entropy

MDPI

Article

Extending Hilbert–Schmidt Independence Criterion for Testing Conditional Independence

Bingyuan Zhang * and Joe Suzuki

Graduate School of Engineer Science, Osaka University, Toyonaka 560-0043, Japan
* Correspondence: zhang@sigmath.es.osaka-u.ac.jp

Abstract: The Conditional Independence (CI) test is a fundamental problem in statistics. Many nonparametric CI tests have been developed, but a common challenge exists: the current methods perform poorly with a high-dimensional conditioning set. In this paper, we considered a nonparametric CI test using a kernel-based test statistic, which can be viewed as an extension of the Hilbert–Schmidt Independence Criterion (HSIC). We propose a local bootstrap method to generate samples from the null distribution $H_0 : X \perp\!\!\!\perp Y \mid Z$. The experimental results showed that our proposed method led to a significant performance improvement compared with previous methods. In particular, our method performed well against the growth of the dimension of the conditioning set. Meanwhile, our method can be computed efficiently against the growth of the sample size and the dimension of the conditioning set.

Keywords: conditional independence test; dependence measure; local bootstrap

check for updates

Citation: Zhang, B.; Suzuki, J. Extending Hilbert–Schmidt Independence Criterion for Testing Conditional Independence. *Entropy* **2023**, *25*, 425. https://doi.org/10.3390/e25030425

Academic Editor: Carlos Alberto De Bragança Pereira

Received: 26 January 2023
Revised: 22 February 2023
Accepted: 25 February 2023
Published: 26 February 2023

1. Introduction

The Conditional Independence (CI) test is a statistical hypothesis test that examines whether variables X and Y are conditionally independent given another variable Z, denoted as $X \perp\!\!\!\perp Y \mid Z$, when we observe the actual values of the three variables. The CI test plays a critical role in Bayesian network structure learning [1,2] and causal discovery [3].

The task is relatively easy when the sample size n is large and the variable Z is discrete, because then, we can test the independence of X, Y for each value of Z [4]. On the other hand, if X, Y, Z have a joint Gaussian distribution, then the CI reduces to a zero partial correlation between X and Y given Z [5], which can also be easily tested. In this paper, we considered X, Y, Z without making any assumption on the joint distribution. X, Y, Z can be either continuous or discrete variables. The problem becomes challenging with a growing dimension d_Z due to the curse of dimensionality [6], when Z may be a set of d_Z variables or any d_Z-dimensional random vector.

Another major challenge in CI tests is the need to sample from the null distribution $H_0 : X \perp\!\!\!\perp Y \mid Z$. In general, statistical hypothesis tests require us to obtain the distribution of the test statistic under the null hypothesis H_0. However, when we are only given the observations, the exact distribution for any test statistic under the CI case ($H_0 : X \perp\!\!\!\perp Y \mid Z$) is unknown. The two approaches below are the most-popular ways to obtain an approximated null distribution:

- **Permutation method:**
  One approach is by permuting the observed samples. In the independence test, where $H_0 : X \perp\!\!\!\perp Y$, though X and Y in each pair $(x_1, y_1), \ldots, (x_n, y_n)$ are not independent, we may regard X and Y of shifted pairs of $(x_1, y_2), \ldots, (x_{n-1}, y_n), (x_n, y_1)$ to be independent. Thus, we can compute the test statistic values on the shifted pairs, which mimic H_0, and obtain a histogram as an approximated null distribution. However, in the CI test, as the conditioning set Z exists, we cannot shift $\{x_i\}, \{y_i\}, \{z_i\}$ in order to make them conditionally independent [7,8].

- **Asymptotic method:**
  The other approach utilizes the asymptotic distributions of the test statistics [9–11]. For some test statistics, their asymptotic distributions are derived. In that case, the asymptotic distribution of a test statistic can be used to approximate the null distribution. Though these asymptotic distributions can be generated efficiently, they are less accurate when the sample size n is small or with a high-dimensional Z [8,12].

Our contributions: In this paper, we propose a new CI test including a novel test statistic and a local bootstrap method to sample from $H_0 : X \perp\!\!\!\perp Y \mid Z$. In many CI tests, test statistics directly evaluate the conditioning set Z, which becomes difficult when Z is high-dimensional or has a complex density. Our proposed test statistic does not directly access conditioning set Z, which alleviates the curse of dimensionality. Such a test statistic is expected to be more robust for a high-dimensional conditional set. The experiment result showed that our proposed test had a comparable performance when Z is low-dimensional and notably outperformed others when Z is high-dimensional. Moreover, our proposed method can be computed efficiently regarding the growing sample size n and growing dimension of Z. We summarize our main contributions as follows:

- We designed a novel test statistic in the following procedure: we first subdivided Z into several local clusters, then measured the unconditional independence in each cluster, and finally, combined the unconditional independence measures into a single number as the measure of conditional independence. In particular, we used k-means to find clusters of Z and the Hilbert–Schmidt Independence Criterion (HSIC) [13] as the measure of unconditional independence in each cluster. We took the sum of the local HSIC values as our test statistic for conditional dependence.
- We propose to use a local bootstrap method to sample from the CI case $H_0 : X \perp\!\!\!\perp Y \mid Z$. We extended the local bootstrap strategy in [14] and showed the theoretical consistency of the bootstrap distribution. The local bootstrap method worked well combined with the proposed test statistic, but can also be applied to other CI tests.

The paper is organized as follows. In Section 2, we discuss some related works on the CI test. In Section 3, we introduce the notations and provide an overview of the HSIC, a kernel-based measure of unconditional independence. In Section 4, we show the details about the test procedure and explain both the test statistic and the local bootstrap method. In Section 5, we compare with other representative CI tests based on the synthetic data. Finally, we summarize our results in Section 6.

2. Related Work

Recently, numerous nonparametric methods have been proposed for CI testing. Many test statistics have been constructed by embedding distributions in Reproducing Kernel Hilbert Spaces (RKHSs). Fukumizu et al. [7] proposed a measure of CI based on cross-covariance operators. However, its asymptotic distribution under the null hypothesis is unknown, and the bin-based permutation degrades as the dimension of conditioning variable Z grows. Later, Zhang et al. [10] proposed the KCIT, based on the partial association of functions in some universal RKHS. A major advantage of the KCIT is a known asymptotic distribution that can be efficiently approximated using Monte Carlo simulations. For the CI test on a large-scale dataset, Strobl et al. [11] proposed the RCIT and RCoT to use random Fourier features to approximate the KCIT efficiently. Huang et al. [15] proposed a Kernel Partial Correlation (KPC), a generalization of a partial correlation to measure conditional dependence. Beyond kernel-based methods, Runge [12] used a Conditional Mutual Information (CMI) estimator as the test statistic and proposed a k-nearest-neighbor-based permutation to generate samples from the null distribution. Shah and Peters [16] proposed a Generalized Covariance Measure (GCM) as the test statistic based on regression method. Doran et al. [8] turned the CI test into a two-sample test by finding a permutation matrix and measuring the Maximum Mean Discrepancy (MMD) [17] between the two distributions. Sen et al. [18] proposed a method called the CCIT, which turns the CI test into a classification problem. In [8,18], they both gave an additional sampling step involving

data splitting, potentially reducing the power when the dataset is small. Some other model-powered methods also make use of the GAN [19,20] and Double-GAN [21].

While nonparametric CI tests make no assumption about the joint distribution of X, Y, Z, imposing additional assumptions helps to simplify the problem. Some milder assumptions are considered. In particular, X and Y are assumed to be in function forms of variable Z plus an additive independent noise term, which has a zero mean:

$$X = f(Z) + \varepsilon_x, \quad Y = g(Z) + \varepsilon_y.$$

If the estimated noise terms are independent $\varepsilon_x \perp\!\!\!\perp \varepsilon_y$, we conclude that $X \perp\!\!\!\perp Y \mid Z$ [22–26]. The methods in this category need to find a regression function and then test for the unconditional independence of the residuals.

For further details about the different characterizations of CI, see [27]. From a theoretical perspective, Shah and Peters [16] proved there exists no universally valid CI testing for all CI cases. Precisely, no CI test can control Type-I error for all the CI cases while having a higher power against any alternative. However, a desirable CI test is supposed to be computationally efficient.

3. Background on Kernel Methods

This section introduces the notations and gives the basic definitions related to the kernel methods. For further details, see [13,28,29]. We use X, Y, Z and $x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}$ to represent random variables and their observed samples and use $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ to denote the associated domains. We considered a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that corresponds to a Hilbert space $\mathcal{H}$ and a feature map $\Psi : \mathcal{X} \to \mathcal{H}$ such that

$$k(x_1, x_2) = \langle \Psi(x_1), \Psi(x_2) \rangle_{\mathcal{H}}$$

for $x_1, x_2 \in \mathcal{X}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product of the Hilbert space $\mathcal{H}$. Such an $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) with respect to the kernel k, denoted as $\mathcal{H}_k$. For example, the Gaussian kernel $k(x_1, x_2) = \exp\left(-\|x_1 - x_2\|^2/\sigma\right)$ is a positive definite kernel, and we considered it a default choice in the paper. Let k be a kernel defined on $\mathcal{X}$ and its corresponding RKHS be $\mathcal{H}_k$. We fixed a set $\mathcal{P}$ of measures.

Definition 1 (Kernel embedding)**.** *The kernel embedding of the measure μ into the RKHS $\mathcal{H}_k$ is the map $m_k : \mathcal{P} \to \mathcal{H}_k$ defined by*

$$\mathcal{P} \ni \mu \mapsto m_k(\mu) := \int k(\cdot, x) d\mu(x) \in \mathcal{H}_k \,.$$

From the above definition, a direct consequence is

$$\int f(x) d\mu(x) = \langle f, m_k(\mu) \rangle_{\mathcal{H}_k}, \forall f \in \mathcal{H}_k.$$

Definition 2 (MMD)**.** *The Maximum Mean Discrepancy (MMD) between $P, Q \in \mathcal{P}$ is*

$$MMD(P, Q) := \|m_k(P) - m_k(Q)\|^2_{\mathcal{H}_k}.$$

It is easy to see that the MMD takes non-negative values. In particular, for characteristic kernels (e.g., the Gaussian kernel), the MMD(P, Q) becomes zero if and only if the measures P, Q coincide [17].

Finally, we considered an unconditional dependence measure for variables X and Y. Let $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ be kernels on $\mathcal{X}$ and $\mathcal{Y}$ and $\mathcal{H}_{k_{\mathcal{X}}}$ and $\mathcal{H}_{k_{\mathcal{Y}}}$ be the corresponding RKHSs. Gretton et al. [13] defined the Hilbert–Schmidt Independence Criterion (HSIC), which can be viewed as the MMD between a measure P_{XY} of X, Y and the product $P_X P_Y$ of the marginalized measures P_X, P_Y. The HSIC is a state-of-the-art dependence measure, which suits both continuous and discrete variables. The HSIC has been well studied as a test statistic in

independence testing [10,13,17]. For a characteristic kernel, the HSIC(X, Y) is zero if and only if $P_{XY} = P_X P_Y$, which indicates $X \perp\!\!\!\perp Y$.

More precisely, we may express the HSIC as follows:

Definition 3 (HSIC)**.**

$$\begin{aligned} HSIC(X, Y) &:= \|m_k - m_{k_\mathcal{X}} m_{k_\mathcal{Y}}\|^2_\mathcal{H} \\ &= \|\mathbb{E}_{XY}[k_\mathcal{X}(X, \cdot)k_\mathcal{Y}(Y, \cdot)] - \mathbb{E}_X[k_\mathcal{X}(X, \cdot)]\mathbb{E}_Y[k_\mathcal{Y}(Y, \cdot)]\|^2_\mathcal{H}, \end{aligned}$$

where $\mathcal{H}$ *is the corresponding RKHS of the kernel* $k := k_\mathcal{X} k_\mathcal{Y}$ *defined by*

$$k((x, y), (x', y')) = k_\mathcal{X}(x, x')k_\mathcal{Y}(y, y')$$

for $(x, y), (x', y') \in \mathcal{X} \times \mathcal{Y}$.

The HSIC(X, Y) is known to have an alternative expression:

$$\text{HSIC}(X, Y) = \mathbb{E}_{XYX'Y'}[\text{C}(X, Y, X', Y')] \tag{1}$$

where $\text{C}(X, Y, X', Y')$ is

$$\Big[k_\mathcal{X}(X, X') - \mathbb{E}_{X''}\big[k_\mathcal{X}(X, X'')\big]\Big]\Big[k_\mathcal{Y}(Y, Y') - \mathbb{E}_{Y''}\big[k_\mathcal{Y}(Y', Y'')\big]\Big], \tag{2}$$

and (X', Y') are independent copies of (X, Y). Given data points $(x_1, y_1), \ldots, (x_n, y_n)$, we considered the following estimator [13]:

$$\widehat{\text{HSIC}}(X, Y) = \frac{1}{n^2}\text{tr}(\boldsymbol{K}_X \boldsymbol{H} \boldsymbol{K}_Y \boldsymbol{H}) \tag{3}$$

where $(\boldsymbol{K}_X)_{ij} = k(x_i, x_j)$, $(\boldsymbol{K}_Y)_{ij} = k(y_i, y_j)$, $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^T$, and $\mathbb{1}$ is an n vector of ones. Intuitively, we expect an estimator of the HSIC to be a small value when $X \perp\!\!\!\perp Y$.

4. Proposed Method

In this section, we introduce our proposed method. First, we present a novel test statistic. We considered using characteristic kernels as a default choice, i.e., the Gaussian kernel. Next, we explain the local bootstrap algorithm to generate samples from $\text{H}_0 : X \perp\!\!\!\perp Y \mid Z$. The test is summarized in Algorithm 1. Finally, we discuss the effect of the parameters and provide a time complexity analysis of the overall procedure.

We start by looking at the CI definition: $X \perp\!\!\!\perp Y \mid Z$ means X and Y are independent for any fixed value of Z. Here, we used $\text{HSIC}(X, Y \mid Z = z) := \mathbb{E}_{XYX'Y'}[C(X, Y, X', Y')|Z = z]$ to represent the HSIC on (X, Y) with a fixed Z value, where (X', Y') are copies of (X, Y).

$$\begin{aligned} X \perp\!\!\!\perp Y \mid Z &\iff X \perp\!\!\!\perp Y \mid Z = z,\ \forall z \in \mathcal{Z}. \\ &\iff \text{HSIC}(X, Y \mid Z = z) = 0,\ \forall z \in \mathcal{Z}. \end{aligned}$$

As a direct result, we have the following proposition.

Proposition 1 (Characterization of CI)**.**

$$X \perp\!\!\!\perp Y \mid Z \iff \int HSIC(X, Y \mid Z) d\mu(Z) = 0 \tag{4}$$

where $\mu(Z)$ *is the probability measure on* Z.

Proof sketch: By definition, $\mathrm{HSIC}(X, Y \mid Z = z) = 0$ always takes non-negative values. Thus, for a characteristic kernel, the integral becomes zero if and only if $\mathrm{HSIC}(X, Y \mid Z = z) = 0$, $\forall z \in \mathcal{Z}$, which indicates $X \perp\!\!\!\perp Y \mid Z$.

Based on the above fact, conditional dependence can be measured by the marginal unconditional dependence measure. Here, we considered the following procedure to calculate our test statistic:

1. Perform the clustering algorithm to subdivide Z into M clusters, and let its index set be C_M.
2. Measure the unconditional dependence $\widehat{\mathrm{HSIC}}_{C_m}(X, Y)$ for each cluster C_m.
3. Combine the sum of the values as the test statistic:

$$T = \sum_{m=1}^{M} \widehat{\mathrm{HSIC}}_{C_m}(X, Y). \tag{5}$$

We used the sum of the local unconditional dependence measure as the conditional dependence measure, which is similar in spirit to [4]. Margaritis [4] considered dividing a univariate $Z \in \mathbb{R}^1$ into local bins and using the product of the local measure as a single number. Our method applies to a high-dimensional Z and takes the sum of kernel-based measures. Given the data $(x_i, y_i, z_i), i = 1, \ldots, n$, we divided them into M clusters based on the value of Z, and the estimator is

$$\widehat{\mathrm{HSIC}}_{C_m}(X, Y) = \frac{1}{|C_m|^2} \mathrm{tr}(\boldsymbol{K}_X^{(m)} \boldsymbol{H} \boldsymbol{K}_Y^{(m)} \boldsymbol{H})$$

where $|C_m|$ is the size of C_m and $\boldsymbol{K}_X^{(m)}$ and $\boldsymbol{K}_Y^{(m)}$ are the corresponding kernel matrices for samples (x_i, y_i), $\forall i \in C_m$. It is easy to see that the conditioning set Z is only used in deciding the local clusters. By doing that, we alleviate the influence of the dimension of Z.

4.1. Local Bootstrap

In this subsection, we introduce the local bootstrap method to sample from $\mathrm{H}_0 : X \perp\!\!\!\perp Y \mid Z$, which completes the CI test. The key is to break the dependence between X and Y while keeping the dependence between (X, Z) and (Y, Z). An example of an ideal CI permutation is explained in Figure 1.

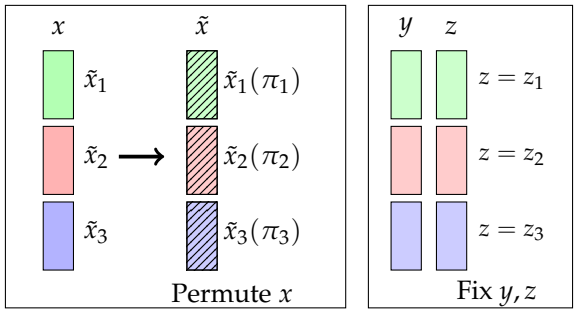

Figure 1. An ideal permutation in the CI. Given the data, we first divided different bins (green, red, and blue), and each bin $\tilde{x}$ includes samples that have the same z. From that, we fixed y and z and shuffled x within each bin with some permutations (π_1, π_2, π_3) to generate new data $(\tilde{x}, y, z)$. An ideal permutation successfully generates samples that keep the dependence between (X, Z) and (Y, Z) while satisfying $X \perp\!\!\!\perp Y$.

In practice, it is impossible to perform the ideal permutation because we do not have enough samples that have the same z. As an alternative method, we used a local bootstrap.

First, given with different z^*, we generated (x^*, y^*) independently from the following discrete distribution:

$$\begin{aligned} x^* &\sim \hat{G}_{x|z^*} : \begin{pmatrix} x_1 & x_2 & \dots, & x_n \\ w_1 & w_2 & \dots, & w_n \end{pmatrix}, \\ y^* &\sim \hat{G}_{y|z^*} : \begin{pmatrix} y_1 & y_2 & \dots, & y_n \\ w_1 & w_2 & \dots, & w_n \end{pmatrix}, \end{aligned} \tag{6}$$

where $w_j = \frac{K(z_j - z^*/\gamma)}{\sum_{j=1}^{n} K(z_j - z^*/\gamma)}$ are the probabilities to sample the index j. Under the mild assumptions [30], we show the consistency of the bootstrap distribution at each z^*. See the proof in Appendix A.

Proposition 2. *The empirical bootstrap distribution converges to a conditional distribution with each fixed point of z^*:*

$$|\hat{G}_{x|z^*}\hat{G}_{y|z^*} - P(X \mid Z = z^*)P(Y \mid Z = z^*)| \to 0, \quad as \quad n \to \infty. \tag{7}$$

The local bootstrap strategy is an extension of [14], which was original designed for sampling (x, y) according to a regression model. We extended it using a Nadaraya–Watson kernel estimator [31] to assign the weights for indexes to be sampled. If z_j is close to z^*, w_j is large; thus, the index j has a larger possibility of being sampled. Moreover, x^* and y^* are sampled independently. Thus, it is less possible for both x_j and y_j to be sampled simultaneously, which breaks the dependence between X and Y. Shi [14] suggested that the bandwidth γ should be varied for different z^*. Here, we narrowed the candidates from $1, \dots, n$ to the 10-nearest neighbors of each z^* and let the local bandwidth γ of z^* be the squared Euclidean distance between z^* and its 10-th nearest neighbor.

The local bootstrap is summarized in Algorithm 1. Each time, we generated n samples as if they come from $\mathrm{H}_0 : X \perp\!\!\!\perp Y \mid Z$ and calculated the test statistic value T. We repeated this K-times on the generated samples and calculated the p-value based on the histogram. We reject H_0 if the p value is smaller than a predefined significance level. Otherwise, we accept H_0. We summarize the procedure in Algorithm 2.

Algorithm 1: Local bootstrap

Input: Data (x_i, y_i, z_i), $i = 1, \dots, n$.
Output: New samples: (x_i^*, y_i^*, z_i^*), $i = 1, \dots, n$

for $i \leftarrow 1$ **to** n **do**
    Let $z_i^* = z_i$
    Sample $x_i^* \sim \hat{G}_{x|z_i^*}$, $\quad y_i^* \sim \hat{G}_{y|z_i^*}$.
end

Algorithm 2: Test

Input: Data (x_i, y_i, z_i), $i = 1, \dots, n$.
Cluster number M.
Times to repeat K.
Output: p-value.

Find M clusters.
Estimate the T.
for $k \leftarrow 1$ **to** K **do**
    Generate samples with Algorithm 1
    Estimate T_i on the generated samples.
end
Compute the p-value:

$$p = \frac{1}{K} \sum_{t=1}^{T} \{T_k \geq T\}.$$

4.2. Effect of M

The choice of the cluster number M affects the test performance. M needs to be chosen properly so that we can focus on the pairs (x, y) who have similar z values while having enough pairs (x, y) in each local cluster to make a good estimation of the local HSIC. Besides, the number of M has an effect on the computational complexity: a smaller M makes bigger clusters on average and takes more time to find local HSICs. In practice, we fixed the average cluster size and used k-means to decide the clusters. We let M be $\lceil n/\bar{C} \rceil$, where $\lceil x \rceil$ takes the least integer that is not smaller than x, and $\bar{C}$ is the average cluster size.

4.3. Complexity Analysis

We discuss the time complexity of the test procedure. In the beginning, our method found M clusters and weights w in the bootstrap. Both were calculated once and took little time. The major computational cost was in repeatedly finding the test statistic T_k. Estimating T scales less than $\mathcal{O}(M|\tilde{C}|^2)$, where M is the number of clusters and the $|\tilde{C}|$ is the maximum set size among all clusters and is smaller than n. This was repeated T-times over the generated samples to construct the histogram of the null distribution. The test took $\mathcal{O}(Mn^2K)$. The bootstrap part can be easily parallelized to promote the speed further, but this was beyond the scope of the paper.

5. Experiments

In this section, we compare the proposed methods with other nonparametric CI tests. We denote our proposed method as a Bundle of HSICs (**BHSIC**). The evaluation was focused on the Type-I error rate, Type-II error rate, and runtime. A lower Type-II error rate and computational efficiency are essential for a good CI test. In particular, we compared with some representative methods: **KCIT** [10], **RCIT, RCoT** [11], **CCIT** [18], and **CMIknn** [12]. For details about these methods, see Section 2. All methods have source codes that are available online. Different methods are implemented in different programming languages, and we focused on how these methods scale with the sample size and the dimension of Z instead of a direct comparison of the runtimes.

We were interested in the performance of the methods with different settings. In our simulations, we considered the following two models. The first model was a simple linear regression model. The second model was a post-nonlinear noise model, which is a commonly used setting in evaluating CI tests [10–12]. The functional forms of X and Y on Z are as follows:

$$\text{Model 1}: \quad X = \sum_{i=1}^{d_Z} \alpha_i Z_i + c\varepsilon_b + \varepsilon_1, \quad Y = \sum_{i=1}^{d_Z} \beta_i Z_i + c\varepsilon_b + \varepsilon_2,$$

$$\text{Model 2}: \quad X = g_1\Big(\sum_{i=1}^{d_Z} Z_i + c\varepsilon_b + \varepsilon_1\Big), \quad Y = g_2\Big(\sum_{i=1}^{d_Z} Z_i + c\varepsilon_b + \varepsilon_2\Big),$$

where $Z = (Z_1, \ldots, Z_{d_Z})$, ε_1, ε_2, and ε_b are independent standard Gaussian. The coefficients $\alpha_i, \beta_i \sim Uniform(-0.5/d_z, 0.5/d_z)$ and the functions $g_1(\cdot)$ and $g_2(\cdot)$ were uniformly chosen from $\{(\cdot), (\cdot)^2, (\cdot)^3, \tanh(\cdot), \exp(-\|\cdot\|_2)\}$. We considered (a) $\mathrm{H}_0 : X \perp\!\!\!\perp Y \mid Z$ with $c = 0$ and (b) $\mathrm{H}_1 : X \not\perp\!\!\!\perp Y \mid Z$ with $c = 1$.

In the following simulations, we studied the test performance on different sample sizes and dimensions of Z. The sample sizes n varied from $\{100, 200, 400, 600, 800\}$ with fixed dimensions of $d_Z = 1$ and $d_Z = 10$. The dimensions d_Z varied from $\{1, 2, 5, 10, 20\}$ with a fixed sample size of $n = 400$. We also studied the effect of the cluster number M in our proposed method. The significance levels were set to be $\alpha = 0.05$ in all the simulations. The evaluations of the Type-I error rate, Type-II error rate, and mean runtimes are reported over 100 replications. The Type-I error rate is the false rejection percentage when the underlying truth is $\mathrm{H}_0 : X \perp\!\!\!\perp Y \mid Z$ with $c = 0$, and the Type-II error rate is the false acceptance percentage when the underlying truth is $\mathrm{H}_1 : X \not\perp\!\!\!\perp Y \mid Z$ with $c = 1$. Runtime is the average time to perform one test.

5.1. Hyperparameters

The choice of the hyperparameters affects the results. For the KCIT, RCIT, and RCoT, the bandwidths in the Gaussian kernels were set to be the squared median Euclidean distance between (X, Y) using all the pairs (or the first 500 pairs if $n > 500$) double the conditioning set size, which was recommended in [11]. Thew CMIknn has two hyperparameters: the neighbor size $k_{\text{CMI}} = 0.1n$ in finding the estimator of the CMI, and $k_{perm} = 5$ in the permutation, respectively. The permutation in CMIknn was repeated 1000-times as the default [12].

In our proposed methods, the bandwidths were set to be the squared median Euclidean distance between (X, Y) in each local cluster. The number of clusters M was set to be $\lceil n/50 \rceil$ when $n <= 200$ and $\lceil n/80 \rceil$ when $n > 200$, where $\lceil x \rceil$ takes the least integer that is bigger than or equal to x. On average, each cluster had 50 samples when $n <= 200$ and 80 samples otherwise. The local bootstrap was repeated 1000-times.

5.2. When Z Is Low-Dimensional

We first examined the test performance when Z is generated independently of a standard Gaussian distribution. The sample size n changed from 100 to 800. The simulation results on Linear Model 1 and Nonlinear Model 2 are reported in Figure 2 and Figure 3, respectively. Both the Type-I error rate and Type-II error rate are reported. Because runtime is independent of the model, we only report it in Figure 3.

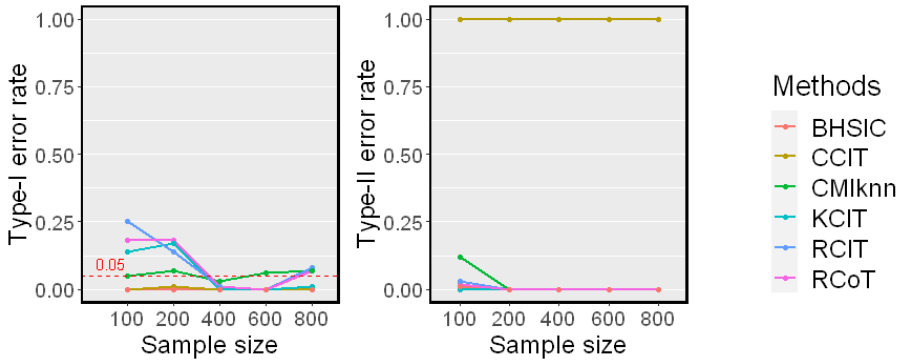

Figure 2. Simulation results on Linear Model 1 ($d_Z = 1$). The significance level is $\alpha = 0.05$. Type-I error rates and Type-II error rates are reported.

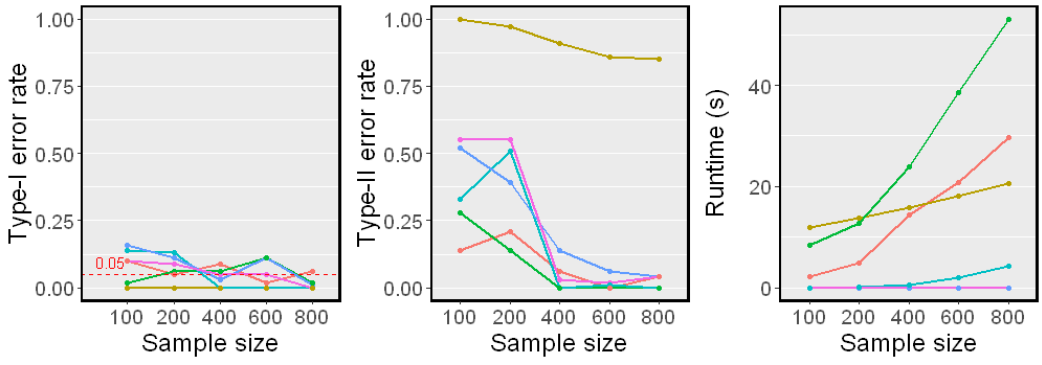

Figure 3. Simulation results on Nonlinear Model 2 ($d_Z = 1$). The significance level is $\alpha = 0.05$. Type-I error rates, Type-II error rates, and mean runtimes are reported.

The Linear Model 1 setting with a single conditioning variable Z is very simple. All methods had controlled Type-I error rates around $\alpha = 0.05$ and almost zero Type-II error rates, except for the CCIT. In our experiments, the performance of the CCIT was constantly among the worst. The data splitting procedure in the CCIT seems to reduce the power of the test when the sample size is small. In the Nonlinear Model 2 setting, all methods had controlled Type-I error rates around $\alpha = 0.05$. However, it was shown

that the proposed method and the CMIknn had better powers against the others when the sample size n was smaller. This matched the result in [12] that CMIknn performed well with a low-dimensional conditioning set Z. When the sample size n was larger than 400, most methods had relatively low Type-II error rates. From the runtime plot, the proposed method was less efficient than the KCIT, RCIT, and RCoT, which are based on an asymptotic distribution. Though the BHSIC and CMIknn were slower, the sampling procedure can readily be parallelized.

5.3. When Z Is High-Dimensional

We next examined the test performance when Z was a set of 10 variables, and each variable in conditioning set Z was generated independently from a standard Gaussian distribution. The sample size n changed from 100 to 800. The simulation results on Linear Model 1 and Nonlinear Model 2 are reported in Figure 4 and Figure 5, respectively.

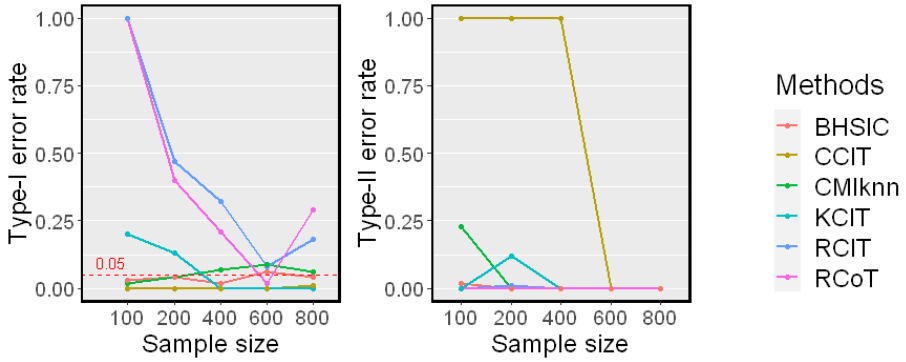

Figure 4. Simulation results on Linear Model 1 ($d_Z = 10$). The significance level $\alpha = 0.05$. Type-I error rates and Type-II error rates are reported.

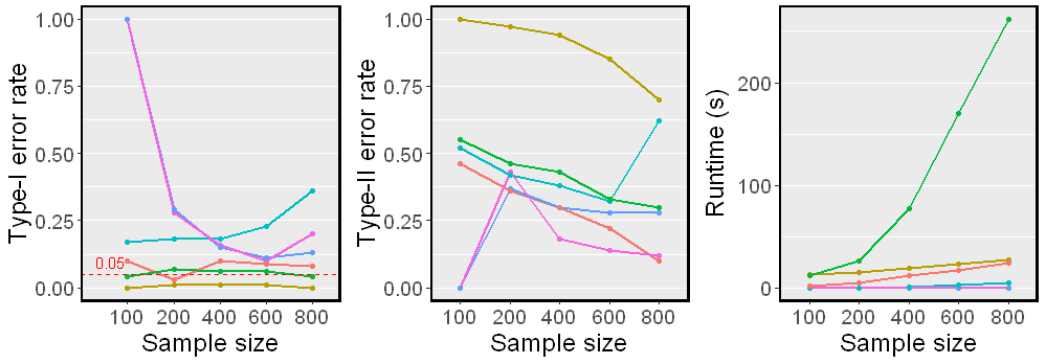

Figure 5. Simulation results on Nonlinear Model 2 ($d_Z = 10$). The significance level $\alpha = 0.05$. Type-I error rates, Type-II error rates, and mean runtimes are reported.

In both linear and nonlinear settings, the RCIT and RCoT failed and had high Type-I error rates. The RCIT and RCoT approximated the KCIT by using random Fourier features and were designed for large-scale datasets. Though they are more scalable than the KCIT, their performances were poor when the sample size was relatively small. The KCIT, CMIknn, and BHSIC performed well in the linear model setting. In the nonlinear model setting, the KCIT showed greater Type I error rates and Type II error rates because the high-dimensional Z led to a less accurate estimation of the asymptotic distribution. We noticed that the BHSIC showed a higher power than the other methods. As we expected, it was beneficial to avoid evaluating the high-dimensional Z directly, which made the method more robust.

5.4. When d_Z Changes

Next, we examined the performance when the dimension of Z changes. We fixed the sample size $n = 400$ and changed d_Z from 1 to 20 in Nonlinear Model 2. The results are shown in Figure 6. Our proposed method performed well against the growth of the dimension of Z and showed a higher power than other methods. Moreover, the dimension of Z did not affect the runtimes since Z was used in k-means only once, which coincided with our complexity analysis.

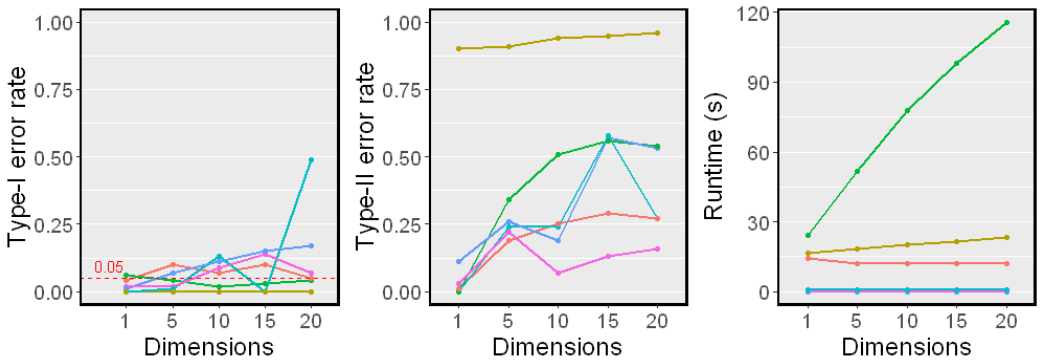

Figure 6. Simulation results on different dimensions of Z ($d_Z = 1, 5, 10, 15, 20$). The sample size $n = 400$ was fixed. The significant level $\alpha = 0.05$. Type-I error rates, Type-II error rates, and mean runtimes are reported.

5.5. Effect on M

Now, we study the effect on the cluster number M. We fixed the sample size $n = 400$ in Nonlinear Model 2 and changed M from 2 to 20. We examined both low-dimensional ($d_Z = 1$) and high-dimensional ($d_Z = 10$) cases. The results are shown in Figure 7.

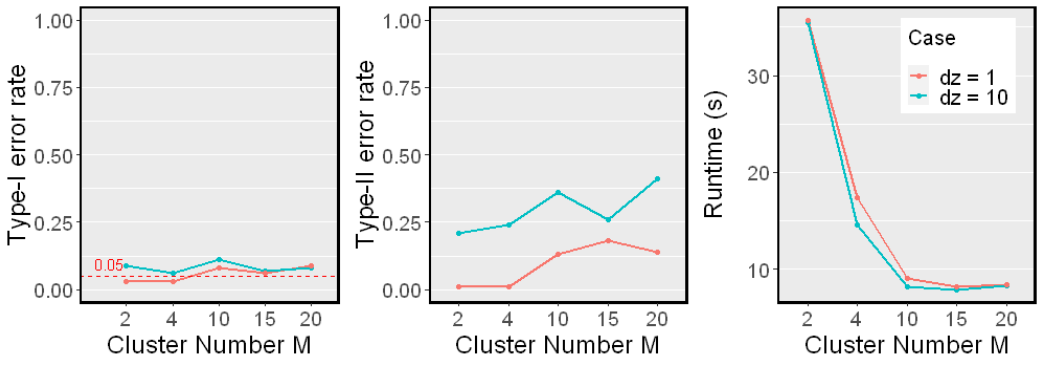

Figure 7. Simulation results on a different cluster number M. The sample size $n = 400$ was fixed. Results on different dimensionality of Z are reported ($d_Z = 1$, red line; $d_Z = 10$, blue line). The significant level $\alpha = 0.05$. Type-I error rates, Type-II error rates, and mean runtimes are reported.

We noticed that the Type I error rates were controlled when $d_Z = 1$ and $d_Z = 10$. As the cluster number M grew to more than 10, the Type-II error rate increased, but the runtimes reduced. The reason is that when we divided the sample points into more clusters, each cluster had fewer points. Thus, the estimation of each local HSIC value became less accurate. On the other hand, the computational cost of the proposed test reduced as the clusters became smaller when the cluster number M grew. The number of samples in each cluster depends on the choice of the clustering algorithms as well. In our experiment, we simply used naive k-means.

6. Conclusions

In this paper, we proposed a novel CI test including a new test statistic and a local bootstrap method to generate samples from the null hypothesis. We first performed

clustering to avoid directly evaluating the high-dimensional conditioning set Z. Then, we used the clustering result and combined several local dependence measures as a measure of conditional dependence. Consequently, the problems caused by a high-dimensional Z can be suppressed. The experimental results showed that our method is robust and performs well against the growth of the dimension of the conditioning set.

Author Contributions: Methodology, B.Z. and J.S.; validation and writing, B.Z.; supervision, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Japan Science and Technology Agency, Grant Number JPMJSP2138 and the Grant-in-Aid for Scientific Research (C) 18K11192.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The evaluation in the paper is based on synthetic data described above.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Proposition 2

We assumed that

$$X = g_x(Z) + \epsilon, \quad Y = g_y(Z) + \epsilon, \tag{A1}$$

and the local bootstrap method generates samples from $X \perp\!\!\!\perp Y \mid Z$.

For each $z_i^* = z_i$, the empirical conditional probability density functions are

$$\hat{G}_{x|z^*}(x < t) = \sum_{j=1}^{n} w_j \mathbf{1}_{x_j < t}, \tag{A2}$$

$$\hat{G}_{y|z^*}(y < t) = \sum_{j=1}^{n} w_j \mathbf{1}_{y_j < t}. \tag{A3}$$

where $w_j = \frac{K\left(\frac{z_j - z^*}{\gamma}\right)}{\sum_{j=1}^{n} K\left(\frac{z_j - z^*}{\gamma}\right)}$. We used $G_{x|z^*}$ and $G_{y|z^*}$ to denote the underlying truth:

$$G_{x|z^*}(x < t) = P(X < t \mid Z = z^*),$$
$$G_{y|z^*}(y < t) = P(Y < t \mid Z = z^*).$$

Consider pairs $(x, z) \in \mathbb{R} \times \mathbb{R}^d$. Devroye and Wagner (1980) proved the L^p convergence of $\hat{g}_x(z) := \sum_{j=1}^{n} w_j x_j$ to $g_x(z)$:

A1 $\mathbb{E}[|Z|^p] \leq \infty, \quad p \geq 1.$
A2 $\gamma \to 0$ as $n \to \infty$.
A3 $n\gamma^d \to 0$ as $n \to \infty$.
A4 The kernel K satisfies the following:

- K is a nonnegative function on $\mathbb{R}^d$ bounded by $k^* < \infty$.
- K has a compact support A.
- $k \geq \beta \mathbf{1}_B$ for some $\beta > 0$ and some closed sphere B centered at the origin with positive radius.

Lemma A1 (Devroye and Wagner, 1980)**.** *Under the above assumption, we have*

$$\mathbb{E}_X[\int |\hat{g}_x(z) - g_x(z)|^p d_z] \to 0, \quad \text{as } n \to \infty. \tag{A4}$$

Let $x_i = \mathbf{1}_{x_i<t}$, and notice that

$$\mathbf{1}_{x_i<t} = P(X_i < t) + e_i$$

where $E(e_i) = 0$ and $\sup_i E(e_i^k) \leq 1$ for all $k \geq 2$. This can be viewed as another non-parametric regression model as in (1). We have

$$\mathbb{E}_X[\int |\hat{G}_{x|z} - G_{x|z}|^p dz] \to 0, \quad \text{as} \quad n \to \infty.$$

Using Lemma 1, we proved the L^p convergence of $\hat{g}_x(z)\hat{g}_y(z)$:

Lemma A2.

$$\mathbb{E}_{XY}[\int |\hat{g}_x(z)\hat{g}_y(z) - g_x(z)g_y(z)|^p d_z] \to 0, \quad as \quad n \to \infty. \tag{A5}$$

Proof:

$$\begin{aligned}
&\mathbb{E}_{XY}[\int |\hat{g}_x(z)\hat{g}_y(z) - g_x(z)g_y(z)|^p d_z] \\
=& \mathbb{E}_{XY}[\int |\hat{g}_x(z)\{\hat{g}_y(z) - g_y(z)\} + g_y(z)\{\hat{g}_x(z) - g_x(z)\}|^p d_z] \\
\leq& \mathbb{E}_{XY}[\int |\hat{g}_x(z)|^p |\hat{g}_y(z) - g_y(z)|^p d_z] \\
&+ \mathbb{E}_{XY}[\int |g_y(z)|^p |\hat{g}_x(z) - g_x(z)|^p d_z] \\
\leq& (k^*)^p \left\{ \mathbb{E}_Y[\int |\hat{g}_y(z) - g_y(z)|^p d_z] + \mathbb{E}_X[\int |\hat{g}_x(z) - g_x(z)|^p d_z] \right\}
\end{aligned} \tag{A6}$$

and the right side becomes 0 as $n \to \infty$.

Similarly, we have

$$\mathbb{E}_{XY}[\int |\hat{G}_{x|z}\hat{G}_{y|z} - G_{x|z}G_{y|z}|^p d_z] \to 0, \quad \text{as} \quad n \to \infty, \tag{A7}$$

and

$$|\hat{G}_{x|z^*}\hat{G}_{y|z^*} - G_{x|z^*}G_{y|z^*}| \to 0, \quad as \quad n \to \infty.$$

References

1. Pearl, J. *Causality: Models, Reasoning and Inference*; Cambridge University Press: Cambridge, UK, 2000; Volume 19.
2. Koller, D.; Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*; MIT Press: Cambridge, MA, USA, 2009.
3. Spirtes, P.; Glymour, C.; Acheines, R. *Causation, Prediction, and Search*; The MIT Press: Cambridge, MA, USA, 2000.
4. Margaritis, D. Distribution-Free Learning of Bayesian Network Structure in Continuous Domains. In *AAAI'05: Proceedings of the 20th National Conference on Artificial Intelligence—Volume 2*; AAAI Press: Palo Alto, CA, USA, 2005; pp. 825–830.
5. Lawrance, A.J. On Conditional and Partial Correlation. *Am. Stat.* **1976**, *30*, 146–149. [CrossRef]
6. Bergsma, W.P. *Testing Conditional Independence for Continuous Random Variables*; Eurandom: Eindhoven, The Netherlands, 2004.
7. Fukumizu, K.; Gretton, A.; Sun, X.; Schölkopf, B. Kernel Measures of Conditional Dependence. In *Proceedings of the Advances in Neural Information Processing Systems*; Platt, J., Koller, D., Singer, Y., Roweis, S., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2008; Volume 20.
8. Doran, G.; Muandet, K.; Zhang, K.; Schölkopf, B. A Permutation-Based Kernel Conditional Independence Test. In Proceedings of the UAI, Quebec City, QC, Canada, 23–27 July 2014; pp. 132–141.
9. Huang, T.M. Testing conditional independence using maximal nonlinear conditional correlation. *Ann. Stat.* **2010**, *38*, 2047–2091. [CrossRef]
10. Zhang, K.; Peters, J.; Janzing, D.; Schölkopf, B. Kernel-based conditional independence test and application in causal discovery. *arXiv* **2012**, arXiv:1202.3775.
11. Strobl, E.V.; Zhang, K.; Visweswaran, S. Approximate Kernel-Based Conditional Independence Tests for Fast Non-Parametric Causal Discovery. *J. Causal Inference* **2019**, *7*, 20180017. [CrossRef]
12. Runge, J. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Lanzarote, Spain, 9–11 April 2018; pp. 938–947.

13. Gretton, A.; Fukumizu, K.; Teo, C.H.; Song, L.; Schölkopf, B.; Smola, A.J. A kernel statistical test of independence. *Adv. Neural Inf. Process. Syst.* **2007**, *20*, 585–592.
14. Shi, S.G. Local bootstrap. *Ann. Inst. Stat. Math.* **1991**, *43*, 667–676. [CrossRef]
15. Huang, Z.; Deb, N.; Sen, B. Kernel Partial Correlation Coefficient—A Measure of Conditional Dependence. *arXiv* **2020**, arXiv:2012.14804.
16. Shah, R.D.; Peters, J. The hardness of conditional independence testing and the generalised covariance measure. *Ann. Stat.* **2020**, *48*, 1514–1538. [CrossRef]
17. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
18. Sen, R.; Suresh, A.T.; Shanmugam, K.; Dimakis, A.G.; Shakkottai, S. Model-powered conditional independence test. *arXiv* **2017**, arXiv:1709.06138.
19. Bellot, A.; van der Schaar, M. Conditional independence testing using generative adversarial networks. *Adv. Neural Inf. Process. Syst.* **2019**, 2202–2211.
20. Ahuja, K.; Sattigeri, P.; Shanmugam, K.; Wei, D.; Ramamurthy, K.N.; Kocaoglu, M. Conditionally independent data generation. In Proceedings of the Uncertainty in Artificial Intelligence, Online, 27–30 July 2021; pp. 2050–2060.
21. Shi, C.; Xu, T.; Bergsma, W.; Li, L. Double Generative Adversarial Networks for Conditional Independence Testing. *J. Mach. Learn. Res.* **2021**, *22*, 285–1.
22. Shimizu, S.; Hoyer, P.O.; Hyvärinen, A.; Kerminen, A.; Jordan, M. A linear non-Gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.* **2006**, *7*, 2003–2030.
23. Zhang, K.; Hyvarinen, A. On the identifiability of the post-nonlinear causal model. *arXiv* **2012**, arXiv:1205.2599.
24. Zhang, H.; Zhou, S.; Zhang, K.; Guan, J. Causal Discovery Using Regression-Based Conditional Independence Tests. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 1250–1256.
25. Zhang, Q.; Filippi, S.; Flaxman, S.; Sejdinovic, D. Feature-to-Feature Regression for a Two-Step Conditional Independence Test. In Proceedings of the Uncertainty in Artificial Intelligence, Sydney, Australia, 11–15 August 2017.
26. Zhang, H.; Zhang, K.; Zhou, S.; Guan, J.; Zhang, J. Testing Independence Between Linear Combinations for Causal Discovery. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 6538–6546.
27. Li, C.; Fan, X. On nonparametric conditional independence tests for continuous variables. In *Wiley Interdisciplinary Reviews: Computational Statistics*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2019; Volume 12. [CrossRef]
28. Suzuki, J. *Kernel Methods for Machine Learning with R*; Springer: Berlin/Heidelberg, Germany, 2022.
29. Zhang, Q.; Filippi, S.; Gretton, A.; Sejdinovic, D. Large-scale kernel methods for independence testing. *Stat. Comput.* **2018**, *28*, 113–130. [CrossRef]
30. Devroye, L.P.; Wagner, T.J. Distribution-Free Consistency Results in Nonparametric Discrimination and Regression Function Estimation. *Ann. Stat.* **1980**, *8*, 231–239. [CrossRef]
31. Nadaraya, E.A. On estimating regression. *Theory Probab. Appl.* **1964**, *9*, 141–142. [CrossRef]