CCMI : Classifier based Conditional Mutual Information Estimation

Sudipto Mukherjee, Himanshu Asnani, Sreeram Kannan
Electrical and Computer Engineering,
University of Washington, Seattle, WA.
{sudipm, asnani, ksreeram}@uw.edu

Abstract

Conditional Mutual Information (CMI) is a measure of conditional dependence between random variables X and Y, given another random variable Z. It can be used to quantify conditional dependence among variables in many data-driven inference problems such as graphical models, causal learning, feature selection and time-series analysis. While k-nearest neighbor (kNN) based estimators as well as kernel-based methods have been widely used for CMI estimation, they suffer severely from the curse of dimensionality. In this paper, we leverage advances in classifiers and generative models to design methods for CMI estimation. Specifically, we introduce an estimator for KL-Divergence based on the likelihood ratio by training a classifier to distinguish the observed joint distribution from the product distribution. We then show how to construct several CMI estimators using this basic divergence estimator by drawing ideas from conditional generative models. We demonstrate that the estimates from our proposed approaches do not degrade in performance with increasing dimension and obtain significant improvement over the widely used KSG estimator. Finally, as an application of accurate CMI estimation, we use our best estimator for conditional independence testing and achieve superior performance than the state-of-the-art tester on both simulated and real data-sets.

1 INTRODUCTION

Conditional mutual information (CMI) is a fundamental information theoretic quantity that extends the nice properties of mutual information (MI) in conditional settings. For three continuous random variables, X, Y and Z, the conditional mutual information is defined as:

$$I(X;Y|Z) = \iiint p(x,y,z) \log \frac{p(x,y,z)}{p(x,z)p(y|z)} dxdydz$$

assuming that the distributions admit the respective densities $p(\cdot)$. One of the striking features of MI and CMI is that they can capture non-linear dependencies between the variables. In scenarios where Pearson correlation is zero even when the two random variables are dependent, mutual information can recover the truth. Likewise, in the sense of conditional independence for the case of three random variables X,Y and Z, conditional mutual information provides strong guarantees, i.e., $X \perp Y|Z \iff I(X;Y|Z) = 0$.

The conditional setting is even more interesting as dependence between X and Y can potentially change based on how they are connected to the conditioning variable. For instance, consider a simple Markov chain where $X \to Z \to Y$. Here, $X \perp Y|Z$. But a slightly different relation $X \to Z \leftarrow Y$ has $X \not\perp Y|Z$, even though X and Y may be independent as a pair. It is a well known fact in Bayesian networks that a node is independent of its non-descendants given its parents. CMI goes beyond stating whether the pair (X, Y) is conditionally dependent or not. It also provides a quantitative strength of dependence.

1.1 PRIOR ART

The literature is replete with works aimed at applying CMI for data-driven knowledge discovery. Fleuret (2004) used CMI for fast binary feature selection to improve classification accuracy. Loeckx et al. (2010) improved non-rigid image registration by using CMI as a similarity measure instead of global mutual information. CMI has been used to infer gene-regulatory networks (Liang and Wang 2008) or protein modulation (Giorgi et al. 2014) from gene expression data. Causal discovery (Li et al. 2011; Hlinka et al. 2013; Vejmelka and Paluš 2008) is yet another application area of CMI estimation.

Despite its wide-spread use, estimation of conditional mutual information remains a challenge. One naive method may be to estimate the joint and conditional densities from data and plug it into the expression for CMI. But density estimation is not sample efficient and is often more difficult than estimating the quantities directly. The most widely used technique expresses CMI in terms of appropriate arithmetic of differential entropy estimators (referred to here as ΣH estimator): $I(X;Y|Z) = h(X,Z)+h(Y,Z)-h(Z)-h(X,Y,Z)$, where $h(X) = -\int_{\mathcal{X}} p(x)\log p(x)\,dx$ is known as the differential entropy.

The differential entropy estimation problem has been studied extensively by Beirlant et al. (1997); Nemenman et al. (2002); Miller (2003); Lee (2010); Leśniewicz (2014); Sricharan et al. (2012); Singh and Póczos (2014) and can be estimated either based on kernel-density (Kandasamy et al. 2015; Gao et al. 2016) or k-nearest-neighbor estimates (Sricharan et al. 2013; Jiao et al. 2018; Pál et al. 2010; Kozachenko and Leonenko 1987; Singh et al. 2003; Singh and Póczos 2016). Building on top of k-nearest-neighbor estimates and breaking the paradigm of ΣH estimation, a coupled estimator (which we address henceforth as KSG) was proposed by Kraskov et al. (2004). It generalizes to mutual information, conditional mutual information as well as for other multivariate information measures, including estimation in scenarios when the distribution can be mixed (Runge 2018; Frenzel and Pompe 2007; Gao et al. 2017, 2018; Vejmelka and Paluš 2008; Rahimzamani et al. 2018).

The kNN approach has the advantage that it can naturally adapt to the data density and does not require extensive tuning of kernel band-widths. However, all these approaches suffer from the curse of dimensionality and are unable to scale well with dimensions. Moreover, Gao et al. (2015) showed that exponentially many samples are required (as MI grows) for the accurate estimation using kNN based estimators. This brings us to the central motivation of this work : ***Can we propose estimators for conditional mutual information that estimate well even in high dimensions ?***

1.2 OUR CONTRIBUTIONS

In this paper, we explore various ways of estimating CMI by leveraging tools from classifiers and generative models. To the best of our knowledge, this is the first work that deviates from the framework of kNN and kernel based CMI estimation and introduces neural networks to solve this problem. The main contributions of the paper can be summarized as follows :

Classifier Based MI Estimation: We propose a novel KL-divergence estimator based on classifier two-sample approach that is more stable and performs superior to the recent neural methods (Belghazi et al. 2018).

Divergence Based CMI Estimation: We express CMI as the KL-divergence between two distributions $p_{xyz} = p(z)p(x|z)p(y|x,z)$ and $q_{xyz} = p(z)p(x|z)p(y|z)$, and explore candidate generators for obtaining samples from $q(\cdot)$. The CMI estimate is then obtained from the divergence estimator.

Difference Based CMI Estimation: Using the improved MI estimates, and the difference relation $I(X;Y|Z) = I(X;YZ) - I(X;Z)$, we show that estimating CMI using a difference of two MI estimates performs best among several other proposed methods in this paper such as divergence based CMI estimation and KSG.

Improved Performance in High Dimensions: On both linear and non-linear data-sets, all our estimators perform significantly better than KSG. Surprisingly, our estimators perform well even for dimensions as high as 100, while KSG fails to obtain reasonable estimates even beyond 5 dimensions.

Improved Performance in Conditional Independence Testing: As an application of CMI estimation, we use our best estimator for conditional independence testing (CIT) and obtain improved performance compared to the state-of-the-art CIT tester on both synthetic and real data-sets.

2 ESTIMATION OF CONDITIONAL MUTUAL INFORMATION

The CMI estimation problem from finite samples can be stated as follows. Let us consider three random variables X, Y, $Z \sim p(x,y,z)$, where $p(x,y,z)$ is the joint distribution. Let the dimensions of the random variables be d_x, d_y and d_z respectively. We are given n samples $\{(x_i,y_i,z_i)\}_{i=1}^n$ drawn i.i.d from $p(x,y,z)$. So $x_i \in \mathbb{R}^{d_x}, y_i \in \mathbb{R}^{d_y}$ and $z_i \in \mathbb{R}^{d_z}$. The goal is to estimate $I(X;Y|Z)$ from these n samples.

2.1 DIVERGENCE BASED CMI ESTIMATION

Definition 1. *The Kullback-Leibler (KL) divergence between two distributions $p(\cdot)$ and $q(\cdot)$ is given as :*

$$D_{KL}(p||q) = \int p(x)\log\frac{p(x)}{q(x)}\,dx$$

Definition 2. *Conditional Mutual Information (CMI) can be expressed as a KL-divergence between two distributions $p(x,y,z)$ and $q(x,y,z) = p(x,z)p(y|z)$, i.e.,*

$$I(X;Y|Z) = D_{KL}(p(x,y,z)||p(x,z)p(y|z))$$

The definition of CMI as a KL-divergence naturally leads to the question : *Can we estimate CMI using an estimator for divergence ?* However, the problem is still non-trivial since we are only given samples from $p(x,y,z)$ and the divergence estimator would also require samples

from $p(x,z)p(y|z)$. This further boils down to whether we can learn the distribution $p(y|z)$.

2.1.1 Generative Models

We now explore various techniques to learn the conditional distribution $p(y|z)$ given samples $\sim\ p(x,y,z)$. This problem is fundamentally different from drawing independent samples from the marginals $p(x)$ and $p(y)$, given the joint $p(x,y)$. In this simpler setting, we can simply permute the data to obtain $\{x_i, y_{\pi(i)}\}_{i=1}^n$ (π denotes a permutation, $\pi(i) \neq i$). This would emulate samples drawn from $q(x,y) = p(x)p(y)$. But, such a permutation scheme does not work for $p(x,y,z)$ since it would destroy the dependence between X and Z. The problem is solved using recent advances in generative models which aim to learn an unknown underlying distribution from samples.

Conditional Generative Adversarial Network (CGAN): There exist extensions of the basic GAN framework (Goodfellow et al. 2014) in conditional settings, CGAN (Mirza and Osindero 2014). Once trained, the CGAN can then generate samples from the generator network as $y = \mathcal{G}(s,z), s \sim p(s), z \sim p(z)$.

Conditional Variational Autoencoder (CVAE): Similar to CGAN, the conditional setting, CVAE (Kingma and Welling 2014) (Sohn et al. 2015), aims to maximize the conditional log-likelihood. The input to the decoder network is the value of z and the latent vector s sampled from standard Gaussian. The decoder Q gives the conditional mean and conditional variance (parametric functions of s and z) from which y is then sampled.

kNN based permutation: A simpler algorithm for generating the conditional $p(y|z)$ is to permute data values where $z_i \approx z_j$. Such methods are popular in conditional independence testing literature (Sen et al. 2017; Doran et al.). For a given point $\{x_i, y_i, z_i\}$, we find the k-nearest neighbor of z_i. Let us say it is z_j with the corresponding data point as $\{x_j, y_j, z_j\}$. Then $\{x_i, y_j, z_i\}$ is a sample from $q(x,y,z)$.

Now that we have outlined multiple techniques for estimating $p(y|z)$, we next proceed to the problem of estimating KL-divergence.

2.1.2 Divergence Estimation

Recently, Belghazi et al. (2018) proposed a neural network based estimator of mutual information (MINE) by utilizing lower bounds on KL-divergence. Since MI is a special case of KL-divergence, their neural estimator can be extended for divergence estimation as well. The estimator can be trained using back-propagation and was shown to out-perform traditional methods for MI estimation. The core idea of MINE is cradled in a dual representation of KL-divergence. The two main lower bounds used by MINE are stated below.

Definition 3. *The Donsker-Varadhan representation expresses KL-divergence as a supremum over functions,*

$$D_{KL}(p||q) = \sup_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{x\sim p}[f(x)] - \log(\mathop{\mathbb{E}}_{x\sim q}[\exp(f(x))]) \tag{1}$$

where the function class $\mathcal{F}$ includes those functions that lead to finite values of the expectations.

Definition 4. *The f-divergence bound gives a lower bound on the KL-divergence:*

$$D_{KL}(p||q) \geq \sup_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{x\sim p}[f(x)] - \mathop{\mathbb{E}}_{x\sim q}[\exp(f(x)-1)] \tag{2}$$

MINE uses a neural network f_θ to represent the function class $\mathcal{F}$ and uses gradient descent to maximize the RHS in the above bounds.

Even though this framework is flexible and straightforward to apply, it presents several practical limitations. The estimation is very sensitive to choices of hyper-parameters (hidden-units/layers) and training steps (batch size, learning rate). We found the optimization process to be unstable and to diverge at high dimensions (Section 4. Experimental Results). Our findings resonate those by Poole et al. (2019) in which the authors found the networks difficult to tune even in toy problems.

2.2 DIFFERENCE BASED CMI ESTIMATION

Another seemingly simple approach to estimate CMI could be to express it as a difference of two mutual information terms by invoking the chain rule, i.e.: $I(X;Y|Z) = I(X;Y,Z) - I(X;Z)$. As stated before, since mutual information is a special case of KL-divergence, viz. $I(X;Y) = D_{KL}(p(x,y)||p(x)p(y))$, this again calls for a stable, scalable, sample efficient KL-divergence estimator as we present in the next Section.

3 CLASSIFIER BASED MI ESTIMATION

In their seminal work on independence testing, Lopez-Paz and Oquab (2017) introduced classifier two-sample test to distinguish between samples coming from two unknown distributions p and q. The idea was also adopted for conditional independence testing by Sen et al. (2017). The basic principle is to train a binary classifier by labeling samples $x \sim p$ as 1 and those coming from $x \sim q$ as 0, and to test the null hypothesis $\mathcal{H}_0 : p = q$. Under the null, the accuracy of the binary classifier will be close to 0.5. It will be away from 0.5 under the alternative. The accuracy of the binary classifier can then be carefully used to define P-values for the test.

We propose to use the classier two-sample principle for estimating the likelihood ratio $\frac{p(x,y)}{p(x)p(y)}$. While existing literature has instances of using the likelihood ratio for MI estimation, the algorithms to estimate the likelihood ratio are quite different from ours. Both (Suzuki et al. 2008; Nguyen et al. 2008) formulate the likelihood ratio estimation as a convex relaxation by leveraging the Legendre-Fenchel duality. But performance of the methods depend on the choice of suitable kernels and would suffer from the same disadvantages as mentioned in the Introduction.

3.1 PROBLEM FORMULATION

Given n i.i.d samples $\{x_i^p\}_{i=1}^n, x_i^p \sim p(x)$ and m i.i.d samples $\{x_j^q\}_{j=1}^m, x_j^q \sim q(x)$, we want to estimate $D_{KL}(p||q)$. We label the points drawn from $p(\cdot)$ as $y = 1$ and those from $q(\cdot)$ as $y = 0$. A binary classifier is then trained on this supervised classification task. Let the prediction for a point l by the classifier is γ_l where $\gamma_l = Pr(y = 1|x_l)$ (Pr denotes probability). Then the point-wise likelihood ratio for data point l is given by $\mathcal{L}(x_l) = \frac{\gamma_l}{1-\gamma_l}$.

The following Proposition is elementary and has already been observed in Belghazi et al. (2018)(Proof of Theorem 4). We restate it here for completeness and quick reference.

Proposition 1. *The optimal function in Donsker-Varadhan representation* (1) *is the one that computes the point-wise log-likelihood ratio, i.e,* $f^*(x) = \log \frac{p(x)}{q(x)} \; \forall \, x$, *(assuming* $p(x) = 0$, *where-ever* $q(x) = 0$*).*

Based on Proposition 1, the next step is to substitute the estimates of point-wise likelihood ratio in (1) to obtain an estimate of KL-divergence.

$$\hat{D}_{KL}(p||q) = \frac{1}{n}\sum_{i=1}^{n} \log \mathcal{L}(x_i^p) - \log\left(\frac{1}{m}\sum_{j=1}^{m} \mathcal{L}(x_j^q)\right) \tag{3}$$

We obtain an estimate of mutual information from (3) as $\hat{I}_n(X;Y) = \hat{D}_{KL}(p(x,y)||p(x)p(y))$. This classifier-based estimator for MI (Classifier-MI) has the following theoretical properties under Assumptions (A1)-(A4) (stated in the Supplementary).

Theorem 1. *Under Assumptions (A1)-(A4), Classifier-MI is consistent, i.e., given* $\epsilon, \delta > 0, \exists\, n \in \mathbb{N}$, *such that with probability at least* $1 - \delta$, *we have*

$$|\hat{I}_n(X;Y) - I(X;Y)| \leq \epsilon$$

Proof. Here, we provide a sketch of the proof. The classifier is trained to minimize the binary cross entropy (BCE) loss on the train set and obtains the minimizer as $\hat{\theta}$. From generalization bound of classifier, the loss value on the test set from $\hat{\theta}$ is close to the loss obtained by the best optimizer in the classifier family, which itself is close to the global minimizer $\gamma*$ of BCE (as a function γ) by Universal Function Approximation Theorem of neural-networks.

The BCE loss is strongly convex in γ. γ links BCE to $I(\cdot\,;\,\cdot)$, i.e., $|\mathrm{BCE}_n(\gamma_{\hat{\theta}}) - \mathrm{BCE}(\gamma^*)| \leq \epsilon' \implies \|\gamma_{\hat{\theta}} - \gamma^*\|_1 \leq \eta \implies |\hat{I}_n(X;Y) - I(X;Y)| \leq \epsilon$. □

While consistency provides a characterization of the estimator in large sample regime, it is not clear what guarantees we obtain for finite samples. The following Theorem shows that even for a small number of samples, the produced MI estimate is a true lower bound on mutual information value with high probability.

Theorem 2. *Under Assumptions (A1)-(A4), the finite sample estimate from Classifier-MI is a lower bound on the true MI value with high probability, i.e., given* n *test samples, we have for* $\epsilon > 0$

$$Pr(I(X;Y) + \epsilon \geq \hat{I}_n(X;Y)) \geq 1 - 2\exp(-Cn)$$

where C *is some constant independent of* n *and the dimension of the data.*

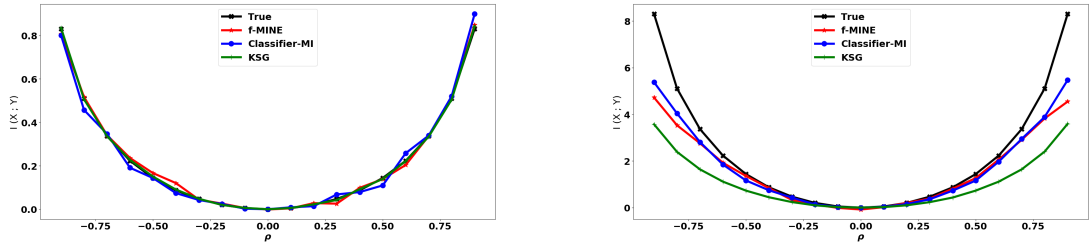
3.2 PROBABILITY CALIBRATION

The estimation of likelihood ratio from classifier predictions $Pr(y = 1|x)$ hinges on the fact that the classifier is well-calibrated. As a rule of thumb, classifiers trained directly on the cross entropy loss are well-calibrated. But boosted decision trees would introduce distortions in the likelihood-ratio estimates. There is an extensive literature devoted to obtaining better calibrated classifiers that can be used to improve the estimation further (Lakshminarayanan et al. 2017; Niculescu-Mizil and Caruana 2005; Guo et al. 2017). We experimented with Gradient Boosted Decision Trees and multi-layer perceptron trained on the log-loss in our algorithms. Multi-layer perceptron gave better estimates and so is used in all the experiments. Supplementary Figures show that the neural networks used in our estimators are well-calibrated.

Even though logistic regression is well-calibrated and might seem to be an attractive candidate for classification in sparse sample regimes, we show that linear classifiers cannot be used to estimate D_{KL} by two-sample approach. For this, we consider the simple setting of estimating mutual information of two correlated Gaussian random variables as a counter-example.

Lemma 1. *A linear classifier with marginal features fails the classifier Two sample MI estimation.*

Proof. Consider two correlated Gaussians in 2 dimensions $(X_1, X_2) \sim \mathcal{N}\left(0, M = \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)\right)$, where ρ is the Pearson correlation. The marginals are standard Gaussans $X_i \sim \mathcal{N}(0,1)$. Suppose we are trying to estimate the mutual information $D_{KL}(p(x_1,x_2)||p(x_1)p(x_2))$.



(a) $d_x = d_y = 1$ (b) $d_x = d_y = 10$

Figure 1: Mutual Information Estimation of Correlated Gaussians : In this setting, X and Y have independent coordinates, with $(X_i, Y_i)\ \forall\, i$ being correlated Gaussians with correlation coefficient ρ. $I^*(X;Y) = -\frac{1}{2} d_x \log(1-\rho^2)$

The classifier decision boundary would seek to find $Pr(y = 1|x_1, x_2) > Pr(y = 0|x_1, x_2)$, thus $p(x_1, x_2) > p(x_1)p(x_2) => x_1 x_2 > \frac{1}{2\rho} \log(1-\rho^2)$ □

The decision boundary is a rectangular hyperbola. Here the classifier would return 0.5 as prediction for either class (leading to $\hat{D}_{KL} = 0$), even when X_1 and X_2 are highly correlated and the mutual information is high.

We use the Classifier two-sample estimator to first compute the mutual information of two correlated Gaussians (Belghazi et al. 2018) for $n = 5,000$ samples. This setting also provides us a way to choose reasonable hyperparameters that are used throughout in all the synthetic experiments. We also plot the estimates of f-MINE and KSG to ensure we are able to make them work in simple settings. In the toy setting $d_x = 1$, all estimators accurately estimate $I(X;Y)$ as shown in Figure 1.

3.3 MODULAR APPROACH TO CMI ESTIMATION

Our classifier based divergence estimator does not encounter an optimization problem involving exponentials. MINE optimizing (1) has biased gradients while that based on (2) is a weaker lower bound (Belghazi et al. 2018). On the contrary, our classifier is trained on cross-entropy loss which has unbiased gradients. Furthermore, we plug in the likelihood ratio estimates into the tighter Donsker-Varadhan bound, thereby, achieving the best of both worlds. Equipped with a KL-divergence estimator, we can now couple it with the generators or use the expression of CMI as a difference of two MIs (which we address from now as MI-Diff.). Algorithm 1 describes the CMI estimation by tying together the generator and Classifier block. For MI-Diff., function block "Classifier-D_{KL}" in Algorithm 1 has to be used twice : once for estimating $I(X;Y,Z)$ and another for $I(X;Z)$. For mutual information, $\mathcal{D}_q$ in "Classifier-D_{KL}" is obtained by permuting the samples of $p(\cdot)$.

For the Classifier coupled with a generator, the generated distribution $g(y|z)$ may deviate from the target distribution $p(y|z)$ - introducing a different kind of bias. The following Lemma suggests how such a bias can be corrected by subtracting the KL divergence of the sub-tuple (Y,Z) from the divergence of the entire triple (X,Y,Z). We note that such a clean relationship is not true for general divergence measures, and indeed require more sophisticated conditions for the total-variation metric (Sen et al. 2018).

Lemma 2 (Bias Cancellation)**.** *The estimation error due to incorrect generated distribution $g(y|z)$ can be accounted for using the following relation :*

$$D_{KL}(p(x,y,z)||p(x,z)p(y|z)) =$$
$$D_{KL}(p(x,y,z)||p(x,z)g(y|z)) -$$
$$D_{KL}(p(y,z)||p(z)g(y|z))$$

4 EXPERIMENTS

In this Section, we compare the performance of various estimators on the CMI estimation task. We used the Classifier based divergence estimator (Section 3) and MINE in our experiments. Belghazi et al. (2018) had two MINE variants, namely Donsker-varadhan (DV) MINE and f-MINE. The f-MINE has unbiased gradients and we found it to have similar performance as DV-MINE, albeit with lower variance. So we used f-MINE in all our experiments.

The "generator"+"Divergence estimator" notation will be used to denote the various estimators. For instance, if we use CVAE for the generation and couple it with f-MINE, we denote the estimator as CVAE+f-MINE. When coupled with the Classifier based Divergence block, it will be denoted as CVAE+Classifier. For MI-Diff. we represent it similarly as MI-Diff.+"Divergence estimator".

We compare our estimators with the widely used KSG estimator.[1] For f-MINE, we used the code provided to us by the author (Belghazi et al. 2018). The same

[1]The implementation of CMI estimator in Non-parametric Entropy Estimation Toolbox (`https://github.com/gregversteeg/NPEET`) is used.

Input: Dataset $\mathcal{D} = \{x_i, y_i, z_i\}_{i=1}^{n}$, number of outer boot-strap iterations B, Inner iterations T, clipping constant τ.
Output: CMI estimatate $\hat{I}(X;Y|Z)$
for $b \in \{1, 2, \dots B\}$ **do**
  Permute the points in dataset $\mathcal{D}$ to obtain $\mathcal{D}^{\pi}$.
  Split $\mathcal{D}^{\pi}$ equally into two parts $\mathcal{D}_{\text{class,joint}} = \{x_i, y_i, z_i\}_{i=1}^{n/2}$ and $\mathcal{D}_{\text{gen}} = \{x_i, y_i, z_i\}_{i=n/2}^{n}$.
  Train the generator $\mathcal{G}(\cdot)$ on $\mathcal{D}_{\text{gen}}$.
  Generate the marginal data-set using points $y_i' = \mathcal{G}(z_i) \, \forall \, z_i \in \mathcal{D}_{\text{class,joint}}(:, Z)$. $\mathcal{D}_{\text{class,marg}} = \{x_i, y_i', z_i\}_{i=1}^{n/2}$
  $\hat{I}_b(X;Y|Z) = \text{Classifier\_D}_{\text{KL}}(\mathcal{D}_{\text{class,joint}}, \mathcal{D}_{\text{class,marg}}, T, \tau)$
end
return $\frac{1}{B}\sum_b \hat{I}_b(X;Y|Z)$
Function Classifier_D_{KL} $(\mathcal{D}_p, \mathcal{D}_q, T, \tau)$ **:**
  Label points $u \in \mathcal{D}_p$ as $l = 1$ and $v \in \mathcal{D}_q$ as $l = 0$.
  for $t \in \{1, 2, \dots T\}$ **do**
    $\mathcal{D}_p^{\text{train}}, \mathcal{D}_p^{\text{eval}} \leftarrow$ SPLIT_TEST_TRAIN($\mathcal{D}_p$).
    $\mathcal{D}_q^{\text{train}}, \mathcal{D}_q^{\text{eval}} \leftarrow$ SPLIT_TEST_TRAIN($\mathcal{D}_q$)
    Train classifier $\mathcal{C}$ on $\{\mathcal{D}_p^{\text{train}}, \vec{1}\}, \{\mathcal{D}_q^{\text{train}}, \vec{0}\}$
    Obtain classifier predictions $Pr(l = 1|w) \, \forall \, w \in \mathcal{D}_p^{\text{eval}} \cup \mathcal{D}_q^{\text{eval}}$, and clip to $[\tau, 1-\tau]$.
    $\hat{D}_{KL}^{t}(p||q) \leftarrow \frac{1}{|\mathcal{D}_p^{\text{eval}}|} \sum_{u \in \mathcal{D}_p^{\text{eval}}} \log \frac{Pr(l=1|u)}{1-Pr(l=1|u)} - \log \left(\frac{1}{|\mathcal{D}_q^{\text{eval}}|} \sum_{v \in \mathcal{D}_q^{\text{eval}}} \frac{Pr(l=1|v)}{1-Pr(l=1|v)} \right)$
  end
  return $\hat{D}_{KL}(p||q) = \frac{1}{\tau}\sum_t \hat{D}_{KL}^{t}(p||q)$

Algorithm 1: GENERATOR + CLASSIFIER

hyper-parameter setting is used in all our synthetic data-sets for all estimators (including generators and divergence blocks). Supplementary contains the details about the hyper-parameter values. For KSG, we vary $k \in \{3, 5, 10\}$ and report the results for the best k for each data-set.

4.1 LINEAR RELATIONS

We start with the simple setting where the three random variables X, Y, Z are related in a linear fashion. We consider the following two linear models.

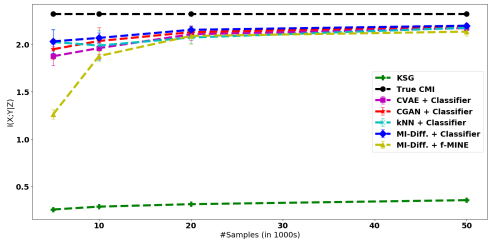
Model I	**Model II**
$X \sim \mathcal{N}(0,1)$	$X \sim \mathcal{N}(0,1)$
$Z \sim \mathcal{U}(-0.5, 0.5)^{d_z}$	$Z \sim \mathcal{N}(0,1)^{d_z}$
	$U = w^T Z, \|w\|_1 = 1$
$\epsilon \sim \mathcal{N}(Z_1, \sigma_\epsilon^2)$	$\epsilon \sim \mathcal{N}(U, \sigma_\epsilon^2)$
$Y \sim X + \epsilon$	$Y \sim X + \epsilon$

Table 1: Linear Models

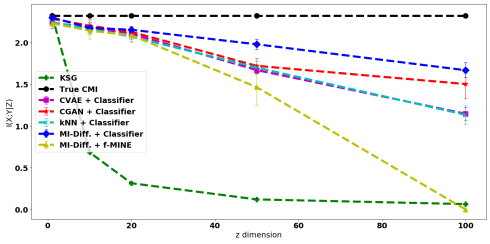
where $\mathcal{U}(-0.5, 0.5)^{d_z}$ means that each co-ordinate of Z is drawn i.i.d from a uniform distribution between -0.5 and 0.5. Similar notation is used for the Gaussian : $\mathcal{N}(0,1)^{d_z}$. Z_1 is the first dimension of Z. We used $\sigma_\epsilon = 0.1$ and obtained the constant unit norm random vector w from $\mathcal{N}(0, I_{d_z})$. w is kept constant for all points during data-set preparation.

As common in literature on causal discovery and independence testing (Sen et al. 2017; Doran et al.), the dimension of X and Y is kept as 1, while d_z can scale. Our estimators are general enough to accommodate multi-dimensional X and Y, where we consider a concatenated vector $X = (X_1, X_2, \dots, X_{d_x})$ and $Y = (Y_1, Y_2, \dots, Y_{d_y})$. This has applications in learning interactions between Modules in Bayesian networks (Segal et al. 2005) or dependence between group variables (Entner and Hoyer 2012; Parviainen and Kaski 2016) such as distinct functional groups of proteins/genes instead of individual entities. Both the linear models are representative of problems encountered in Graphical models and independence testing literature. In Model I, the conditioning set can go on increasing with independent variables $\{Z_k\}_{k=2}^{d_z}$, while Y only depends on Z_1. In Model II, we have the variables in the conditioning set combining linearly to produce Y. It is also easy to obtain the ground truth CMI value in such models by numerical integration.

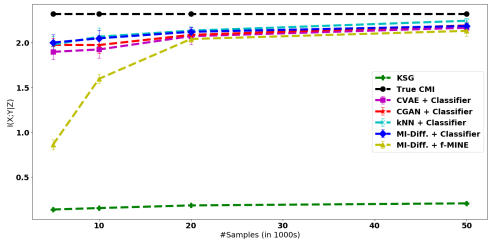
For both these models, we generate data-sets with varying number of samples n and varying dimension d_z to study their effect on estimator performance. The sample size is varied as $n \in \{5000, 10000, 20000, 50000\}$ keeping d_z fixed at 20. We also vary $d_z \in$



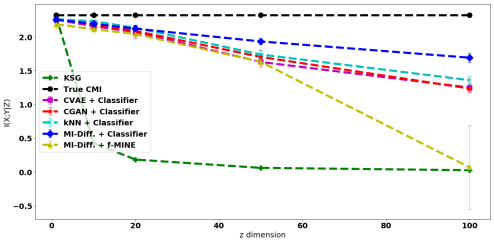
(a) Model I : Variation with n, $d_z = 20$



(b) Model I : Variation with d_z, $n = 20,000$



(c) Model II : Variation with n, $d_z = 20$



(d) Model II : Variation with d_z, $n = 20,000$

Figure 2: CMI Estimation in Linear models : We study the effect of various estimators as either number of samples n or dimension d_z is varied. MI-Diff.+Classifier performs the best among our estimators, while all our proposed estimators improve the estimation significantly over KSG. Average of 10 runs is plotted. Error bars depict 1 standard deviation from mean. (Best viewed in color)

$\{1, 10, 20, 50, 100\}$, keeping sample size fixed at $n = 20000$.

Several observations stand out from the experiments: (1) KSG estimates are accurate at very low dimension but drastically fall with increasing d_z even when the conditioning variables are completely independent and do not influence X and Y (Model-I). (2) Increasing the sample size does not improve KSG estimates once the dimension is kept moderate (even 20!). The dimension issue is more acute than sample scarcity. (3) The estimates from f-MINE have greater deviation from the truth at low sample sizes. At high dimensions, the instability is clearly portrayed when the estimate suddenly goes negative (Truncated to 0.0 to maintain the scale of the plot). (4) All our estimators using Classifier are able to obtain reasonable estimates even at dimensions as high as 100, with MI-Diff.+Classifier performing the best.

4.2 NON-LINEAR RELATIONS

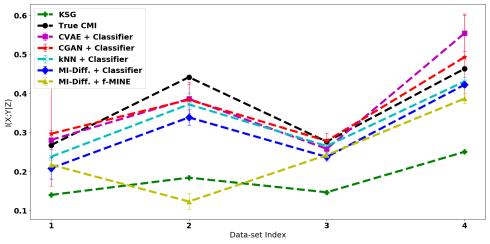
Here, we study models where the underlying relations between X, Y and Z are non-linear. Let $Z \sim \mathcal{N}(\mathbf{1}, I_{d_z}), X = f_1(\eta_1), Y = f_2(A_{zy}Z + A_{xy}X + \eta_2)$. f_1 and f_2 are non-linear bounded functions drawn uniformly at random from $\{cos(\cdot), tanh(\cdot), \exp(-|\cdot|)\}$ for each data-set. A_{zy} is a random vector whose entries are drawn $\mathcal{N}(0, 1)$ and normalized to have unit norm. The vector once generated is kept fixed for a particular data-set. We have the setting where $d_x = d_y = 1$ and d_z can scale. A_{xy} is then a constant. We used $A_{xy} = 2$ in our simulations. The noise variables η_1, η_2 are drawn i.i.d $\mathcal{N}(0, \sigma_\epsilon^2)$, $\sigma_\epsilon^2 = 0.1$.

We vary $n \in \{5000, 10000, 20000, 50000\}$ across each dimension d_z. The dimension d_z itself is then varied as $\{10, 20, 50, 100, 200\}$ giving rise to 20 data-sets. Data-index 1 has $n = 5000, d_z = 10$, data-index 2 has $n = 10000, d_z = 10$ and so on until data-index 20 with $n = 50000, d_z = 200$.

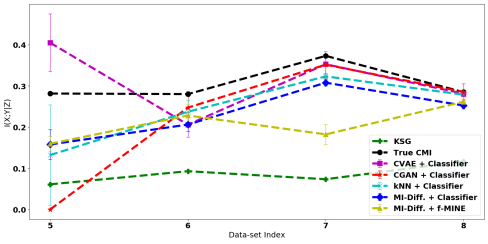
Obtaining Ground Truth $I^*(X;Y|Z)$: Since it is not possible to obtain the ground truth CMI value in such complicated settings using a closed form expression, we resort to using the relation $I(X;Y|Z) = I(X;Y|U)$ where $U = A_{zy}Z$. The dependence of Y on Z can be completely captured once U is given. But, U has dimension 1 and can be estimated accurately using KSG. We generate 50000 samples separately for each data-set to estimate $I(X;Y|U)$ and use it as the ground truth.

We observed similar behavior (as in Linear models) for our estimators in the Non-linear setting.

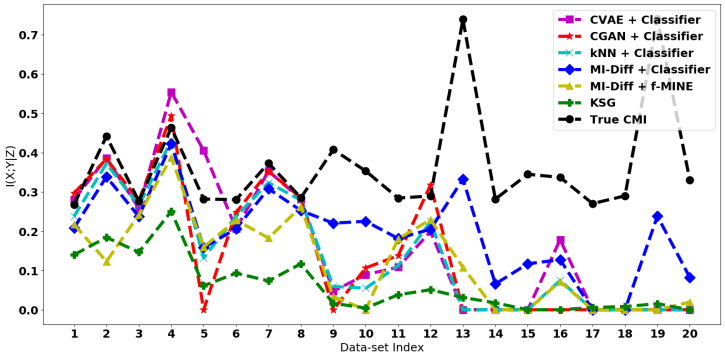
(1) KSG continues to have low estimates even though in this setup the true CMI values are themselves low (< 1.0). (2) Up to $d_z = 20$, we find all our estimators closely tracking $I^*(X;Y|Z)$. But in higher dimensions,



(a) Non-linear Model : Number of samples increase with Data-index, $d_z = 10$ (fixed)



(b) Non-linear Model : Number of samples increase with Data-index, $d_z = 20$ (fixed)



(c) Non-linear Models (All 20 data-sets)

Figure 3: On non-linear data-sets, a similar trend is observed. KSG under-estimates $I^*(X;Y|Z)$, while our estimators track it closely. Average over 10 runs is plotted. (Best viewed in color)

they fail to perform accurately. (3) MI-Diff. + Classifier is again the best estimator, giving CMI estimates away from 0 even at 200 dimensions.

From the above experiments, we found MI-Diff.+Classifier to be the most accurate and stable estimator. We use this combination for our downstream applications and henceforth refer to it as CCMI.

5 APPLICATION TO CONDITIONAL INDEPENDENCE TESTING

As a testimony to accurate CMI estimation, we apply CCMI to the problem of Conditional Independence Testing(CIT). Here, we are given samples from two distributions $p(x, y, z)$ and $q(x, y, z) = p(x, z)p(y|z)$. The hypothesis testing in CIT is to distinguish the null $\mathcal{H}_0 : X \perp Y|Z$ from the alternative $\mathcal{H}_1 : X \not\perp Y|Z$.

We seek to design a CIT tester using CMI estimation by using the fact that $I(X;Y|Z) = 0 \iff X \perp Y|Z$. A simple approach would be to reject the null if $I(X;Y|Z) > 0$ and accept it otherwise. The CMI estimates can serve as a proxy for the P-value. CIT testing based on CMI Estimation has been studied by Runge (2018), where the author uses KSG for CMI estimation and use k-NN based permutation to generate a P-value. The P-value is computed as the fraction of permuted data-sets where the CMI estimate is $\geq$ that of the original data-set. The same approach can be adopted for CCMI to obtain a P-value. But since we report the AuROC (Area under the Receiver Operating Characteristic curve), CMI estimates suffice.

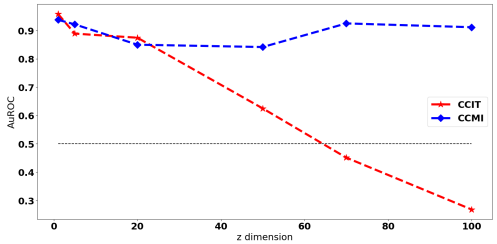
5.1 POST NON-LINEAR NOISE : SYNTHETIC

In this experiment, we generate data based on the post non-linear noise model similar to Sen et al. (2017). As before, $d_x = d_y = 1$ and d_z can scale in dimension. The data is generated using the follow model.

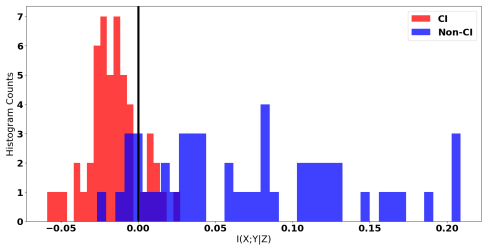
$$Z \sim \mathcal{N}(\mathbf{1}, I_{d_z}),\ X = cos(a_x Z + \eta_1)$$

$$Y = \begin{cases} cos(b_y Z + \eta_2) & \text{if} X \perp Y|Z \\ cos(cX + b_y Z + \eta_2) & \text{if } X \not\perp Y|Z \end{cases}$$

The entries of random vectors(matrices if $d_x, d_y > 1$) a_x and b_y are drawn $\sim \mathcal{U}(0, 1)$ and the vectors are normalized to have unit norm, i.e., $\|a\|_2 = 1, \|b\|_2 = 1$. $c \sim \mathcal{U}[0, 2], \eta_i \sim \mathcal{N}(0, \sigma_e^2), \sigma_e = 0.5$. This is different from the implementation in Sen et al. (2017) where the constant is $c = 2$ in all data-sets. But by varying c, we obtain a tougher problem where the true CMI value



(a) CCIT performance degrades with increasing dz; CCMI retains high AuROC score even at dz = 100.



(b) Estimates for CI data-sets are ≤ 0 and those for non-CI are > 0 at $d_z = 100$. Thresholding CMI estimates at 0 yields Precision = 0.84, Recall = 0.86.

Figure 4: Conditional Independence Testing in Post Non-linear Synthetic Data-set

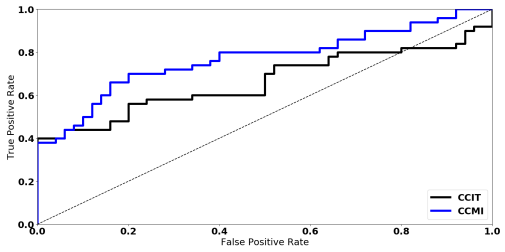


Figure 5: AuROC Curves : Flow-Cytometry Data-set. CCIT obtains a mean AuROC score of 0.6665, while CCMI out-performs with mean of 0.7569.

can be quite low for a dependent data-set and the tester is required to separate it correctly from 0.

a_x, b_y and c are kept constant for generating points for a single data-set and are varied across data-sets. We vary $d_z \in \{1, 5, 20, 50, 70, 100\}$ and simulate 100 data-sets for each dimension. The number of samples is $n = 5000$ in each data-set. Our algorithm is compared with the state-of-the-art CIT tester in Sen et al. (2017), known as CCIT. We used the implementation provided by the authors and ran CCIT with $B = 50$ bootstraps [2]. For each data-set, an AuROC value is obtained. Figure 4 shows the mean AuROC values from 5 runs for both the testers as d_z varies. While both algorithms perform accurately upto $d_z = 20$, the performance of CCIT starts to degrade beyond 20 dimensions. Beyond 50 dimensions, it performs close to random guessing. CCMI retains its superior performance even at $d_z = 100$, obtaining a mean AuROC value of 0.91.

Since AuROC metric finds best performance by varying thresholds, it is not clear what precision and recall is obtained from CCMI when we threshold the CCMI estimate at 0 (and reject or accept the null based on it). So, for $d_z = 100$ we plotted the histogram of CMI estimates separately for CI and non-CI data-sets. Figure 4b shows that there a clear demarcation of CMI estimates between the two data-set categories and choosing the threshold as 0.0 gave the precision as 0.84 and recall as 0.86.

[2] https://github.com/rajatsen91/CCIT

5.2 FLOW-CYTOMETRY : REAL DATA

To extend our estimator beyond simulated settings, we use CMI estimation to test for conditional independence in the protein network data used in Sen et al. (2017). The consensus graph in Sachs et al. (2005) is used as the ground truth. We obtained 50 CI and 50 non-CI relations from the Bayesian network. The basic philosophy used is that a protein X is independent of all other proteins Y in the network given its parents, children and parents of children. Moreover, in the case of non-CI, we notice that a direct edge between X and Y would never render them conditionally independent. So the conditioning set Z can be chosen at random from other proteins. These two settings are used to obtain the CI and non-CI data-sets. The number of samples in each data-set is only 853 and the dimension of Z varies from 5 to 7. CCMI is compared with CCIT and the mean AuROC curves from 5 runs is plotted in Figure 5. The superior performance of CCMI over CCIT is retained in sparse data regime.

6 CONCLUSION

In this work we explored various CMI estimators by drawing from recent advances in generative models and classifies. We proposed a new divergence estimator, based on Classifier-based two-sample estimation, and built several conditional mutual information estimators using this primitive. We demonstrated their efficacy in a variety of practical settings. Future work will aim to approximate the null distribution for CCMI, so that we can compute P-values for the conditional independence testing problem efficiently.

Acknowledgments

This work was partially supported by NSF grants 1651236 and 1703403 and NIH grant 1R01HG008164.

References

Jan Beirlant, Edward J Dudewicz, László Györfi, and Edward C Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

G Doran, K Muandet, K Zhang, and B Schölkopf. A permutation-based kernel conditional independence test. In *30th Conference on Uncertainty in Artificial Intelligence (UAI 2014)*.

Doris Entner and Patrik O Hoyer. Estimating a causal order among groups of variables in linear models. In *International Conference on Artificial Neural Networks*, pages 84–91. Springer, 2012.

François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine learning research*, 5(Nov):1531–1555, 2004.

Stefan Frenzel and Bernd Pompe. Partial mutual information for coupling analysis of multivariate time series. *Physical review letters*, 99(20):204101, 2007.

Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial Intelligence and Statistics*, pages 277–286, 2015.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Breaking the bandwidth barrier: Geometrical adaptive entropy estimation. In *Advances in Neural Information Processing Systems*, pages 2460–2468, 2016.

Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Estimating mutual information for discrete-continuous mixtures. In *Advances in Neural Information Processing Systems*, pages 5988–5999, 2017.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE Transactions on Information Theory*, 64 (8):5629–5661, 2018.

Federico M Giorgi, Gonzalo Lopez, Jung H Woo, Brygida Bisikirska, Andrea Califano, and Mukesh Bansal. Inferring protein modulation from gene expression data using conditional mutual information. *PloS one*, 9(10):e109569, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

Jaroslav Hlinka, David Hartman, Martin Vejmelka, Jakob Runge, Norbert Marwan, Jürgen Kurths, and Milan Paluš. Reliability of inference of directed climate networks using conditional mutual information. *Entropy*, 15(6):2023–2045, 2013.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Jiantao Jiao, Weihao Gao, and Yanjun Han. The nearest neighbor information estimator is adaptively near minimax rate-optimal. In *Advances in neural information processing systems*, 2018.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems*, pages 397–405, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR*, 2014.

LF Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Intae Lee. Sample-spacings-based density and entropy estimators for spherically invariant multidimensional data. *Neural Computation*, 22(8):2208–2227, 2010.

Marek Leśniewicz. Expected entropy as a measure and criterion of randomness of binary sequences. *Przegląd Elektrotechniczny*, 90(1):42–46, 2014.

Zhaohui Li, Gaoxiang Ouyang, Duan Li, and Xiaoli Li. Characterization of the causality between spike trains with permutation conditional mutual information. *Physical Review E*, 84(2):021929, 2011.

Kuo-Ching Liang and Xiaodong Wang. Gene regulatory network reconstruction using conditional mutual information. *EURASIP Journal on Bioinformatics and Systems Biology*, 2008(1):253894, 2008.

Dirk Loeckx, Pieter Slagmolen, Frederik Maes, Dirk Vandermeulen, and Paul Suetens. Nonrigid image registration using conditional mutual information. *IEEE transactions on medical imaging*, 29(1):19–29, 2010.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *5th International Conference on Learning Representations, ICLR*, 2017.

Erik G Miller. A new class of entropy estimators for multi-dimensional densities. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–297. IEEE, 2003.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. 2018.

Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited. In *Advances in neural information processing systems*, pages 471–478, 2002.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Advances in neural information processing systems*, pages 1089–1096, 2008.

Alexandru Niculescu-Mizil and Rich Caruana. Obtaining calibrated probabilities from boosting. In *UAI*, 2005.

Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *Advances in Neural Information Processing Systems*, pages 1849–1857, 2010.

Pekka Parviainen and Samuel Kaski. Bayesian networks for variable groups. In *Conference on Probabilistic Graphical Models*, pages 380–391, 2016.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Arman Rahimzamani, Himanshu Asnani, Pramod Viswanath, and Sreeram Kannan. Estimators for multivariate information measures in general probability spaces. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

Jakob Runge. 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*, 2018.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Eran Segal, Dana PeâĂŹer, Aviv Regev, Daphne Koller, and Nir Friedman. Learning module networks. *Journal of Machine Learning Research*, 6(Apr):557–588, 2005.

Rajat Sen, Ananda Theertha Suresh, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Model-powered conditional independence test. In *Advances in Neural Information Processing Systems*, pages 2951–2961, 2017.

Rajat Sen, Karthikeyan Shanmugam, Himanshu Asnani, Arman Rahimzamani, and Sreeram Kannan. Mimic and classify: A meta-algorithm for conditional independence testing. *arXiv preprint arXiv:1806.09708*, 2018.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4): 301–321, 2003.

Shashank Singh and Barnabás Póczos. Exponential concentration of a density functional estimator. In *Advances in Neural Information Processing Systems*, pages 3032–3040, 2014.

Shashank Singh and Barnabás Póczos. Finite-sample analysis of fixed-k nearest neighbor density functional estimators. In *Advances in Neural Information Processing Systems*, pages 1217–1225, 2016.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28*. 2015.

Kumar Sricharan, Raviv Raich, and Alfred O Hero. Estimation of nonlinear functionals of densities with confidence. *IEEE Transactions on Information Theory*, 58 (7):4135–4159, 2012.

Kumar Sricharan, Dennis Wei, and Alfred O Hero. Ensemble estimators for multivariate entropy estimation. *IEEE transactions on information theory*, 59(7):4374–4388, 2013.

Taiji Suzuki, Masashi Sugiyama, Jun Sese, and Takafumi Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. In *New challenges for feature selection in data mining and knowledge discovery*, pages 5–20, 2008.

Martin Vejmelka and Milan Paluš. Inferring the directionality of coupling with conditional mutual information. *Physical Review E*, 77(2):026214, 2008.