Noname manuscript No.
(will be inserted by the editor)

An Efficient Computation of Conditional Kernel Mean Embedding using K-nearest Neighbor Samples

Authors

Received: date / Accepted: date

Abstract In this paper, we consider

Keywords kernel methods · conditional kernel mean embedding · k-nearest neighbors

1 Introduction

Let $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$ be a measurable space, and $\mathcal{M}_1(\mathcal{X})$ be the set of all probability distributions on $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a measurable positive-definite (p.d.) kernel, and $\mathcal{H}$ be the associated reproducing kernel Hilbert space (RKHS) where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product.

In machine learning, kernel methods map *data* $X_1, \ldots, X_n \in \mathcal{X}$ into RKHS $\mathcal{H}$ as RKHS elements $k(\cdot, X_1), \ldots, k(\cdot, X_n) \in \mathcal{H}$[1], and solve problems in the space, by taking advantange of the kernel trick (Schölkopf and Smola, 2002; Steinwart and Christmann, 2008).

In recent kernel methods, as a natural extension, *probability distributions* $P \in \mathcal{M}_1(\mathcal{X})$ are mapped into the RKHS and mapped elements $m_P \in \mathcal{H}$ are handled to operate probability distributions in the space.

Smola et al (2007) introduced *kernel mean embedding* (KME):

$$m_P := \int k(\cdot, x) dP(x) = \mathbb{E}_{X \sim P}[k(\cdot, X)] \in \mathcal{H},$$

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

[1] $k(\cdot, x) \in \mathcal{H}$ denote the RKHS function as a function of $(\cdot)$ with fixed $x \in \mathcal{X}$.

which is defined as the expectation of the random RKHS function $k(\cdot, X)$.[2] Muandet et al (2017) includes extensive surveys on the KME literature.

Fukumizu et al (2004); Sriperumbudur et al (2010) introduced that, if a p.d. kernel k is *characteristic*, then the RKHS $\mathcal{H}$ is large enough, and kernel mean mapping $P \mapsto m_P$ becomes injective. Examples of characteristic kernels include Gaussian kernels and Laplace kernels, frequently used in machine learning, though polynomial kernels are not. Characteristic kernels ensure that similarity between two probabilities $P, Q \in \mathcal{M}_1(\mathcal{X})$ and distance between two probabilities $P, Q \in \mathcal{M}_1(\mathcal{X})$ can be defined via the RKHS inner product $Sim(P, Q) = \langle m_P, m_Q \rangle_{\mathcal{H}}$ and RKHS norm $d(P, Q) = \|m_P - m_Q\|_{\mathcal{H}}$, which are fundamental computation required when comparing two probabilities in machine learning.

Song et al (2009, 2013) introduced *conditional kernel mean embedding* (CKME), which is the KME of conditional distributions. Grünewälder et al (2012a) showed that the nonparametric CKME estimator developed by Song et al (2009, 2013) is equivalent to the solution to the (RKHS-valued) kernel ridge regression. The development of the nonparametric CKME estimator (Song et al, 2009, 2013) has driven various nonparametric inference in the KME form, including kernel Bayes' rule (Fukumizu et al, 2013; Song et al, 2013, 2016), filtering on state space models (Fukumizu et al, 2013; Kanagawa et al, 2014; Nishiyama et al, 2018), smoothing on state space models (Nishiyama et al, 2016), reinforcement learning (Grünewälder et al, 2012b; Nishiyama et al, 2012; Rawlik et al, 2013; Boots et al, 2013), and so on.

In this paper, we focus on computing such an important CKME estimator (Song et al, 2009, 2013). The naive computation of the CKME estimator (Song et al, 2009, 2013) requires $n \times n$ matrix multiplication and inversion, which results in computational cost $O(n^3)$, where n is the training sample size. For large n, memory to store large matrices and time complexity $O(n^3)$ become severe. This issue is common in the abovementioned KME-based applications.

In this paper, to manage the problem, we propose an effiicent and approximate algorithm to compute the nonparametric CKME estimator (Song et al, 2009, 2013). We take a simple approach. We approximately compute CKME estimator only using local training samples $(X_j, Y_j)_{j=1}^m$ ($m \ll n$) close to the input, which will dominantly affect the CKME estimator. We select m local training samples by using m nearest neighbours (mNN), which ensures small computational cost $O(m^3)$ ($m \ll n$) for each matrix multiplication and inversion. Since a p.d. kernel k is endowed in the input space, we can use the p.d. kernel k to measure similarity between inputs. We refer to the proposed algorithm as *localized* CKME (LCKME).

We report that the

The remainder of this paper is structured as follows. In the next section, we

[2] In the KME, mapping data $X_1, \ldots, X_n \in \mathcal{X}$ into RKHS $\mathcal{H}$ as feature functions $k(\cdot, X_1), \ldots, k(\cdot, X_n) \in \mathcal{H}$ is rephrased as mapping delta distributions $\delta_{X_1}, \ldots, \delta_{X_n} \in \mathcal{M}_1(\mathcal{X})$ into RKHS $\mathcal{H}$ as kernel means $m_{\delta_1}, \ldots, m_{\delta_n} \in \mathcal{H}$.

2 Preliminaries

2.1 Conditional Kernel Mean Embedding (CKME)

Let $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ and $(\mathcal{Y}, \mathcal{B}_\mathcal{Y})$ be measurable spaces. Let $k_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_\mathcal{Y} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be measurable p.d. kernels. Let $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$ be the associated RKHSs with inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{X}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{Y}}$, respectively. CKME of conditional distribution $P_{\mathcal{Y}|x}$ on $\mathcal{Y}$ at fixed $x \in \mathcal{X}$ is defined by

$$m_{\mathcal{Y}|x} := \mathbb{E}_{Y \sim P_{\mathcal{Y}|x}}[k_\mathcal{Y}(\cdot, Y) \mid X = x] \in \mathcal{H}_\mathcal{Y}, \quad \forall x \in \mathcal{X},$$

Given n training data $\{(X_i, Y_i)\}_{i=1}^n \overset{iid}{\sim} P_{\mathcal{X}\times\mathcal{Y}}$ and a query input $x \in \mathcal{X}$, Song et al (2009, 2013) proposed a nonparametric estimator:

$$\hat{m}_{\mathcal{Y}|x} := \sum_{i=1}^n w_i k_\mathcal{Y}(\cdot, Y_i), \quad w := (G_X + n\varepsilon_n I_n)^{-1} \mathbf{k}_\mathcal{X}(x) \in \mathbb{R}^n, \tag{1}$$

where $G_X := (k_\mathcal{X}(X_i, X_j))_{ij} \in \mathbb{R}^{n\times n}$ is the Gram matrix on the input space $\mathcal{X}$, $\mathbf{k}_\mathcal{X}(x) := (k_\mathcal{X}(X_i, x))_i \in \mathbb{R}^{n\times 1}$ is the similarity vector between the query input x and training data on $\mathcal{X}$, $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix, and $\varepsilon_n > 0$ is a regularization parameter. Under $\varepsilon_n \to 0$ $(n \to \infty)$ with appropriate conditions, the consistency $\|m_{\mathcal{Y}|x} - \hat{m}_{\mathcal{Y}|x}\|_{\mathcal{H}_\mathcal{Y}} \xrightarrow{p} 0$ is shown (Fukumizu et al, 2013, Theorem 8). Grünewälder et al (2012a) showed that the weight vector w is equivalent to the solution to the (RKHS-valued) kernel ridge regression. In addition, the same estimator w is known in the context of structured prediction under the name of *kernel dependency estimation*, although the connection to CKME was not known at the time (Weston et al, 2003; Cortes et al, 2005).

The CKME estimator (1) can be used to compute expected values and mode estimation as follows:

- **Expected value**: The expected value of an RKHS function $f \in \mathcal{H}_\mathcal{Y}$ with respect to a conditional distribution $P_{\mathcal{Y}|x}$ can be computed by using the CKME estimator (1) as follows:

$$\begin{aligned}\mathbb{E}_{Y\sim P_{\mathcal{Y}|x}}[f(Y)] &= \mathbb{E}_{Y\sim P_{\mathcal{Y}|x}}[\langle k_\mathcal{Y}(Y, \cdot), f\rangle_{\mathcal{H}_\mathcal{Y}}] = \left\langle m_{P_{\mathcal{Y}|x}}, f \right\rangle_{\mathcal{H}_\mathcal{Y}} \\ &\approx \left\langle \hat{m}_{P_{\mathcal{Y}|x}}, f \right\rangle_{\mathcal{H}_\mathcal{Y}} = \sum_{i=1}^n w_i f(Y_i), \quad \forall f \in \mathcal{H}_\mathcal{Y}. \end{aligned} \tag{2}$$

  The first equality holds due to the kernel trick, the second equality holds due to the exchangeability between the expectation operator and inner product, and the fourth equality is obtained by plugging the CKME estimator (1) and using the kernel trick. Thus, the expected value can be computed by a weight sum of function f using CKME weight vector w.
- **Mode estimation**: A mode of conditional distribution $P_{\mathcal{Y}|x}$ is estimated by solving a pre-image problem (Mika et al, 1999; Fukumizu et al, 2013):

$$\hat{y}^{(\text{mode})} := \arg\min_y \|k_\mathcal{Y}(\cdot, y) - \hat{m}_{P_{\mathcal{Y}|x}}\|_{\mathcal{H}_\mathcal{Y}}.$$

If a Gaussian kernel is used for $k_{\mathcal{Y}}$, then the following fixed-point algorithm is known to solve the optimization problem:

$$y^{(t+1)} = \frac{\sum_{i=1}^{n} Y_i w_i k_{\mathcal{Y}}(Y_i, y^{(t)})}{\sum_{i=1}^{n} w_i k_{\mathcal{Y}}(Y_i, y^{(t)})} \quad (t = 0, 1, 2, \dots). \tag{3}$$

2.2 Computational Issue for Computing the CKME Estimator

The following are some computational issues in computing CKME.

- Naive computation of weight vector $w \in \mathbb{R}^n$ in equation (1) costs $O(n^3)$ since there exists a matrix inversion $(G_X + n\varepsilon_n I_n)^{-1} \in \mathbb{R}^{n\times n}$. For large n, memory to store the large matrix and time complexity $O(n^3)$ become severe.
- In addition, if one wants to compute multiple CKMEs $\hat{m}_{\mathcal{Y}|x_1}, \dots, \hat{m}_{\mathcal{Y}|x_l}$ with a set of inputs $U = \{x_1, \dots, x_l\}$, then one needs to compute a weight matrix:
$$W = (w_1, \dots, w_l) := (G_X + n\varepsilon_n I_n)^{-1} G_{XU} \in \mathbb{R}^{n\times l},$$
where $G_{XU} := (k_{\mathcal{X}}(X_i, x_j))_{ij} \in \mathbb{R}^{n\times l}$. Naive computation of matrix W requires a matrix multiplication between $(G_X + n\varepsilon_n I_n)^{-1} \in \mathbb{R}^{n\times n}$ and $G_{XU} \in \mathbb{R}^{n\times l}$.
- Further, it is common in kernel methods that hyperparameters (kernel parameters and regularization parameter ε_n) are tuned by using cross validation (CV) with a grid search. Matrix $(G_X + n\varepsilon_n I_n)^{-1}$ depends on the hyperparameter values and one requires to recompute the matrix with different hyperparameter setting many times.[3]

The above issues commonly arise in CKME applications, as given in Introduction.

3 Proposed Algorithm: Localized CKME (LCKME)

To manage the computational issues of the CKME estimator (1) raised in the last section, we propose an efficient and approximate algorithm to compute the CKME estimator (1). We take a simple approach. We approximately compute CKME estimator (1) only using m local training samples $(X_j, Y_j)_{j=1}^{m} \subset (X_i, Y_i)_{i=1}^{n}$ $(m \ll n)$ close to the input $x \in \mathcal{X}$, which will dominantly contribute to form the CKME estimator (1). To measure closeness and similarity between inputs, we use the p.d. kernel $k_{\mathcal{X}}$ endowed in the input space.

The proposed algorithm is given below:

[3] For example, if Gaussian kernel k_σ with a band-width parameter σ is used, then matrix $(G_X + n\varepsilon_n I_n)^{-1}$ depends on σ and ε_n. In kernel methods, optimal hyperparameters σ^* and ε_n^* are searched, e.g., in a grid search, which requires repeated computation of the inverse matrix.

Algorithm 1 Localized CKME (LCKME) Algorithm

```
**Input:** Determine parameter $m$. Give a query input $x \in \mathcal{X}$.
**Compute:** Compute a similarity vector $\mathbf{k}_{\mathcal{X}}(x) \in \mathbb{R}^n$, and sort the similarity vector in descending order as $\tilde{\mathbf{k}}_{\mathcal{X}}(x)$.
**Compute:** Pick up the $m$ upper sub-vector $\tilde{\mathbf{k}}_{\mathcal{X}}^{(m)}(x)$ and compute the Gram matrix $\tilde{G}_X^{(m)}$ among the $m$ closest training subset.
**Output:** Compute an approximate weight vector $\tilde{w} \in \mathbb{R}^m$ according to equation (4).
```

- **Step1**: Given a query input $x \in \mathcal{X}$, find the m closest training subset $(X_j, Y_j)_{j=1}^m$ from the training samples $(X_i, Y_i)_{i=1}^n$. This is achieved by computing a similarity vector $\mathbf{k}_{\mathcal{X}}(x) := (k_{\mathcal{X}}(X_i, x))_i \in \mathbb{R}^{n \times 1}$, sorting the vector in descending order $\tilde{\mathbf{k}}_{\mathcal{X}}(x)$, and pick up the m upper sub-vector $\tilde{\mathbf{k}}_{\mathcal{X}}^{(m)}(x)$.
- **Step2**: We then compute an approximate CKME estimator by

$$\hat{m}_{\mathcal{Y}|x} := \sum_{j=1}^{m} \tilde{w}_j k_{\mathcal{Y}}(\cdot, Y_j), \quad \tilde{w} := (\tilde{G}_X^{(m)} + m\varepsilon_m I_m)^{-1} \tilde{\mathbf{k}}_{\mathcal{X}}^{(m)}(x) \in \mathbb{R}^m, \quad (4)$$

  where $\tilde{G}_X^{(m)} := (k_{\mathcal{X}}(X_i, X_j))_{ij} \in \mathbb{R}^{m \times m}$ is the Gram matrix among the m closest training subset $(X_j, Y_j)_{j=1}^m$.

Algorithm 1 shows the psudo-code. We refer to the proposed Algorithm 1 as *localized* CKME (LCKME).

Remarks on computation is as follows.

- Memory to store the similarity vector $\mathbf{k}_{\mathcal{X}}(x) \in \mathbb{R}^n$ is $O(n)$. Given an n dimensional vector, various sorting algorithm can be applied. Quick sort costs average time complexity $O(n \log n)$ and worst-case auxiliary space memory $O(n)$.
- Memory to store the sub-vector $\tilde{\mathbf{k}}_{\mathcal{X}}^{(m)}(x)$ is $O(m)$. Computation of the inverse matrix $(\tilde{G}_X^{(m)} + m\varepsilon_m I_m)^{-1}$ costs $O(m^3)$, and its space memory is $O(m^2)$. Multiplying a vector $\tilde{\mathbf{k}}_{\mathcal{X}}^{(m)}(x)$ by the matrix $(\tilde{G}_X^{(m)} + m\varepsilon_m I_m)^{-1}$ costs $O(m^2)$. Memory to store the weight vector $\tilde{w}$ is $O(m)$.

4 Related Works

Another way of approximating the CKME estimator (1) is based on ICF.

4.1 Incomplete Cholesky Factorization (ICF)

Consider an r-rank ($r < n$) approximation of the Gram matrix $G_X \in \mathbb{R}^{n \times n}$ such that $G_X \approx LL^\top$, where $L \in \mathbb{R}^{n \times r}$ is the n by r matrix. Such a low rank matrix $LL^\top$ can be obtained by ICF with time complexity $O(nr^2)$ (Fine and Scheinberg, 2001; Bach and Jordan, 2002). Given matrix L, the weight vector w in the CKME estimator (1) is approximated as

$$w = (G_X + n\varepsilon_n I_n)^{-1}\mathbf{k}_{\mathcal{X}}(x) \approx (LL^\top + n\varepsilon_n I_n)^{-1}\mathbf{k}_{\mathcal{X}}(x)$$
$$= \frac{1}{n\varepsilon_n}(I_n - L(n\varepsilon_n I_r + L^\top L)^{-1}L^\top)\mathbf{k}_{\mathcal{X}}(x), \tag{5}$$

where the third equality holds due to the Woodbury identity.

The following are computational remarks

- Computing inverse matrix $(n\varepsilon_n I_r + L^\top L)^{-1}$ costs time complexity $O(r^3)$, and space memory to store matrix L is $O(nr)$. Multiplying a vector $\mathbf{k}_{\mathcal{X}}(x)$ by the matrix $L^\top$ costs $O(nr)$. Memory to store the weight vector w is $O(n)$.
- Equation (5) requires space memory $O(nr)$ to store matrix L or $O(n^2)$ to store matrix $L(n\varepsilon_n I_r + L^\top L)^{-1}L^\top \in \mathbb{R}^{n\times n}$. For large n, the space memory will become severe. On the other hand, proposed KCKME estimator (4) stores only $O(m^2)$ and can respond to large n.

4.2 Random Sub-sampling

As a naive baseline, one can consider randomly sub-sampling m points to select training subset $(X_j, Y_j)_{j=1}^m$ from the training data $(X_i, Y_i)_{i=1}^n$ (Cortes et al, 2005).

4.3 Nadaraya-Watson kernel regression (NWKR)

As a naive baseline, one can consider the NWKR estimator:

$$\hat{m}_{\mathcal{Y}|x} := \sum_{i=1}^n w_i k_{\mathcal{Y}}(\cdot, Y_i), \quad w := \frac{\mathbf{k}_{\mathcal{X}}(x)}{\|\mathbf{k}_{\mathcal{X}}(x)\|_1} \in \mathbb{R}^n,$$

where the inverse matrix $(G_X + n\varepsilon_n I_n)^{-1}$ is dropped off from the CKME estimator (1), and the similarity vector $\mathbf{k}_{\mathcal{X}}(x)$ is normalized to sum up to 1.

4.4 Random Fourier Features (RFF)

Rahimi and Recht (2007) proposed random Fourier features (RFF) to scale up kernel machine algorithms for a shift-invariant kernel k. Bochner's Theorem (Bochner, 1959) guarantees that a shift invariant kernel $k(\cdot,\cdot)$ is expressed as

$$k(x_1, x_2) = \int_{\mathbb{R}^d} e^{-2\pi\sqrt{-1}\eta^\top(x_1-x_2)}p(\eta)d\eta = \mathbb{E}_{\eta\sim p}[e^{-2\pi\sqrt{-1}\eta^\top(x_1-x_2)}],$$

where $p(\eta)$ is the associated probability density function determined by a shift invariant $k(\cdot,\cdot)$.

If $\eta_1, \ldots, \eta_m \in \mathbb{R}^d$ are i.i.d. samples drawn from $p(\cdot)$ and we define

$$\varphi(x) := \frac{1}{\sqrt{m}} \left(e^{-2\pi\sqrt{-1}\eta_1^\top x}, \ldots, e^{-2\pi\sqrt{-1}\eta_m^\top x} \right)^* \in \mathbb{C}^m,$$

then a shift invariant kernel $k(\cdot, \cdot)$ is approximated as

$$k(x_1, x_2) \approx \tilde{k}(x_1, x_2) := \varphi(x_1)^* \varphi(x_2) = \frac{1}{m} \sum_{i=1}^{m} e^{-2\pi\sqrt{-1}\eta_i^\top (x_1 - x_2)}.$$

If we define a matrix $\Phi := (\varphi(x_1), \ldots, \varphi(x_n))^* \in \mathbb{C}^{n \times m}$, then the approximated Gram matrix $\tilde{G}_X := \{\tilde{k}(x_i, x_j)\}_{ij} \in \mathbb{R}^{n \times n}$ is expressed as

$$\tilde{G}_X = \Phi \Phi^*.$$

Then, RFF estimates CKME by the following estimator:

$$\hat{m}_{\mathcal{Y}|x} := \sum_{i=1}^{n} w_i k_{\mathcal{Y}}(\cdot, Y_i), \quad w := \varphi(x)^* (\Phi^* \Phi + m\varepsilon_m I_m)^{-1} \Phi^* \in \mathbb{C}^n,$$

Computation requires $O(nm^2)$ time complexity and $O(m^2)$ memory.

4.5 Divide and Conquer CKME

Zhang et al (2015) proposed a divide and Conquer kernel ridge regression.

4.6 Fast Randomized CKME

Alaoui and Mahoney (2015) proposed a fast randomized kernel ridge regression.

5 Experiments

5.1 Ground-truth Experiment

Let $\mathcal{X} = \mathbb{R}^{d_x}$, $\mathcal{Y} = \mathbb{R}^{d_y}$. Let $\{(X_i, Y_i)\}_{i=1}^n$ be training samples drawn i.i.d. from a Gaussian distribution $N(\mathbf{0}, V)$ with a mean vector $\mathbf{0} \in \mathbb{R}^{d_x + d_y}$ and covariance matrix $V = \begin{pmatrix} V_X & V_{XY} \\ V_{YX} & V_Y \end{pmatrix}$. Let $k_{\mathcal{X}}(x_1, x_2) = d(x_1 | x_2, \sigma_x I_{d_x})$ $(\sigma_x > 0)$ and $k_{\mathcal{Y}}(y_1, y_2) = d(y_1 | y_2, \sigma_y I_{d_y})$ $(\sigma_y > 0)$ be Gaussian kernels, where $d(x | \mu, \Sigma)$ denotes a Gaussian pdf with a mean vector μ and covariance matrix Σ.

We adopt the following two criterions to evaluate the accuracy of obtained CKME estimator $\hat{m}_{P_{\mathcal{Y}|x}}$:

1. **RKHS norm error**: In the above Gaussian setting, the conditional distribution $P_{\mathcal{Y}|x}$ is given with the pdf:

   $$p(y \mid x) = d(y \mid \bar{\mu}(x), \bar{V}), \quad \bar{\mu}(x) := V_{XY}^{\top} V_X^{-1} x, \quad \bar{V} := V_Y - V_{XY}^{\top} V_X^{-1} V_{XY}.$$

   Then the true CKME is given by

   $$m_{P_{\mathcal{Y}|x}}(y) = d(y \mid \bar{\mu}(x), \bar{V} + \sigma_y I_{d_y}).$$

   The RKHS norm error between the true CKME $m_{P_{\mathcal{Y}|x}}$ and a CKME estimator $\hat{m}_{\mathcal{Y}|x} := \sum_{i=1}^{l} w_i k_{\mathcal{Y}}(\cdot, \bar{Y}_i)$ on a sample $\bar{Y} := \{\bar{Y}_1, \ldots, \bar{Y}_l\} \subset \mathcal{Y}$ is given by

   $$\begin{aligned} \left\| m_{P_{\mathcal{Y}|x}} - \hat{m}_{P_{\mathcal{Y}|x}} \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 &= \left\| m_{P_{\mathcal{Y}|x}} \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 - 2\left\langle \hat{m}_{P_{\mathcal{Y}|x}}, m_{P_{\mathcal{Y}|x}} \right\rangle_{\mathcal{H}_{\mathcal{Y}}} + \left\| \hat{m}_{P_{\mathcal{Y}|x}} \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 \\ &= d(\mathbf{0} \mid \mathbf{0}, 2\bar{V} + \sigma_y I_{d_y}) - 2\sum_{i=1}^{l} w_i m_{P_{\mathcal{Y}|x}}(\bar{Y}_i) + w^{\top} G_{\bar{Y}} w, \end{aligned}$$

   where $G_{\bar{Y}} := (k_{\mathcal{Y}}(\bar{Y}_i, \bar{Y}_j))_{ij} \in \mathbb{R}^{l \times l}$ is the Gram matrix among the sample $\bar{Y}$. We use the above equation to compute the RKHS norm error. For T trials evaluation, we compute the averaged RKHS norm error as

   $$\frac{1}{T} \sum_{i=1}^{T} \left\| m_{P_{\mathcal{Y}|x}}^{(i)} - \hat{m}_{P_{\mathcal{Y}|x}}^{(i)} \right\|_{\mathcal{H}_{\mathcal{Y}}},$$

   where $m_{P_{\mathcal{Y}|x}}^{(i)}$ and $\hat{m}_{P_{\mathcal{Y}|x}}^{(i)}$ are the true CKME and its estimator in the i-th trial.
2. **RMSE of a point estimation**: We evaluate the error between a point estimation $\hat{y}$ and true sample y.[4] For T trials evaluation, we compute the RMSE by

   $$\sqrt{\frac{1}{T} \sum_{i=1}^{T} \| y^{(i)} - \hat{y}^{(i)} \|^2}.$$

   We consider two point estimations, mean and mode:
   (a) **Mean:** The mean of a conditional distribution $P_{\mathcal{Y}|x}$ is estimated by plugging $f(y) = y$ into equation (2):

   $$\mathbb{E}_{Y \sim P_{\mathcal{Y}|x}}[Y] \approx \sum_{i=1}^{n} w_i Y_i.$$

   (b) **Mode:** In the above Gaussian kernel setting, the mode $\hat{y}^{(\mathrm{mode})}$ of a conditional distribution $P_{\mathcal{Y}|x}$ is computed by using a fixed-point algorithm (3).

[4] Note that the true sample $y^{(i)}$ is i.i.d. drawn from the pdf $p(y \mid x^{(i)})$ and fluctuated. Hence, even if the true mode $y^{(\mathrm{mode})}$ is successfully estimated, the RMSE error does not become 0.

Fig. 1 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε and σ_x.

Fig. 2 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of σ_x.

Fig. 3 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε.

Fig. 4 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of d_x.

In each trial, we generate a random covariance matrix $V = A^\top A$ ($A \in \mathbb{R}^{d_x+d_y}$) by sampling each entry A_{ij} i.i.d from $N(0,1)$, and generate $x \in \mathbb{R}^{d_x}$ i.i.d. from $N(0, V_X)$. We run $T = 30$ trials.

Fig. 9 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε and σ_x. z-axis shows RKHS norm error, x-axis shows σ_x, and y-axis shows ε.

Fig. 10 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of σ_x. y-axis shows RKHS norm error and x-axis shows σ_x.

Fig. 11 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε. y-axis shows RKHS norm error and x-axis shows ε.

Fig. 12 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of d_x. y-axis shows RKHS norm error and x-axis shows d_x.

Fig. 13 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of training data size n. y-axis shows RKHS norm error and x-axis shows n.

Fig. 14 shows accuracy (averaged RKHS norm error and RMSE) vs. computational time regarding CKME estimators (CKME, LCKME, NWKR, random). y-axis shows RKHS norm error and x-axis shows computational time.

It is often in the kernel methods that kernel parameters (σ_x and ε) are tuned by cross validation (CV) with a grid search. Let $(\sigma_x^{(CKME)}, \varepsilon^{(CKME)})$, $(\sigma_x^{(LCKME)}, \varepsilon^{(LCKME)})$, $(\sigma_x^{(NWKR)}, \varepsilon^{(NWKR)})$, $(\sigma_x^{(random)}, \varepsilon^{(random)})$ be estimated kernel parameters by using CV with a grid search for methods (CKME,

Fig. 5 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of training data size n.

Fig. 6 Accuracy (averaged RKHS norm error and RMSE) vs. computational time regarding CKME estimators (CKME, LCKME, NWKR, random).

Fig. 7 Estimation error of kernel parameters estimated by CV for methods (CKME, LCKME, NWKR, random).

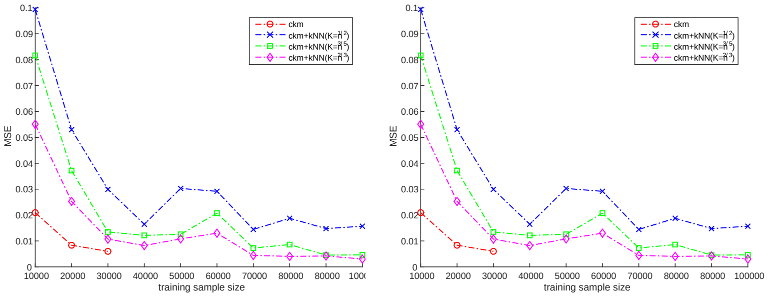

Fig. 8 kNN v.s. Original

LCKME, NWKR, random). we evaluated the parameter estimation error by

$$(\varDelta\sigma_x, \varDelta\varepsilon) = (\sigma_x^{(LCKME)} - \sigma_x^{(CKME)}, \varepsilon^{(LCKME)} - \varepsilon^{(CKME)})$$
$$(\varDelta\sigma_x, \varDelta\varepsilon) = (\sigma_x^{(NWKR)} - \sigma_x^{(CKME)}, \varepsilon^{(NWKR)} - \varepsilon^{(CKME)})$$

Fig. 15 shows estimation error of kernel parameters estimated by CV for methods (CKME, LCKME, NWKR, random). y-axis shows error $\varDelta\sigma_x$ and x-axis shows error $\varDelta\varepsilon$.

$\mathcal{X} = \mathcal{Y} \in \mathbb{R}^2$

– X と Y の次元数 dim は 2
– 訓練サンプルは (X,Y)∼N((0,1),V), V=A’A
  A の要素は N(3, 1) から発生
– 訓練サンプル数 n は 1 万飛ばしで 1 万∼10 万
– 3 万サンプルが計算可能な上限値と仮定
– X と Y のガウスカーネルのバンド幅は 1*eye(dim)
– 正則化パラメータ λ は $0.001/\sqrt{n}$
– 各訓練サンプル数で観測値が異なる 10 回の実験を行い, その平均値をプロットした

5.2 Ground-truth: nonlinear regression function

Let $\mathcal{X} = \mathbb{R}^{d_x}$, $\mathcal{Y} = \mathbb{R}^{d_y}$. Let an additive nonlinear Gaussian noise model be

$$y = f(x) + \epsilon, \quad f(x) = \sum_{i=1}^{c} \gamma_i k(x, U_i), \quad \epsilon \sim N(\mathbf{0}, V), \tag{6}$$

where $k(x, \tilde{x}) = d(x|\tilde{x}, \sigma I_{d_x})$ $(\sigma > 0)$ is a Gaussian kernel, $\gamma = (\gamma_1, \ldots, \gamma_c)^\top \in \mathbb{R}^c$, and $\{U_1, \ldots, U_c\} \subset \mathbb{R}^{d_x}$. Let $\{(X_i, Y_i)\}_{i=1}^n$ be training samples drawn i.i.d. from $X_i \sim \text{Uni}[-1, 1]$ and equation (6).

Let $k_{\mathcal{X}}(x_1, x_2) = d(x_1|x_2, \sigma_x I_{d_x})$ $(\sigma_x > 0)$ and $k_{\mathcal{Y}}(y_1, y_2) = d(y_1|y_2, \sigma_y I_{d_y})$ $(\sigma_y > 0)$ be Gaussian kernels, where $d(x|\mu, \Sigma)$ denotes a Gaussian pdf with a mean vector μ and covariance matrix Σ.

We adopt the following two criterions to evaluate the accuracy of obtained CKME estimator $\hat{m}_{P_{\mathcal{Y}|x}}$:

1. **RKHS norm error**: In the above Gaussian setting, the conditional distribution $P_{\mathcal{Y}|x}$ is given with the pdf:

$$p(y \mid x) = d(y \mid f(x), V).$$

Then the true CKME is given by

$$m_{P_{\mathcal{Y}|x}}(y) = d(y \mid f(x), V + \sigma_y I_{d_y}).$$

The RKHS norm error between the true CKME $m_{P_{\mathcal{Y}|x}}$ and a CKME estimator $\hat{m}_{\mathcal{Y}|x} := \sum_{i=1}^l w_i k_{\mathcal{Y}}(\cdot, \bar{Y}_i)$ on a sample $\bar{Y} := \{\bar{Y}_1, \ldots, \bar{Y}_l\} \subset \mathcal{Y}$ is given by

$$\begin{aligned}\left\| m_{P_{\mathcal{Y}|x}} - \hat{m}_{P_{\mathcal{Y}|x}} \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 &= \left\| m_{P_{\mathcal{Y}|x}} \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 - 2\left\langle \hat{m}_{P_{\mathcal{Y}|x}}, m_{P_{\mathcal{Y}|x}} \right\rangle_{\mathcal{H}_{\mathcal{Y}}} + \left\| \hat{m}_{P_{\mathcal{Y}|x}} \right\|_{\mathcal{H}_{\mathcal{Y}}}^2 \\ &= d(\mathbf{0} \mid \mathbf{0}, 2V + \sigma_y I_{d_y}) - 2 \sum_{i=1}^{l} w_i m_{P_{\mathcal{Y}|x}}(\bar{Y}_i) + w^\top G_{\bar{Y}} w,\end{aligned}$$

where $G_{\bar{Y}} := (k_{\mathcal{Y}}(\bar{Y}_i, \bar{Y}_j))_{ij} \in \mathbb{R}^{l \times l}$ is the Gram matrix among the sample $\bar{Y}$. We use the above equation to compute the RKHS norm error. For T trials evaluation, we compute the averaged RKHS norm error as

$$\frac{1}{T} \sum_{i=1}^{T} \left\| m_{P_{\mathcal{Y}|x}}^{(i)} - \hat{m}_{P_{\mathcal{Y}|x}}^{(i)} \right\|_{\mathcal{H}_{\mathcal{Y}}},$$

where $m_{P_{\mathcal{Y}|x}}^{(i)}$ and $\hat{m}_{P_{\mathcal{Y}|x}}^{(i)}$ are the true CKME and its estimator in the i-th trial.

Fig. 9 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε and σ_x.

Fig. 10 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of σ_x.

Fig. 11 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε.

Fig. 12 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of d_x.

2. **RMSE of mode estimate**: In the above Gaussian kernel setting, mode estimation $\hat{y}^{(\mathrm{mode})}$ of CKME estimator $\hat{m}_{P_{\mathcal{Y}|x}}$ can be computed by using a fixed-point algorithm (3). We evaluate the error between the mode estimate $\hat{y}$ and true sample y.[5] For T trials evaluation, we compute the RMSE of mode estimate by

$$\sqrt{\frac{1}{T}\sum_{i=1}^{T}\|y^{(i)}-\hat{y}^{(i)}\|^2},$$

In each trial, we generate $\gamma_i \sim \mathrm{Unif}[-1,1]$, $U_i \sim \mathrm{Unif}[-1,1]$. We run $T = 30$ trials.

Fig. 9 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε and σ_x. z-axis shows RKHS norm error, x-axis shows σ_x, and y-axis shows ε.

Fig. 10 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of σ_x. y-axis shows RKHS norm error and x-axis shows σ_x.

Fig. 11 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of ε. y-axis shows RKHS norm error and x-axis shows ε.

Fig. 12 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of d_x. y-axis shows RKHS norm error and x-axis shows d_x.

[5] Note that the true sample $y^{(i)}$ is i.i.d. drawn from the pdf $p(y \mid x^{(i)})$ and fluctuated. Hence, even if the true mode $y^{(\mathrm{mode})}$ is successfully estimated, the RMSE error does not become 0.

Fig. 13 Accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of training data size n.

Fig. 14 Accuracy (averaged RKHS norm error and RMSE) vs. computational time regarding CKME estimators (CKME, LCKME, NWKR, random).

Fig. 15 Estimation error of kernel parameters estimated by CV for methods (CKME, LCKME, NWKR, random).

Fig. 13 shows accuracy (averaged RKHS norm error and RMSE) of CKME estimators (CKME, LCKME, NWKR, random) as a function of training data size n. y-axis shows RKHS norm error and x-axis shows n.

Fig. 14 shows accuracy (averaged RKHS norm error and RMSE) vs. computational time regarding CKME estimators (CKME, LCKME, NWKR, random). y-axis shows RKHS norm error and x-axis shows computational time.

It is often in the kernel methods that kernel parameters (σ_x and ε) are tuned by cross validation (CV) with a grid search. Let $(\sigma_x^{(CKME)}, \varepsilon^{(CKME)})$, $(\sigma_x^{(LCKME)}, \varepsilon^{(LCKME)})$, $(\sigma_x^{(NWKR)}, \varepsilon^{(NWKR)})$, $(\sigma_x^{(random)}, \varepsilon^{(random)})$ be estimated kernel parameters by using CV with a grid search for methods (CKME, LCKME, NWKR, random). we evaluated the parameter estimation error by

$$(\varDelta\sigma_x, \varDelta\varepsilon) = (\sigma_x^{(LCKME)} - \sigma_x^{(CKME)}, \varepsilon^{(LCKME)} - \varepsilon^{(CKME)})$$
$$(\varDelta\sigma_x, \varDelta\varepsilon) = (\sigma_x^{(NWKR)} - \sigma_x^{(CKME)}, \varepsilon^{(NWKR)} - \varepsilon^{(CKME)})$$

Fig. 15 shows estimation error of kernel parameters estimated by CV for methods (CKME, LCKME, NWKR, random). y-axis shows error $\varDelta\sigma_x$ and x-axis shows error $\varDelta\varepsilon$.

5.3 An Infinite-sites Coalescent Model in Population Genetics

We apply each method to a dataset, *coal and coalobs*, prepared in Nunes and Prangle (2015). The dataset, coal, consists of a 100000 × 9 matrix, where 100000 is the number of sample, and 9 is the number of parameters (P1-2) and resulting summary statistics (C1-7) generated from an infinite-sites coalescent model for genetic variation. The meaning of parameters and resulting summary statistics are listed below:

- P1: scaled mutation rate $\tilde{\theta}$.
- P2: scaled recombination rate ρ.
- C1: number of segregating sites.

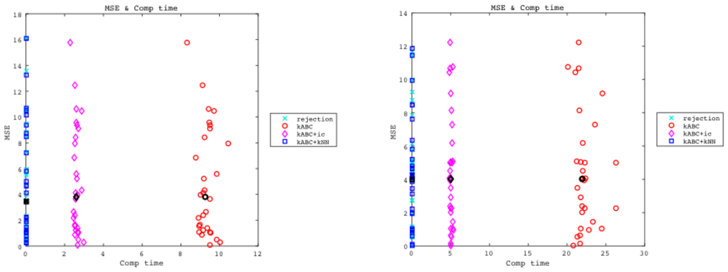

Fig. 16 kNN v.s. Original

- C2: spurious statistic, irrelevant to P1-2, a standard uniform random deviate.
- C3: pairwise mean number of nucleotidic differences.
- C4: mean R^2 across pairs separated by $< 10\%$ of the simulated genomic regions.
- C5: number of distinct haplotypes.
- C6: frequency of the most common haplotype.
- C7: number of singleton haplotypes.

In the generating model, given the two parameters (P1-2), summary statistics (C1-7) are computed. A task here is, given observed summary statistics (C1-7), to estimate the two parameters (P1-2). In the notational setting used in previous sections, $\mathcal{X} = \mathbb{R}^7$ and $\mathcal{Y} = \mathbb{R}^2$.

We adopt the following criterion to evaluate the accuracy:

1. **MSE of a point estimation:** We use the same mean and mode estimations.

Fig. 16 shows MSE vs. computational cost of each method. Fig. shows accuracy of estimated $\tilde{\theta}$ and ρ. x-axis shows the error of $\tilde{\theta}$ and y-axis shows the error of ρ.

5.4 Cross Validation

We set that

- 訓練サンプルは 200 個
  $Y = 10sin(4X) + n, n \sim N(0,1)$
  $X = 2r - 1, r \sim U(0,1)$
- σ_{k_x},σ_{k_y} は $10^{-10} \sim 10^{10}$ の範囲を検証
- σ_{k_x} の検証時の σ_{k_y} は 1 に設定
- k は $n^{2/3}$

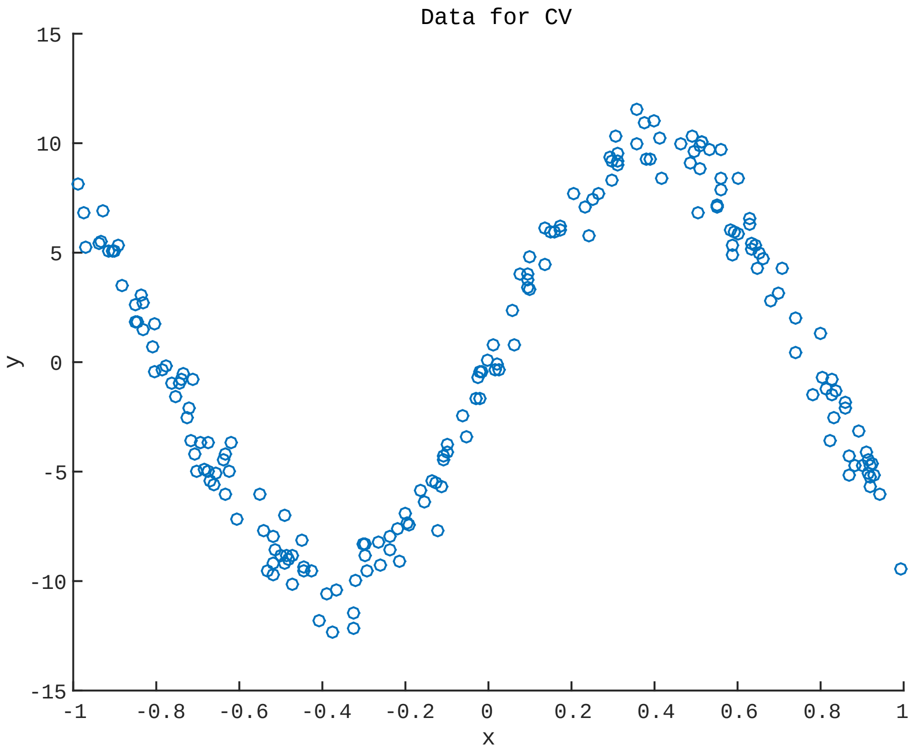

Fig. 17 kNN v.s. Original

Table 1 Please write your table caption here

first	second	third
number	number	number
number	number	number

References

Alaoui A, Mahoney MW (2015) Fast randomized kernel ridge regression with statistical guarantees. In: Advances in Neural Information Processing Systems 28, pp 775–783

Bach F, Jordan MI (2002) Kernel Independent Component Analysis. Journal of Machine Learning Research 3:1–48

Bochner S (1959) Lectures on fourier integrals. with an author's supplement on monotonic functions, stieltjes integrals, and harmonic analysis. In: Princeton University Press, Princeton, NJ

Boots B, Gordon G, Gretton A (2013) Hilbert Space Embeddings of Predictive State Representations. In: The Conference on Uncertainty in Artificial Intelligence (UAI), pp 92–101

Cortes C, Mohri M, Weston J (2005) A General Regression Technique for Learning Transductions. In: International Conference on Machine Learning (ICML), pp 153–160

Fine S, Scheinberg K (2001) Efficient SVM Training Using Low-Rank Kernel Representations. Journal of Machine Learning Research 2:243–264

Fukumizu K, Bach FR, Jordan MI (2004) Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces. Journal of Machine Learning Research 5:73–99

Fukumizu K, Song L, Gretton A (2013) Kernel bayes' rule: Bayesian inference with positive definite kernels. Journal of Machine Learning Research pp 3753–3783

Grünewälder S, Lever G, Baldassarre L, Patterson S, Gretton A, Pontil M (2012a) Conditional mean embeddings as regressors - supplementary. In: International Conference on Machine Learning (ICML), pp 1823–1830

Grünewälder S, Lever G, Baldassarre L, Pontil M, Gretton A (2012b) Modelling transition dynamics in MDPs with RKHS embeddings. In: International Conference on Machine Learning (ICML), pp 535–542

Kanagawa M, Nishiyama Y, Gretton A, Fukumizu K (2014) Monte Carlo Filtering Using Kernel Embedding of Distributions. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp 1897–1903

Mika S, Schölkopf B, Smola A, Müller K, Scholz M, Rätsch G (1999) Kernel PCA and de-noising in feature spaces. In: Neural Information Processing Systems (NIPS), pp 536–542

Muandet K, Fukumizu K, Sriperumbudur B, Schölkopf B (2017) Kernel mean embedding of distributions: A review and beyond. Foundations and Trends in Machine Learning 10(1-2):1–141

Nishiyama Y, Boularias A, Gretton A, Fukumizu K (2012) Hilbert Space Embeddings of POMDPs. In: The Conference on Uncertainty in Artificial Intelligence (UAI), pp 644–653

Nishiyama Y, Afsharinejad AH, Naruse S, Boots B, Song L (2016) The Nonparametric Kernel Bayes' Smoother. In: International Conference on Artificial Intelligence and Statistics (AISTATS), pp 547–555

Nishiyama Y, Kanagawa M, Gretton A, Fukumizu K (2018) Model-based Kernel Sum Rule: Kernel Bayesian Inference with Probabilistic Models. In: arXiv: 1409.5178

Nunes MA, Prangle D (2015) abctools: An R Package for Tuning Approximate Bayesian Computation Analyses. The R Journal 7(2):189–205, URL `https://journal.r-project.org/archive/2015/RJ-2015-030/index.html`

Rahimi A, Recht B (2007) Random features for large-scale kernel machines. In: In Neural Infomration Processing Systems

Rawlik K, Toussaint M, Vijayakumar S (2013) Path Integral Control by Reproducing Kernel Hilbert Space Embedding. Proc 23rd Int Joint Conference on Artificial Intelligence (IJCAI)

Schölkopf B, Smola A (2002) Learning with Kernels. MIT Press, Cambridge

Smola A, Gretton A, Song L, Schölkopf B (2007) A Hilbert space embedding for distributions. In: International Conference on Algorithmic Learning Theory (ALT), pp 13–31

Song L, Huang J, Smola A, Fukumizu K (2009) Hilbert Space Embeddings of Conditional Distributions with Applications to Dynamical Systems. In: International Conference on Machine Learning (ICML), pp 961–968

Song L, Fukumizu K, Gretton A (2013) Kernel embedding of conditional distributions. IEEE Signal Processing Magazine 30(4):98–111

Song Y, Zhu J, Ren Y (2016) Kernel Bayesian Inference with Posterior Regularization. In: Advances in Neural Information Processing Systems 29 (NIPS 2016), pp 4763–4771

Sriperumbudur B, Gretton A, Fukumizu K, Lanckriet G, Schölkopf B (2010) Hilbert Space Embeddings and Metrics on Probability Measures. Journal of Machine Learning Research 11:1517–1561

Steinwart I, Christmann A (2008) Support Vector Machines. Information Science and Statistics. Springer

Weston J, Chapelle O, Elisseeff A, Schölkopf B, Vapnik V (2003) Kernel Dependency Estimation. In: Advances in Neural Information Processing Systems 15, pp 873–880

Zhang Y, Duchi JC, Wainwright MJ (2015) Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. Journal of Machine Learning Research 16:3299–3340

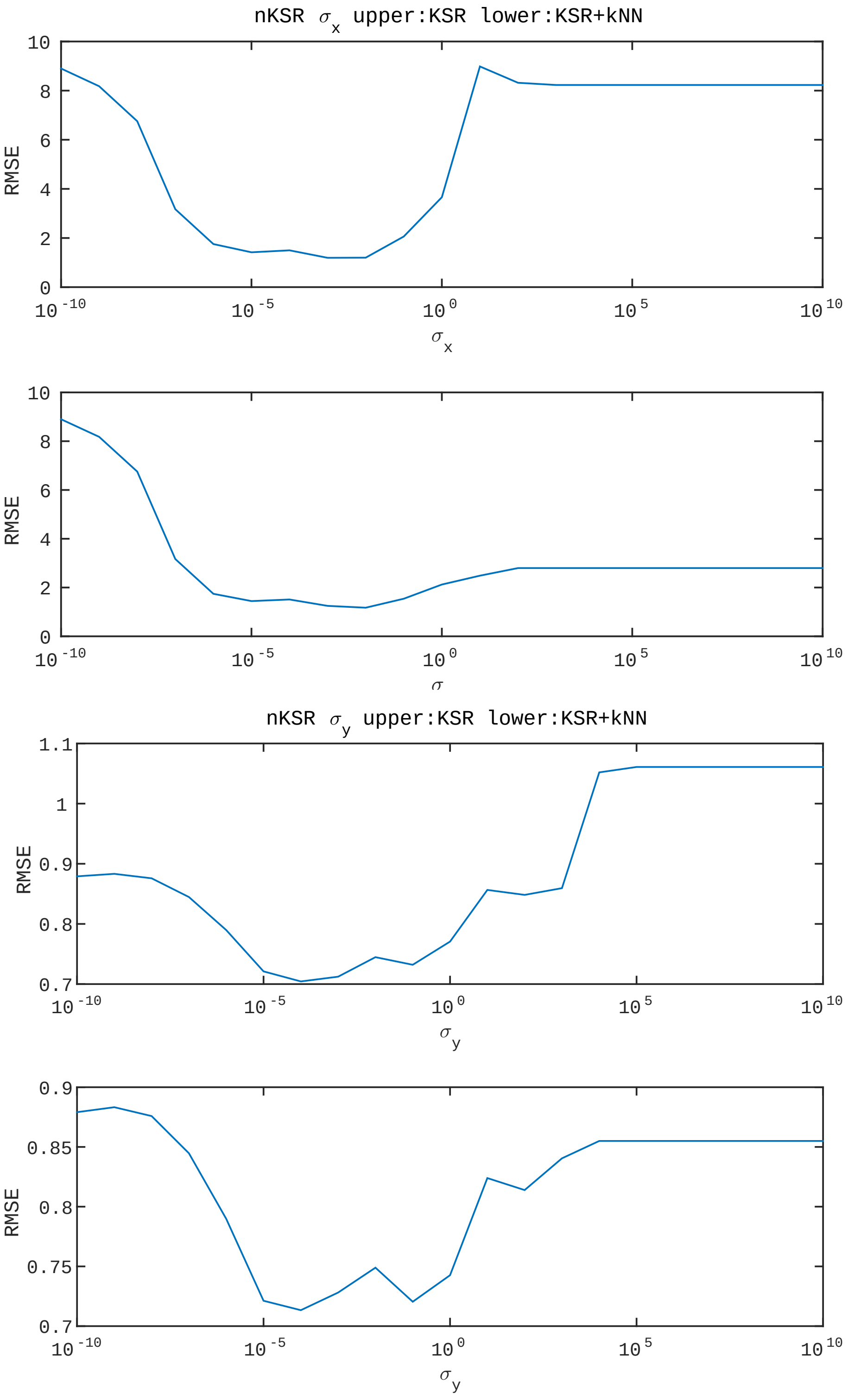

Fig. 18 kNN v.s. Original

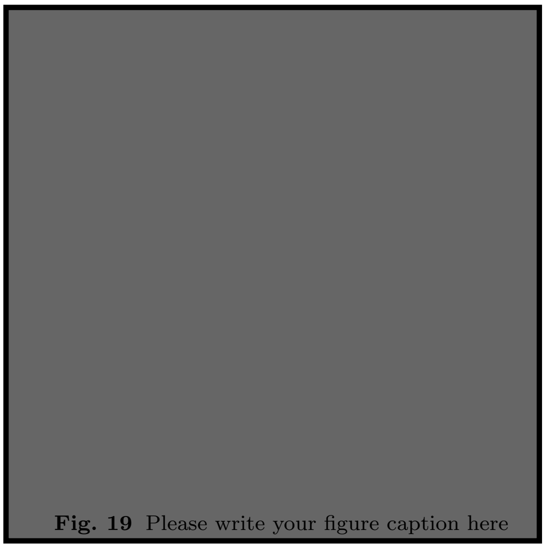

Fig. 19 Please write your figure caption here

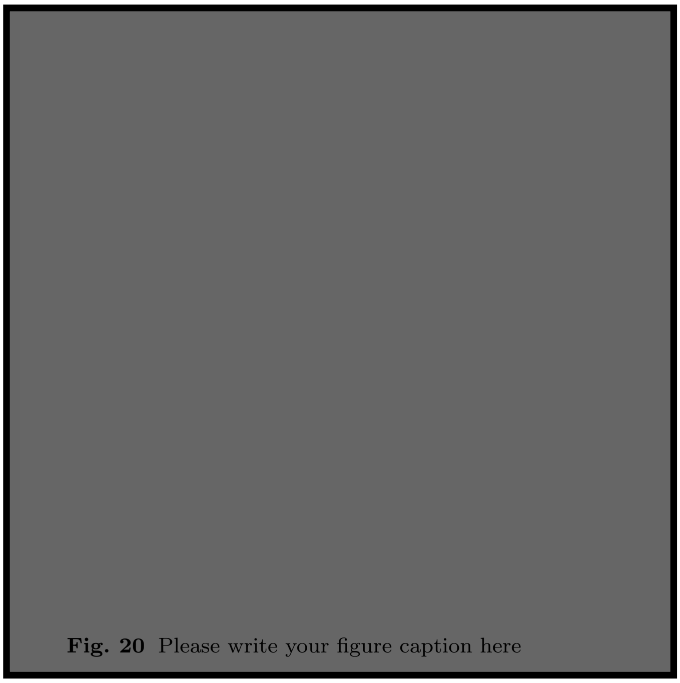

Fig. 20 Please write your figure caption here