of the output variable given input variables [Sager and Thisted, 1982, Collomb et al., 1986, Yao et al., 2012, Feng et al., 2017]. In stark contrast with the Gaussian noise assumption in conventional regression methods, modal regression has much weaker noise assumptions, and has led to regression methods robust to a wide-range of noises. Applications of modal regression includes prediction of Alzheimer's disease [Wang et al., 2017], face recognition [Wang et al., 2019], etc. See also a recent comprehensive review article by Chen [2018].

A number of approaches have been adopted so far to propose regression methods against nonGaussian and/or nonstationary noises. A well-known approach is to use robust loss functions such as least absolute deviations or Huber loss [Huber and Ronchetti, 2009]. However, most of robust loss functions are intended for symmetric noises, and thus can be vulnerable to highly skewed (i.e., asymmetric) noises. A sophisticated approach to handle nonstationary noises is the heteroscedastic Gaussian process regression (HGPR) [Kersting et al., 2007, Lázaro-Gredilla and Titsias, 2011]. However, HGPR usually assumes that the noise density is a zero-mean Gaussian with a nonstationary variance, and therefore may not also work well to skewed noises. In contrast, again, modal regression makes weak assumptions, and heavy-tailed, skewed and/or nonstationary noises are acceptable.

The mode of the conditional density has been often estimated through maximization of the empirical *modal regression risk* (MRR), which is defined as the sample average of the conditional density (or the joint density) [Sager and Thisted, 1982, Yao et al., 2012, Feng et al., 2017]. A naive approach to modal regression takes a two-step approach of firstly approximating the empirical MRR via conditional (or joint) density estimation (e.g., by kernel density estimation (KDE)), and secondly of maximizing the approximated risk by some gradient method. However, the fundamental challenge in maximization is accurate approximation of the gradient of MRR rather than MRR itself. Thus, this two-step approach might be suboptimal because a good MRR approximator does not necessarily mean a good gradient

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

approximator of MRR. Yao et al. [2012] apply an EM algorithm, but the gradient of a risk function is still computed through the naive two-step approach with KDE.

Another approach to modal regression employs a surrogate risk of MRR [Lee, 1989, Yao and Li, 2014, Feng et al., 2017, Wang et al., 2017]. The advantage of this approach is that high-dimensional density estimation can be avoided. However, a drawback is that the surrogate risk includes a manually tuning hyperparameter, and it is not straightforward to select it since the surrogate risk itself depends on the hyperparameter. Moreover, when neural networks are used in large datasets, the hyperparameter selection can be computationally very expensive.

In this paper, we propose two methods for modal regression based on kernels or neural networks. In contrast with existing methods, we do not go through the approximation of MRR itself, but rather more directly approximate the gradient of MRR. The key challenge in our approach is direct estimation of the derivative of the log-conditional or joint density. To this end, under the Fisher divergence [Cox, 1985, Sasaki et al., 2014], we develop methods which *directly* estimates the log-joint density derivative without density estimation. The Fisher divergence has been employed to detect *multiple* local maxima of the conditional density [Sasaki et al., 2016], but our target is the *single* global maximum, and thus the proposed methods here are substantially different.

First, based on reproducing kernels, we develop an estimator for the log-joint density derivative, and then propose a modal regression method. As shown later, thanks to the analytic solution of our derivative estimator, a computationally efficient model selection is possible for leave-one-out cross validation. Furthermore, for modal regression, this kernel-based derivative estimator enables to derive a novel parameter update rule based on a fixed-point method for conditional mode estimation. Then, the derived update rule is theoretically investigated.

Second, we employ neural networks both for the density derivative estimation and modal regression. Our approach of directly approximating the gradient of MRR is clearly nonstandard, but turns out to be well-compatible with sophisticated stochastic gradient methods such as Adam [Kingma and Ba, 2015]: The learning rate is adaptively determined by the gradient of the empirical risk, and thus only the gradient is required to update parameters. Combined with some stochastic gradient methods, we can develop a neural-network-based method for modal regression without any efforts, and to the best of our knowledge, this is the first attempt to make use of neural networks for modal regression. Finally, we demonstrate that our modal regression methods perform well on various artificial and benchmark datasets.

2 Background

This section gives some background of modal regression and states our approach.

Problem formulation: Suppose that we are given n observations of pairs of input and output data samples drawn from the joint density $p_{\mathrm{yx}}(y, \boldsymbol{x})$ for $y \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^{d_{\mathrm{x}}}$ as $\mathcal{D} := \left\{(y_i, \boldsymbol{x}_i^\top)^\top\right\}_{i=1}^n$. Under the assumption that the conditional mode uniquely exists, our goal is to estimate the following *modal regression function* f_{M}:

$$f_{\mathrm{M}}(\boldsymbol{x}) := \operatorname*{argmax}_{t \in \mathbb{R}} \log p_{\mathrm{y|x}}(t|\boldsymbol{x}), \tag{1}$$

where $p_{\mathrm{y|x}}$ denotes the conditional density of y given $\boldsymbol{x}$.

Review of modal regression: To make our approach clearer, we adopt the terminologies in Feng et al. [2017]. Let us assume that the output variable y is generated from the following model:

$$y = f^*(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \tag{2}$$

where $\epsilon(\boldsymbol{x})$ denotes an additive noise, and f^* is an unknown function called the *conditional mode function*. The fundamental assumption in (2) is the *zero conditional mode assumption*:

$$\mathrm{Mode}(\epsilon|\boldsymbol{x}) := \operatorname*{argmax}_{t \in \mathbb{R}} \log p_{\epsilon|\mathrm{x}}(t|\boldsymbol{x}) = 0, \tag{3}$$

where $p_{\epsilon|\boldsymbol{x}}$ denotes the conditional density of the noise ϵ given input variables $\boldsymbol{x}$. This mode assumption is very general and weak because $p_{\epsilon|\boldsymbol{x}}$ can be skewed (i.e., asymmetric), heavy-tailed, and/or dependent to $\boldsymbol{x}$ (i.e., nonstationary). As in heteroscedastic Gaussian process [Kersting et al., 2007], a zero-mean Gaussian noise even with a nonstationary variance is a special case of (3). Mode assumption (3) ensures that $f_{\mathrm{M}}(\boldsymbol{x}) = f^*(\boldsymbol{x})$ from (2).

To estimate f_{M} by a model $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$, the *modal regression risk* (MRR) is defined as

$$\mathcal{R}(\boldsymbol{\theta}) := \int p_{\mathrm{x}}(\boldsymbol{x}) \log p_{\mathrm{y|x}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})|\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{4}$$

where p_{x} denotes the marginal density of $\boldsymbol{x}$. An alternative risk has been also defined using the joint density $p_{\mathrm{yx}}(y, \boldsymbol{x})$ [Sager and Thisted, 1982, Yao et al., 2012]. Following Theorem 3 in Feng et al. [2017], it can be proved that the (global) maximizer of $\mathcal{R}(\boldsymbol{\theta})$ equals to the modal regression function f_M when both $f_{\boldsymbol{\theta}}$ and f_M belong to the same function set. In practice, the empirical version of $\mathcal{R}(\boldsymbol{\theta})$ is used as

$$\widetilde{\mathcal{R}}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \log p_{\mathrm{y|x}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)|\boldsymbol{x}_i). \tag{5}$$

Then, $\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ can be maximized using the following gradient with respect to parameters $\boldsymbol{\theta}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial \boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\frac{\partial}{\partial y}\log p_{\mathrm{y}|\mathrm{x}}(y|\boldsymbol{x}_i)\Big|_{y=f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial \boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x}_i)\Big|_{y=f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}, \tag{6}$$

where note that $\frac{\partial}{\partial y}\log p_{\mathrm{y}|\mathrm{x}}(y|\boldsymbol{x}) = \frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})$. To approximate the gradient (6), the fundamental task is to estimate $\frac{\partial}{\partial y}\log p_{\mathrm{y}|\mathrm{x}}(y|\boldsymbol{x})$ or $\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})$. To this end, a naive approach takes two steps of firstly estimating $\log p_{\mathrm{y}|\mathrm{x}}(y|\boldsymbol{x})$ or $\log p_{\mathrm{yx}}(y,\boldsymbol{x})$ and then of computing the derivative with respect to y. However, such a naive estimation procedure can be suboptimal because a good density estimator does not necessarily mean a good log-density derivative estimator. Thus, a better approach to approximate the gradient (6) would be to directly estimate $\frac{\partial}{\partial y}\log p_{\mathrm{y}|\mathrm{x}}(y|\boldsymbol{x})$ or $\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})$ without going through density estimation.

Another approach for modal regression employs the following empirical surrogate risk [Lee, 1989, Yao and Li, 2014, Feng et al., 2017, Wang et al., 2017], which has been used in the maximum correntropy criterion as well [Gunduz and Principe, 2009]:

$$\widetilde{\mathcal{R}}^{\sigma}(\boldsymbol{\theta}) := \frac{1}{n\sigma}\sum_{i=1}^{n}\psi\left(\frac{y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\sigma}\right), \tag{7}$$

where σ is a positive width parameter, ψ is a nonnegative function such that $\psi(u) = \psi(-u)$, $\psi(u) \leq \psi(0)$ for all u and $\int \psi(u)\mathrm{d}u = 1$. Feng et al. [2017] proved the following relation:

$$\widetilde{\mathcal{R}}^{\sigma}(\boldsymbol{\theta}) \xrightarrow{n\to\infty} \frac{1}{\sigma}\int \psi\left(\frac{y - f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\sigma}\right) p_{\mathrm{yx}}(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x} \xrightarrow{\sigma\to 0} \int p_{\mathrm{y}|\mathrm{x}}(f_{\boldsymbol{\theta}}(\boldsymbol{x})|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.$$

Thus, $\widetilde{\mathcal{R}}^{\sigma}(\boldsymbol{\theta})$ can be regarded as a surrogate of $\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ in (5) without the logarithm. This approach seems appealing because we can avoid high-dimensional density estimation. On the other hand, a significant drawback is that the performance strongly depends on the choice of the hyperparameter σ, and it is not straightforward to choose a right value. We may use cross validation (CV) in practice, but this approach can be problematic because of the following two reasons: First, it seems unclear what criterion in CV should be used to select σ because $\widetilde{\mathcal{R}}^{\sigma}$ itself depends on σ[1]; Second, even if there was a valid criterion for CV, then we have to perform a nested CV to choose both σ and hyperparameters in $f_{\boldsymbol{\theta}}$ (e.g., the width parameter in a kernel function), which is computationally very expensive. Furthermore, if neural networks are employed, a grid-search for selection of σ in large datasets could be computationally costly.

[1]The squared-loss may be used in CV. However, the squared-loss implicitly assumes the Gaussian noise, and thus may prohibit us to make full use of the advantages of modal regression.

Here, our approach is to directly approximate the gradient $\frac{\partial}{\partial \boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ without going through any approximation of $\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ itself. To this end, the key idea is to directly estimate the log-density derivative $\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})$ in (6). With the direct approximation, we propose two methods for modal regression based on reproducing kernels and neural networks.

3 Direct log-density derivative estimation

This section develops a method which directly estimates the derivative of the logarithmic joint density based on reproducing kernels. Here, our contributions are twofold: Theorem 1 and an analytic form of the leave-one-out cross-validation score for model selection. Another derivative estimator based on neural networks is proposed in Section 4.2 under the same divergence.

Kernelized estimator: To estimate the log-joint density derivative, we *directly* fit a model $r(y,\boldsymbol{x})$ under the Fisher divergence [Cox, 1985, Sasaki et al., 2014]:

$$\begin{aligned} J(r) &:= \frac{1}{2}\int\left\{r(y,\boldsymbol{x}) - \frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})\right\}^2 p_{\mathrm{yx}}(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2}\int\{r(y,\boldsymbol{x})\}^2 p_{\mathrm{yx}}(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x} \\ &\quad - \int r(y,\boldsymbol{x})\left\{\frac{\partial}{\partial y}p_{\mathrm{yx}}(y,\boldsymbol{x})\right\}\mathrm{d}y\mathrm{d}\boldsymbol{x} + C, \end{aligned} \tag{8}$$

where $C := \frac{1}{2}\int\left\{\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})\right\}^2 p_{\mathrm{yx}}(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x}$. The second term in (8) seems difficult to estimate, but the well-known *integration by parts* technique transforms it as follows:

$$\begin{aligned} &\int r(y,\boldsymbol{x})\left\{\frac{\partial}{\partial y}p_{\mathrm{yx}}(y,\boldsymbol{x})\right\}\mathrm{d}y\mathrm{d}\boldsymbol{x} \\ &\quad = -\int\left\{\frac{\partial}{\partial y}r(y,\boldsymbol{x})\right\}p_{\mathrm{yx}}(y,\boldsymbol{x})\mathrm{d}y\mathrm{d}\boldsymbol{x}, \end{aligned} \tag{9}$$

where we assumed that for all $\boldsymbol{x}$,

$$\lim_{|y|\to\infty} r(y,\boldsymbol{x})p_{\mathrm{yx}}(y,\boldsymbol{x}) = 0. \tag{10}$$

The right-hand side on (9) is the expectation of the derivative of the model, which can be easily estimated from samples. Finally, the empirical Fisher divergence

up to the ignorable constant is obtained as

$$\widehat{J}(r) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}r(y_i, \boldsymbol{x}_i)^2 + \frac{\partial}{\partial y}r(y_i, \boldsymbol{x}_i)\right]. \quad (11)$$

Next, we define our estimator based on a reproducing kernel Hilbert space $\mathcal{H}$ (RKHS) as

$$\widehat{r} = \operatorname*{argmin}_{r\in\mathcal{H}}\left[\widehat{J}(r) + \frac{\lambda}{2}\|r\|_{\mathcal{H}}^2\right], \quad (12)$$

where $\|\cdot\|_{\mathcal{H}}$ and $\lambda(>0)$ denote the RKHS norm and regularization parameter, respectively. Then, the following theorem shows that $\widehat{r}$ can be efficiently obtained by solving systems of linear equations:

Theorem 1. *Let us express $(y, \boldsymbol{x}^\top)^\top$ by $\boldsymbol{z}$. $\widehat{r}$ is given by*

$$\widehat{r}(\boldsymbol{z}) = \sum_{i=1}^{n}\left[\widehat{\alpha}_i k(\boldsymbol{z}, \boldsymbol{z}_i) - \frac{1}{n\lambda}\frac{\partial}{\partial y'}k(\boldsymbol{z}, \boldsymbol{z}')\Big|_{\boldsymbol{z}'=\boldsymbol{z}_i}\right], \quad (13)$$

where $k(\boldsymbol{z}, \boldsymbol{z}')$ denotes the kernel function in $\mathcal{H}$, $\boldsymbol{z}_i := (y_i, \boldsymbol{x}_i^\top)^\top$ and $\boldsymbol{z}' := (y', \boldsymbol{x}'^\top)^\top$. The coefficients $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_n)^\top$ are the solution of the following system of linear equations:

$$(\boldsymbol{K} + n\lambda\mathbf{I}_n)\widehat{\boldsymbol{\alpha}} = \frac{1}{n\lambda}\boldsymbol{G}\mathbf{1}_n, \quad (14)$$

where $\mathbf{1}_n = (1, 1, \ldots, 1)^\top$ is an n-dimensional vector, $\mathbf{I}_n$ denotes the n by n identity matrix, $[\boldsymbol{K}]_{ij} = k(\boldsymbol{z}_i, \boldsymbol{z}_j)$ and $[\boldsymbol{G}]_{ij} = \frac{\partial}{\partial y'}k(\boldsymbol{z}_i, \boldsymbol{z}')|_{\boldsymbol{z}'=\boldsymbol{z}_j}$.

The proof is deferred to Section A in the supplementary material. This paper calls this method the *kernelized least-squares log-density derivatives* (K-LSLD). Section 4 develops a modal regression method based on K-LSLD. Previously, based on the same divergence (11), Cox [1985] proposed a practical estimator with a one-dimensional piecewise polynomial kernel [Wahba, 1990], while Sasaki et al. [2014, 2016] applied the ℓ_2 regularizer for model parameters. In contrast, we employed the general kernel function and regularizer of the RKHS norm.

Leave-one-out cross-validation: The performance of K-LSLD depends on model selection (parameters in the kernel function and regularization parameter). Here, we perform the leave-one-out cross-validation (LOOCV) for model selection whose score is given by

$$\mathrm{LOOCV} = \frac{1}{n}\sum_{l=1}^{n}\left[\frac{1}{2}\{\widehat{r}^{(l)}(y_l, \boldsymbol{x}_l)\}^2 + \frac{\partial}{\partial y}\widehat{r}^{(l)}(y_l, \boldsymbol{x}_l)\right],$$

where $\widehat{r}^{(l)}$ denotes the estimator obtained from the collection of data samples except for the l-th data sample (i.e. $\mathcal{D}\backslash(y_l, \boldsymbol{x}_l^\top)^\top$). LOOCV is usually time-consuming. However, thanks to the analytic solution in Theorem 1, the LOOCV score can be efficiently computed. The concrete form of the LOOCV score is not expressed here because it is rather complicated. Details are presented in Section B of the supplementary material.

4 Modal regression with direct gradient approximation

This section first develops a kernel-based method for modal regression where reproducing kernels are used twice in log-density derivative and mode estimation. Based on K-LSLD, we derive a parameter update rule using a fixed-point method. Then, the derived update rule is theoretically investigated. Finally, another modal regression method is also proposed based on neural networks.

4.1 Direct modal regression with kernels

Fixed-point-based parameter update rule: Here, we assume that a model $f_{\boldsymbol{\theta}}$ to estimate the conditional mode belongs to an RKHS. Then, under the empirical modal regression risk (5), the representer theorem [Kimeldorf and Wahba, 1971] suggests the optimal form of $f_{\boldsymbol{\theta}}$ as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{n}\theta_k k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_k) = \boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}), \quad (15)$$

where $k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_i)$ denotes a kernel function, $\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}) = (k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_1), k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_2), \ldots, k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_n))^\top$, and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^\top$. With this kernel model $f_{\boldsymbol{\theta}}$, we obtain the gradient of the empirical MRR from (6) as

$$\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y, \boldsymbol{x}_i)\Big|_{y=\boldsymbol{\theta}^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)}. \quad (16)$$

Next we substitute K-LSLD into $\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y, \boldsymbol{x})$, and derive a update rule for $\boldsymbol{\theta}$ using a fixed-point method. To this end, we first express K-LSLD in (13) as

$$\widehat{r}(y, \boldsymbol{x}) := \sum_{l=1}^{n}\left\{\widehat{\alpha}_l k_{\mathrm{y}}(y, y_l) - \frac{1}{n\lambda}\frac{\partial}{\partial y'}k_{\mathrm{y}}(y, y')\Big|_{y'=y_l}\right\}k_{\mathrm{x}}(\boldsymbol{x}, \boldsymbol{x}_l), \quad (17)$$

where we used the kernel function in K-LSLD (not in $f_{\boldsymbol{\theta}}$) as $k(\boldsymbol{z}, \boldsymbol{z}') = k_{\mathrm{y}}(y, y') \times k_{\mathrm{x}}(\boldsymbol{x}, \boldsymbol{x}')$ with two kernel functions, k_{y} and k_{x}. Then, we further restrict the form of k_{y} as

$$k_{\mathrm{y}}(y, y') = \phi\left\{\frac{(y-y')^2}{2\sigma_{\mathrm{y}}^2}\right\},$$

where $\sigma_{\mathrm{y}}(>0)$ denotes the width parameter, ϕ is a convex, and monotonically non-increasing function. For instance, $\phi(t)=\exp(-t)$, $k_{\mathrm{y}}(y,y')$ is the Gaussian kernel.

Substituting $\widehat{r}(y,\boldsymbol{x})$ into $\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})$ in (16) enables us to approximate the gradient $\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ as

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta}) &\approx \frac{1}{n}\sum_{i=1}^{n}\widehat{r}(\boldsymbol{\theta}^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i),\boldsymbol{x}_i)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)\\ &= \boldsymbol{h}(\boldsymbol{\theta})-\boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{\theta},\end{aligned} \tag{18}$$

where with $\varphi(t):=-\frac{\mathrm{d}}{\mathrm{d}t}\phi(t)$,

$$\begin{aligned}\boldsymbol{H}(\boldsymbol{\theta}) &:= \frac{1}{n^2\lambda\sigma_{\mathrm{y}}^2}\sum_{i=1}^{n}\sum_{l=1}^{n}\varphi\left\{\frac{(\boldsymbol{\theta}^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)-y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\\ &\quad\times k_{\mathrm{x}}(\boldsymbol{x}_i,\boldsymbol{x}_l)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)^\top,\end{aligned} \tag{19}$$

$$\begin{aligned}\boldsymbol{h}(\boldsymbol{\theta}) &:= \frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{n}\left[\widehat{\alpha}_l\phi\left\{\frac{(\boldsymbol{\theta}^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)-y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\right.\\ &\left.+\frac{y_l}{n\lambda\sigma_{\mathrm{y}}^2}\varphi\left\{\frac{(\boldsymbol{\theta}^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)-y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\right]k_{\mathrm{x}}(\boldsymbol{x}_i,\boldsymbol{x}_l)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i).\end{aligned}$$

Then, under the assumption that $\boldsymbol{H}(\boldsymbol{\theta})$ is invertible, setting the right-hand side in (18) to zero gives the following iterative update rule based on a fixed-point method:

$$\boldsymbol{\theta}^{\tau+1}=\boldsymbol{H}^{-1}(\boldsymbol{\theta}^\tau)\boldsymbol{h}(\boldsymbol{\theta}^\tau), \tag{20}$$

where $\boldsymbol{\theta}^\tau$ denotes the τ-th update of $\boldsymbol{\theta}$. The following relation would be helpful to intuitively understand how the update rule (20) works, which is derived by multiplying $\boldsymbol{H}^{-1}(\boldsymbol{\theta}^\tau)$ to both sides of (18) and applying (20):

$$\boldsymbol{\theta}^{\tau+1}\approx\boldsymbol{\theta}^\tau+\boldsymbol{H}^{-1}(\boldsymbol{\theta}^\tau)\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\tau}. \tag{21}$$

Eq.(21) implies that the update rule (20) performs gradient ascent when $\boldsymbol{H}(\boldsymbol{\theta})$ is positive definite. Compared with a standard gradient method, the update rule (20) is advantageous because there are no additional tuning parameter such as a step size parameter, which is reminiscent of the Newton method. Below, we more rigorously investigate a property of the update rule (20).

An outline of our kernel-based algorithm called the *direct modal regression with kernels* (DMR-K) is given in Algorithm 1. The important problem is how to determine the initial parameters $\boldsymbol{\theta}_0$ because the maximization of the modal regression risk may require to solve a non-convex optimization problem. As a remedy, we first perform some regression method based on the squared loss or absolute deviations, and use the estimated coefficient vector as $\boldsymbol{\theta}_0$. In addition, to ensure that $\boldsymbol{H}(\boldsymbol{\theta})$ is invertible, we may add a small constant to the diagonals of $\boldsymbol{H}(\boldsymbol{\theta})$ in practice.

Algorithm 1: Direct modal regression with kernels (DMR-K)

Input: Data $\{(y_i,\boldsymbol{x}_i)\}_{i=1}^n$, initial parameters $\boldsymbol{\theta}_0$.

1. Estimate $\frac{\partial}{\partial y}\log p_{\mathrm{yx}}(y,\boldsymbol{x})$ as in Theorem 1.
2. After initializing $\boldsymbol{\theta}$ by $\boldsymbol{\theta}_0$, repeat to update $\boldsymbol{\theta}$ by (20) until some criterion is satisfied.

Output: $\widehat{f}_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x}):=\widehat{\boldsymbol{\theta}}^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})$ with the optimized $\widehat{\boldsymbol{\theta}}$.

Monotonic ascending property of DMR-K: Here, we theoretically investigate DMR-K. In particular, we focus on the *monotonic ascending property* where the following inequality holds at every parameter update τ:

$$\widetilde{\mathcal{R}}(\boldsymbol{\theta}^{\tau+1})-\widetilde{\mathcal{R}}(\boldsymbol{\theta}^\tau)\geq 0. \tag{22}$$

This inequality indicates that $\boldsymbol{\theta}$ is updated such that $\widetilde{\mathcal{R}}$ is monotonically increased at every update. However, it is not straightforward to investigate this monotonic ascending property in our method because there is no approximation of the empirical risk $\widetilde{\mathcal{R}}(\boldsymbol{\theta})$.

To cope with this problem, we employ the well-known formula of *path integral* [Strang, 1991]: Regarding the vector field $\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ and a differentiable curve $\boldsymbol{\theta}(t)$ from $\boldsymbol{\theta}(0)=\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}(1)=\boldsymbol{\theta}_2$, the path integral is given by

$$D[\boldsymbol{\theta}_2|\boldsymbol{\theta}_1]:=\int_0^1\langle\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta}(t)),\dot{\boldsymbol{\theta}}(t)\rangle\mathrm{d}t=\widetilde{\mathcal{R}}(\boldsymbol{\theta}_2)-\widetilde{\mathcal{R}}(\boldsymbol{\theta}_1), \tag{23}$$

where $\dot{\boldsymbol{\theta}}(t):=\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta}(t)$ and $\langle\cdot,\cdot\rangle$ denotes the inner product. The key point is that the right-hand side is independent to any choice of paths and computed only using $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Our analysis uses the following simple path:

$$\boldsymbol{\theta}(t)=\boldsymbol{\theta}_1+t(\boldsymbol{\theta}_2-\boldsymbol{\theta}_1), \tag{24}$$

where $0\leq t\leq 1$.

To investigate the monotonic ascending property (22), we approximate the difference of the empirical risk function, $\widetilde{\mathcal{R}}(\boldsymbol{\theta}_2)-\widetilde{\mathcal{R}}(\boldsymbol{\theta}_1)$, by substituting our gradient approximator (18) into $\frac{\partial}{\partial\boldsymbol{\theta}}\widetilde{\mathcal{R}}(\boldsymbol{\theta})$ in (23) under the path (24) as follows:

$$\begin{aligned}&\widehat{D}_{\widehat{r}}[\boldsymbol{\theta}_2|\boldsymbol{\theta}_1]\\ &:=\frac{1}{n}\sum_{i=1}^{n}\int_0^1\widehat{r}(\boldsymbol{\theta}(t)^\top\boldsymbol{k}(\boldsymbol{x}_i),\boldsymbol{x}_i)\boldsymbol{k}(\boldsymbol{x}_i)^\top(\boldsymbol{\theta}_2-\boldsymbol{\theta}_1)\mathrm{d}t.\end{aligned} \tag{25}$$

The path integral formula (23) clearly indicates that (25) is an approximator of $\widetilde{\mathcal{R}}(\boldsymbol{\theta}_2) - \widetilde{\mathcal{R}}(\boldsymbol{\theta}_1)$. Thus, our update rule (20) can be interpreted as having the monotonic ascending property when $\widehat{D}_{\widehat{\tau}}[\boldsymbol{\theta}^{\tau+1}|\boldsymbol{\theta}^{\tau}] \geq 0$ for every τ. The following theorem establishes sufficient conditions:

Theorem 2. *Assume that k_{x} is non-negative, and ϕ is a convex and monotonically non-increasing function. Then, if $\boldsymbol{H}(\boldsymbol{\theta})$ is positive definite and $\widehat{\alpha}_l = 0$ for all $l = 1, \ldots, n$, the following inequality holds under the update rule* (20)*:*

$$\widehat{D}_{\widehat{\tau}}[\boldsymbol{\theta}^{\tau+1}|\boldsymbol{\theta}^{\tau}] \geq 0.$$

The proof is deferred to Section C in the supplementary material. Conditions for k_{x} and ϕ can be easily satisfied by using the Gaussian kernel, which also ensures that $\boldsymbol{H}(\boldsymbol{\theta})$ is positive definite by definition (19). On the other hand, the condition for $\widehat{\alpha}_l$ is not fulfilled in general. However, we experimentally observed that the update rule (20) gives good results without satisfying the condition to $\widehat{\alpha}_l$. This would be because the update rule (20) possibly performs gradient ascent as implied by (21), and we also conjecture that there exists milder conditions to improve Theorem 2.

A similar analysis using path integral has been done in mode-seeking clustering [Sasaki et al., 2018]. However, Sasaki et al. [2018] proved a monotonic ascending property with respect to the probability density function, while our analysis here is for the empirical modal regression risk. Thus, the proof is substantially different.

4.2 Direct modal regression with neural networks

Here, we propose another modal regression method based on neural networks. With a neural network $f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ parametrized by $\boldsymbol{\theta}$, we compute the gradient of the empirical modal regression risk as follows:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \widetilde{\mathcal{R}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{\partial}{\partial \boldsymbol{\theta}} f_{\mathrm{NN}}(\boldsymbol{x}_i; \boldsymbol{\theta}) \right\} r^*_{\mathrm{NN}}(\boldsymbol{x}_i; \boldsymbol{\theta}), \quad (26)$$

where $r^*_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}) := \frac{\partial}{\partial y} \log p_{\mathrm{yx}}(y, \boldsymbol{x}) \Big|_{y = f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})}$. Our approach of directly approximating the gradient of an empirical modal regression risk (26) is clearly non-standard because there is no empirical risk function. However, interestingly, our direct approach is rather well-compatible with recent sophisticated stochastic gradient methods such as Adagrad [Duchi et al., 2011] and Adam [Kingma and Ba, 2015]: The learning rate is adaptively determined based on the gradient of the (mini-batch) empirical risk. Thus, approximating only the gradient is sufficient to use these stochastic optimization methods.

In addition to $f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$, we estimate $\frac{\partial}{\partial y} \log p_{\mathrm{yx}}(y, \boldsymbol{x})$ based on another neural network model $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$ with parameters $\boldsymbol{\gamma}$ based on the Fisher divergence (8). However, when feedforward neural networks were employed for $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$, we preliminarily observed that the second term in the empirical Fisher divergence (11) often diverged. This is presumably because Assumption (10) might not be fulfilled because neural networks can be sharply unbounded. To cope with this problem, we use the following form for $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$:

$$r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma}) = \sum_{k=1}^{K} w_k \exp\left[-\frac{\{y - \mu_k^{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\gamma}')\}^2}{2\sigma_k^2} \right],$$

where w_k are parameters to be estimated, σ_k denote (fixed) width parameters, and μ_k^{NN} are modelled by neural networks with parameters $\boldsymbol{\gamma}'$. This model would satisfy Assumption (10) because r_{NN} approaches to zero as $|y| \to \infty$. By substituting $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$ into $r(y, \boldsymbol{x})$ in the empirical Fisher divergence (11), we estimate all parameters $\boldsymbol{\gamma}$ (i.e., $\{w_k\}_{k=1}^K$ and $\boldsymbol{\gamma}'$) with a minibatch stochastic gradient method.

An outline of our algorithm called the *direct modal regression with neural networks* (DMR-NN) can be seen in Algorithm 2. As in DMR-K, it is an important problem to choose good initial parameters $\boldsymbol{\theta}_0$. Here, we perform *pretraining* where $f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ is trained based on the squared loss or absolute deviations in advance.

Algorithm 2: Direct modal regression with neural networks (DMR-NN)

Input: Data $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$, initial parameters $\boldsymbol{\theta}_0$

1. Estimate $\frac{\partial}{\partial y} \log p_{\mathrm{yx}}(y, \boldsymbol{x})$ by using $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$ under the empirical Fisher divergence (11).
2. Repeat the following with a neural network $f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ initialized by $\boldsymbol{\theta} = \boldsymbol{\theta}_0$:

   (A) With a random minibatch $\{\boldsymbol{x}_b^{(B)}\}_{b=1}^B$, approximate the gradient (26) by

   $$\boldsymbol{g}^{(B)} = \frac{1}{B} \sum_{b=1}^{B} \left\{ \frac{\partial}{\partial \boldsymbol{\theta}} f_{\mathrm{NN}}(\boldsymbol{x}_b^{(B)}; \boldsymbol{\theta}) \right\} \widehat{r}_{\mathrm{NN}}(\boldsymbol{x}_b^{(B)}; \boldsymbol{\theta}),$$

   where $\widehat{r}_{\mathrm{NN}}(\boldsymbol{x}_b^{(B)}; \boldsymbol{\theta}) := r_{\mathrm{NN}}(y, \boldsymbol{x}; \widehat{\boldsymbol{\gamma}})$ of the optimized $\widehat{\boldsymbol{\gamma}}$ above.

   (B) Update $\boldsymbol{\theta}$ by applying a stochastic gradient method (e.g., Adam) using $\boldsymbol{g}^{(B)}$.

Output: $\widehat{f}_{\mathrm{NN}}(\boldsymbol{x}) := f_{\mathrm{NN}}(\boldsymbol{x}; \widehat{\boldsymbol{\theta}})$ with the optimized $\widehat{\boldsymbol{\theta}}$

4.3 Discussion between DMR-K and DMR-NN

We briefly discuss pros and cons of DMR-K and DMR-NN. When n is not so large, DMR-K would be computationally efficient, and a good initialization procedure is available because least-squares (LS) or least-absolute deviations (LAD) with kernels solves a convex optimization problem and gives good initial parameters for DMR-K in practice. On the other hand, stochastic gradient methods in DMR-NN often require a huge number of iterations to update parameters. Furthermore, it could be more difficult to obtain good initial parameters than DMR-K because optimization problems with neural networks are non-convex in general even for LS and LAD.

When n is very large, it is not straightforward to use DMR-K because the inverse of the n by n matrix in (20) has to be computed. This is a general problem in most of kernel-based methods: n by n matrices (e.g., gram matrices) make kernel methods difficult or almost impossible to be applied to very large n datasets in terms of computational costs and memories. As a remedy, there exist approximation methods such as the Nyström approximation [Williams and Seeger, 2001] and random Fourier features [Rahimi and Recht, 2008]. However, the reduction of computational costs and memories by these methods may come at a cost of limiting the function approximation ability of kernel models. Regarding DMR-NN, minibatch stochastic gradient methods are applicable to very large datasets, and neural networks have been empirically shown to work well even to high-dimensional data. DMR-NN would be more advantageous to possibly high-dimensional and large datasets.

5 Numerical illustration

This section numerically illustrates the performance of DMR-K and DMR-NN.

5.1 Illustration of DMR-K on artificial datasets

Here, we investigate how DMR-K works on artificial data, and comparison is made against existing regression methods based on kernels. To estimate the conditional mode, in all methods, we used the same kernel model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})$ in (15) and employed the Gaussian kernel where the width parameter was fixed at the median of the pairwise distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ (i.e., the median trick) as done in Gretton et al. [2012]. The following regression methods were applied to the same datasets:

- *Kernel ridge regression (KRR)*: The kernel model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ was estimated under the squared-loss with the RKHS norm regularization.
- *Least absolute deviations (LAD)*: The absolute deviation (i.e., $|y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)|$) was used as the loss function with same regularization as KRR.
- *Huber loss (Huber)*: Huber loss was employed to estimate the kernel model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$.
- *Variational heteroscedastic Gaussian process regression (VHGPR)* [Lázaro-Gredilla and Titsias, 2011].
- *Modal regression with kernel density estimation (MR_{KDE})*: A variant of DMR-K with kernel density estimation following the naive two-step approach.
- *Direct modal regression with kernels (DMR-K)*: A proposed method based on reproducing kernels.

Details of the regression methods such as model selection and optimization methods are described in Section D of the supplementary material.

We sampled input data $\boldsymbol{x}_i$ from the uniform density on $[-1, 1]^{d_{\mathrm{x}}}$, and generated the output data y_i according to the generative model (2) where the following two functions were used for the conditional mode function f^*:

(M1) $f^*(\boldsymbol{x}) = \sin\left(\frac{\pi}{d_{\mathrm{x}}} \sum_{j=1}^{d_{\mathrm{x}}} |x^{(j)}|\right)$ where $x^{(j)}$ denotes the j-th element in $\boldsymbol{x}$.

(M2) $f^*(\boldsymbol{x}) = \frac{1}{d_{\mathrm{x}}} \sum_{j=1}^{d_{\mathrm{x}}} (x^{(j)})^2$.

The generated noise ϵ was as follows:

- *Gaussian noise*: ϵ was sampled from the Gaussian density with mean 0 and variance 0.5.
- *Outlier noise*: 90% of noises were sampled from the Gaussian density with mean 0 and variance 0.5, while the remainings were drawn from the uniform density on $[1, 5]$
- *Skewed noise*: ϵ was sampled from the exponential density with mean 0.5.
- *Nonstationary noise*: $\epsilon(\boldsymbol{x}) = |\cos(\pi x^{(1)})| \times \zeta$ where ζ was drawn from the exponential density with mean 0.5. Note that this noise is also skewed.

The total number of samples was $n = 500$. The estimation error was measured by $\frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} |\widehat{y}_i^{\mathrm{te}} - f^*(\boldsymbol{x}_i^{\mathrm{te}})|$, where n_{te} denotes the number of test samples, $\boldsymbol{x}_i^{\mathrm{te}}$ is a test sample generated in the same way as the training samples, and $\widehat{y}_i^{\mathrm{te}}$ is the predicted output by each method from $\boldsymbol{x}_i^{\mathrm{te}}$. We set $n_{\mathrm{te}} = 100,000$ in this illustration.

Table 1 shows the averages of estimation errors in $d_{\mathrm{x}} = 1, 5, 10$. KRR or VHGPR achieves the best performance for the Gaussian noise because these methods make use

Table 1: Averages of estimation errors over 30 runs. The top and bottom panels correspond to when $f^*(\boldsymbol{x})$ is (M1) and (M2), respectively. The numbers in parentheses indicate standard deviations. The best and comparable methods judged by the t-test at the significance level 1% are described in boldface.

(M1)	KRR	LAD	VHGPR	Huber	MR_{KDE}	DMR-K
Gauss noise						
$d_{\text{x}} = 1$	**0.07(0.01)**	0.08(0.01)	**0.06(0.01)**	0.07(0.01)	0.12(0.02)	0.08(0.03)
$d_{\text{x}} = 5$	0.10(0.01)	0.12(0.01)	**0.08(0.01)**	0.10(0.01)	0.19(0.01)	**0.09(0.03)**
$d_{\text{x}} = 10$	0.10(0.01)	0.13(0.02)	**0.08(0.03)**	0.10(0.01)	0.29(0.07)	**0.09(0.05)**
Outlier noise						
$d_{\text{x}} = 1$	0.31(0.03)	**0.10(0.02)**	0.18(0.11)	**0.10(0.02)**	0.11(0.03)	**0.09(0.03)**
$d_{\text{x}} = 5$	0.29(0.02)	0.13(0.02)	0.19(0.10)	0.13(0.02)	0.20(0.02)	**0.10(0.02)**
$d_{\text{x}} = 10$	0.30(0.03)	0.14(0.02)	0.17(0.14)	0.14(0.02)	0.29(0.05)	**0.08(0.03)**
Skewed noise						
$d_{\text{x}} = 1$	0.49(0.02)	0.35(0.02)	0.49(0.03)	0.36(0.02)	**0.21(0.04)**	0.27(0.02)
$d_{\text{x}} = 5$	0.49(0.03)	0.37(0.03)	0.46(0.05)	0.39(0.02)	0.27(0.03)	**0.18(0.03)**
$d_{\text{x}} = 10$	0.49(0.02)	0.36(0.03)	0.42(0.07)	0.39(0.04)	0.33(0.04)	**0.16(0.04)**
Nonstationary noise						
$d_{\text{x}} = 1$	0.31(0.02)	0.22(0.02)	0.31(0.02)	0.22(0.02)	**0.20(0.02)**	0.23(0.02)
$d_{\text{x}} = 5$	0.31(0.02)	0.21(0.02)	0.28(0.03)	0.21(0.02)	0.15(0.01)	**0.11(0.01)**
$d_{\text{x}} = 10$	0.31(0.02)	0.20(0.02)	0.26(0.06)	0.21(0.02)	0.19(0.02)	**0.09(0.02)**

(M2)	KRR	LAD	VHGPR	Huber	MR_{KDE}	DMR-K
Gauss noise						
$d_{\text{x}} = 1$	**0.05(0.01)**	0.06(0.02)	**0.04(0.01)**	0.05(0.02)	0.10(0.02)	0.06(0.03)
$d_{\text{x}} = 5$	**0.09(0.01)**	0.10(0.02)	**0.09(0.01)**	**0.09(0.01)**	0.19(0.01)	**0.10(0.02)**
$d_{\text{x}} = 10$	**0.11(0.01)**	0.13(0.01)	**0.10(0.02)**	0.11(0.01)	0.29(0.07)	**0.10(0.03)**
Outlier noise						
$d_{\text{x}} = 1$	0.31(0.03)	0.09(0.02)	0.26(0.07)	**0.08(0.02)**	0.10(0.02)	**0.07(0.03)**
$d_{\text{x}} = 5$	0.30(0.02)	**0.12(0.02)**	0.19(0.07)	**0.12(0.02)**	0.20(0.02)	**0.11(0.02)**
$d_{\text{x}} = 10$	0.31(0.03)	0.15(0.02)	**0.15(0.11)**	0.15(0.02)	0.29(0.04)	**0.11(0.03)**
Skewed noise						
$d_{\text{x}} = 1$	0.49(0.02)	0.35(0.02)	0.49(0.02)	0.36(0.03)	**0.21(0.05)**	**0.20(0.02)**
$d_{\text{x}} = 5$	0.50(0.02)	0.37(0.03)	0.48(0.04)	0.38(0.03)	0.27(0.03)	**0.21(0.02)**
$d_{\text{x}} = 10$	0.49(0.02)	0.36(0.03)	0.43(0.06)	0.38(0.04)	0.33(0.04)	**0.17(0.03)**
Nonstationary noise						
$d_{\text{x}} = 1$	0.31(0.02)	0.22(0.02)	0.31(0.02)	0.22(0.02)	0.17(0.04)	**0.13(0.01)**
$d_{\text{x}} = 5$	0.32(0.02)	0.20(0.02)	0.30(0.03)	0.20(0.01)	0.15(0.01)	**0.12(0.01)**
$d_{\text{x}} = 10$	0.31(0.02)	0.21(0.02)	0.28(0.04)	0.22(0.02)	0.19(0.02)	**0.11(0.01)**

of the Gaussian assumption, while KRR performs poorly to the other noises. LAD and Huber do not work to the skewed noise because LAD and Huber implicitly assume symmetric noises. The performance of MR_{KDE} is good to the skewed noise in $d_{\text{x}} = 1$, but the performance is worsened as the data dimension d_{x} is increased. Regarding the nonstationary noise, VHGPR shows rather worse performance than other methods. This would be because the noise is nonstationary but skewed, and VHGPR assumes a nonstationary Gaussian noise. Overall, DMR-K works best or is comparable to the best on a wide-range of data dimensions and noises. These results substantiate that our direct approach is suitable. More results can be seen in Section F of the supplementary material.

5.2 Illustration on benchmark datasets

Finally, we investigate the practical performance of DMR-NN and DMR-K on benchmark datasets downloaded from the web [Bache and Lichman, 2013, Chang and Lin, 2011]. Each dataset was randomly divided into training (80%) and test (20%) data samples. Each data sample was standardized by the empirical means and standard deviations of the training samples.

We trained a neural network $f_{\text{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ to predict the output variable by least squares (LS), least absolute de-

Table 2: Averages of the performance score (27) over 20 runs. The numbers in parentheses indicate standard deviations. The best and comparable methods judged by the t-test at the significance level 5% are described in boldface. Note that larger numbers indicate better results. The symbol "-" means that we could not perform DMR-K even with an approximative method because of limited computer memories.

LS	LAD	DMR-NN	DMR-K
space-ga ($d_{\mathrm{x}} = 6, n = 3107$)			
0.81(0.20)	**0.95(0.17)**	**0.97(0.21)**	0.61(0.19)
abalone ($d_{\mathrm{x}} = 8, n = 4177$)			
0.87(0.16)	**0.91(0.25)**	**0.95(0.19)**	0.69(0.19)
cpusmall ($d_{\mathrm{x}} = 12, n = 8192$)			
3.48(0.28)	**3.90(0.33)**	3.51(0.36)	2.80(0.28)
cadata ($d_{\mathrm{x}} = 8, n = 20640$)			
1.31(0.08)	**1.56(0.09)**	**1.62(0.12)**	0.78(0.08)
energy ($d_{\mathrm{x}} = 24, n = 19735$)			
1.20(0.11)	2.64(0.19)	**2.81(0.13)**	2.30(0.20)
superconductivty ($d_{\mathrm{x}} = 81, n = 21263$)			
3.35(0.25)	5.01(0.31)	**5.28(0.48)**	1.35(0.32)
slice loc. ($d_{\mathrm{x}} = 384, n = 53500$)			
14.56(1.07)	20.70(0.60)	**24.55(1.29)**	-
sgemm ($d_{\mathrm{x}} = 14, n = 241600$)			
10.82(0.74)	**14.01(0.99)**	12.75(0.80)	1.93(0.10)
yearpred. ($d_{\mathrm{x}} = 90, n = 515345$)			
0.76(0.02)	**0.93(0.02)**	**0.93(0.09)**	0.36(0.31)

viation (LAD), and DMR-NN. $f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ in all methods was modelled by a three-layer feedforward neural network where the numbers of hidden units were $2d_{\mathrm{x}}$ and d_{x}, and the activation functions were all ReLU. Regarding the log-density derivative estimator in DMR-NN, $\mu_k^{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\gamma}')$ in $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$ were also modelled by a three layer neural network: The numbers of two hidden units were $2K$ and K, and the activation function was the sigmoid function. σ_k in $r_{\mathrm{NN}}(y, \boldsymbol{x}; \boldsymbol{\gamma})$ were selected from 1 to 10 at the regular interval in logarithmic scale, and we set $K = 50$ in $d_{\mathrm{x}} < 30$ otherwise $K = 100$. All parameters were optimized by Adam [Kingma and Ba, 2015] for 500 epochs and regularized with weight decay where the regularization parameter was 10^{-4}. For DMR-NN, we performed pretraining for $f_{\mathrm{NN}}(\boldsymbol{x})$ by LAD. As a kernel-based method, we only applied DMR-K to the same benchmark datasets because it mostly performed best or comparable to the best method in Section 5.1. Unlike the previous experiments on artificial data, we implemented DMR-K with the Nyström approximation [Williams and Seeger, 2001] to reduce the computational costs and memories where only 500 randomly chosen data samples were used as the center points in the kernel functions.

For this illustration, the performance score is important. Here, we used the following score:

$$\frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i^{\mathrm{te}} - \widehat{f}_{\mathrm{NN}}(\boldsymbol{x}_i^{\mathrm{te}}))^2}{2\sigma^2} \right), \quad (27)$$

where σ is the width parameter, n_{te} denotes the number of test samples, y_i^{te} and $\boldsymbol{x}_i^{\mathrm{te}}$ are test samples for input and output data respectively, and $\widehat{f}_{\mathrm{NN}}$ is an estimated neural network by each method. As reviewed in Section 2, (27) is a special case of the surrogate empirical risk $\widetilde{\mathcal{R}}^{\sigma}$ (i.e., $\psi(t) = \exp(-t^2/2)/\sqrt{2\pi}$ in (7)), and approaches to the (non-log) modal regression risk as $n_{\mathrm{te}} \to \infty$ and $\sigma \to 0$ [Feng et al., 2017]. Here, we set $\sigma = n_{\mathrm{te}}^{-1/5}$, which is proved to minimize an upper bound of the excess risk in modal regression [Feng et al., 2017, Proof of Theorem 17]. To support that this choice of σ is fairly good, other results with smaller and larger values of σ are presented in Section G of the supplementary material.

Table 2 shows that DMR-NN works often better than or comparable to LAD, while LS and DMR-K perform poorly. A possible reason of the poor performance of DMR-K is that the Nyström approximation might reduce the function approximation ability. Thus, our method based on neural networks is promising for high-dimensional and large datasets.

6 Conclusion

In this paper, we proposed two modal regression methods based on kernels and neural networks. The key idea is to directly approximate the gradient of the empirical modal regression risk. To this end, we developed direct estimators for the logarithmic derivative of the joint density. For the kernel-based modal regression method, the novel parameter update rule was derived based on a fixed-point method, and some sufficient conditions were given for the monotonic ascending property. Since our direct approach is well-compatible with recent sophisticated stochastic gradient methods, a modal regression method based on neural networks was also developed, and to the best of our knowledge, this work is the first attempt to apply neural networks for modal regression. The superior performance of the proposed methods was empirically demonstrated on various artificial and benchmark datasets.

Acknowledgement

The authors would like to thank Dr. Takashi Takenouchi for his helpful discussion. H.S. was partially supported by JSPS KAKENHI Grant Number 18K18107. T.K. was partially supported by JSPS KAKENHI Grant Numbers 17H00764, 19H04071, and 20H00576.

References

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml/`.

C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Y.-C. Chen. Modal regression using kernel density estimation: A review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(4):e1431, 2018.

G. Collomb, W. Härdle, and S. Hassani. A note on prediction via estimation of the conditional mode function. *Journal of Statistical Planning and Inference*, 15:227–236, 1986.

D. D. Cox. A penalty method for nonparametric estimation of the logarithmic derivative of a density function. *Annals of the Institute of Statistical Mathematics*, 37 (1):271–288, 1985.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12: 2121–2159, 2011.

Y. Feng, J. Fan, and J. A. Suykens. A statistical learning approach to modal regression. *arXiv:1702.05960*, 2017.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

A. Gunduz and J. C. Principe. Correntropy as a novel measure for nonlinearity tests. *Signal Processing*, 89 (1):14–23, 2009.

P. J. Huber and E. M. Ronchetti. *Robust statistics*. Wiley, 2009.

K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic Gaussian process regression. In *Proceedings of the 24th international conference on machine learning (ICML)*, pages 393–400, 2007.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pages 1–15, 2015.

M. Lázaro-Gredilla and M. K. Titsias. Variational heteroscedastic Gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML)*, pages 841–848, 2011.

M.-J. Lee. Mode regression. *Journal of Econometrics*, 42(3):337–349, 1989.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems (NeurIPS)*, pages 1177–1184, 2008.

T. W. Sager and R. A. Thisted. Maximum likelihood estimation of isotonic modal regression. *The Annals of Statistics*, 10(3):690–707, 1982.

H. Sasaki, A. Hyvärinen, and M. Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Machine Learning and Knowledge Discovery in Databases Part III- European Conference, ECML/PKDD 2014*, volume 8726, pages 19–34, 2014.

H. Sasaki, Y. Ono, and M. Sugiyama. Modal regression via direct log-density gradient estimation. In *Proceedings of the 23th International Conference on Neural Information Processing (ICONIP)*, volume 9948, pages 108–116. Springer, 2016.

H. Sasaki, T. Kanamori, A. Hyvärinen, G. Niu, and M. Sugiyama. Mode-seeking clustering and density ridge estimation via direct estimation of density-derivative-ratios. *Journal of machine learning research*, 18(180), 2018.

G. Strang. *Calculus*. Wellesley-Cambridge Press, 1991.

G. Wahba. *Spline models for observational data*, volume 59. SIAM, 1990.

X. Wang, H. Chen, W. Cai, D. Shen, and H. Huang. Regularized modal regression with applications in cognitive impairment prediction. In *Advances in neural information processing systems (NIPS)*, pages 1448–1458, 2017.

Y. Wang, Y. Y. Tang, L. Li, and H. Chen. Modal regression-based atomic representation for robust face recognition and reconstruction. *IEEE Transactions on Cybernetics*, pages 1–13, 2019.

C. K. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Iormation Processing Systems (NeurIPS)*, pages 682–688, 2001.

W. Yao and L. Li. A new regression model: modal linear regression. *Scandinavian Journal of Statistics*, 41(3): 656–671, 2014.

W. Yao, B. G. Lindsay, and R. Li. Local modal regression. *Journal of nonparametric statistics*, 24(3):647–663, 2012.