EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

An experimental study of neural estimators of the mutual information between random vectors modeling power spectrum features

Donghoon Shin[1,2] and Hyung Soon Kim[1*]

*Correspondence: kimhs@pusan.ac.kr

[1] Department of Electronics Engineering, Pusan Natoinal University, Busan, South Korea
[2] Maritime Technology Research Institute, Agency for Defense Development, Changwon, South Korea

Abstract

Mutual information (MI) quantifies the statistical dependency between a pair of random variables and plays a central role in signal processing and data analysis. Recent advances in machine learning have enabled the estimation of MI from a dataset using the expressive power of neural networks. In this study, we conducted a comparative experimental analysis of several existing neural estimators of MI between random vectors that model power spectrum features. We explored alternative models of power spectrum features by leveraging information-theoretic data processing inequality and bijective transformations. Empirical results demonstrated that each neural estimator of MI covered in this study has its limitations. In practical applications, we recommend the collective use of existing neural estimators in a complementary manner for the problem of estimating MI between power spectrum features.

Keywords: Mutual information, Estimation, Neural network, Power spectrum

1 Introduction

Mutual information (MI) is a measure of the amount of information that one random variable X contains about another random variable Y. Formally, the MI between X and Y, denoted $I(X, Y)$, is the Kullback–Leibler (KL) divergence between the joint probability density $p(x, y)$ and the product distribution $p(x)p(y)$ of marginal densities [1],

$$\begin{aligned} I(X,Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right] \\ &= \int p(x,y)\log\frac{p(x,y)}{p(x)p(y)}\mathrm{d}x\mathrm{d}y. \end{aligned} \tag{1}$$

where, with the base of the logarithm e, the entropy is measured in *nats*.

A processing stage which maximizes the mutual information between its output and its input mixed with noise, is a way to extract useful input features, and provides a model for perceptual functions [2–5]. This infomax principle forms the basis for applications such as speech recognition [6] and blind source separation [5]. In each application, learning rules are devised to maximize MI.

Recent advances in MI estimation and optimization using neural networks have enabled unsupervised learning of representations based on the infomax principle [7, 8] and contrastive predictive coding [9]. The latter combines predictive coding with a probabilistic contrastive loss dependent on MI [9]. These applications involve neural networks that estimate MI from a finite dataset without prior knowledge of the underlying distribution.

However, measuring MI from a finite dataset poses challenges due to inherent statistical limitations. It becomes impractical to obtain a high-confidence variational lower bound for MI larger than $O(\log N)$, where N is the number of data samples [10]. This exponential complexity also applies to specific estimators, for example, conventional k-nearest neighbor estimator [11, 12] and more recent variational estimators based on Donsker and Varadhan's representation of the KL divergence [13, 14].

In this paper, we experimentally compare the performance of various MI estimators, including variational estimators accompanying the formal limitations derived in [10]. The experiments also incorporate an alternative estimator based on a flow-based generative model [14, 15], which has the potential to complement the variational estimators when the true MI is large [10, 13, 14]. Collectively, we quantitatively evaluate the performance of the MI estimators to assess the applicability of recent advances in MI estimation.

While quantitative performance evaluations of MI estimators are present in the literature, they are often confined to the estimation of MI between Gaussian random vectors where the ground-truth MI is available in closed form [13, 14, 22]. This work extends the previous results to MI estimation between exponential random vectors that model power spectrum features in the frequency domain, where consecutive samples are likely to be asymptotically uncorrelated. Additionally, we include experiments in the high MI region, where the flow-based estimator shows a relative advantage over variational estimators.

To quantitatively evaluate the performance of the estimators against the ground-truth MI, we employ a simplified model of the power spectrum features with a tractable expression of the MI. Additionally, we explore more realistic alternative models of the power spectrum features by leveraging information-theoretic data processing inequality. Potential applications of the MI between power spectrum features can be found in the speech processing literature, e.g., [17–19].

2 MI estimators

2.1 Estimators based on variational bounds

Beginning with the classic Barber and Agakov lower bound [20] and following the explanation in [13], we formulate the variational lower bounds of Nguyen [21], van den Oord [9], Belghazi [22], and Song [14]. These formulations, which utilize critic function $f_\theta(x, y)$ parameterized by a neural network to approximate the ratio of the probability density functions $p(x|y)/p(x) = p(y|x)/p(y)$ [13, 14], serve as candidates for the experimental study in this paper.

First, the lower bound of Barber and Agakov is obtained by replacing the intractable conditional probability density function $p(x|y)$ with the variational distribution $q(x|y)$ as follows [13, 20],

$$\begin{aligned} I(X,Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(X|Y)}{p(X)}\right] \\ &\geq \mathbb{E}_{p(x,y)}\left[\log \frac{q(X|Y)}{p(X)}\right] \\ &= \mathbb{E}_{p(x,y)}[\log q(X|Y)] + h(X) = I_{\mathrm{BA}}(X,Y), \end{aligned} \tag{2}$$

where $h(X)$ denotes the differential entropy of the random variable X.

To circumvent the unknown differential entropy $h(X)$, the variational distribution $q(x|y)$ is defined using the critic function $f_\theta(x,y)$ as follows [13],

$$q(x|y) = \frac{p(x)}{g(y)} e^{f_\theta(x,y)}, \tag{3}$$

where the partition function $g(y)$ is given by

$$g(y) = \mathbb{E}_{p(x)}\left[e^{f_\theta(X,y)}\right]. \tag{4}$$

By substituting (3) into the last line of (2) and applying Jensen's inequality, the following lower bound of Donsker and Varadhan is obtained [13],

$$I_{\mathrm{DV}}(X,Y) = \mathbb{E}_{p(x,y)}[f_\theta(X,Y)] - \log \mathbb{E}_{p(x)p(y)}\left[e^{f_\theta(X,Y)}\right]. \tag{5}$$

Applying the inequality $\log(h) \leq h/e$ with $h = \mathbb{E}_{p(x)p(y)}\left[e^{f_\theta(X,Y)}\right]$ in (5), the NWJ estimate of Nguygen, Wainwright, and Jordan is derived as follows [13, 21],

$$I_{\mathrm{NWJ}}(X,Y) = \mathbb{E}_{p(x,y)}[f_\theta(X,Y)] - \mathbb{E}_{p(x)p(y)}\left[e^{f_\theta(X,Y)-1}\right]. \tag{6}$$

The NJW bound in (6) is tight with the optimal critic $f_{\mathrm{opt}}(x,y) = 1 + \log(p(x|y)/p(x))$ [13]. However, the variance of the NJW bound is large, due to the estimation of the partition function whose variance increases exponentially with respect to MI [13, 14].

The variance of the NJW bound can be reduced by using multiple independent samples. The Noise Contrastive Estimation (NCE) lower bound is obtained by averaging the bound over N replicates [13].

$$I_{\mathrm{NCE}}(X,Y) = \mathbb{E}_{p(x_{1:N},y_{1:N})}\left[\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{f_\theta(X_i,Y_i)}}{\frac{1}{N}\sum_{j=1}^{N} e^{f_\theta(X_i,Y_j)}}\right]. \tag{7}$$

However, unlike the NWJ bound, the NCE bound is loose since it is upper bounded by $\log(N)$ [13].

Another neural estimator, known as Mutual Information Neural Estimator (MINE), is also derived from the lower bound of Donsker and Varadhan in (5). The gradient of the Donsker and Varadhan lower bound is given by

$$\nabla_\theta I_{\mathrm{DV}}(X,Y) = \mathbb{E}_{p(x,y)}[\nabla_\theta f_\theta(X,Y)] - \frac{\mathbb{E}_{p(y)}[\nabla_\theta g(Y)]}{\mathbb{E}_{p(y)}[g(Y)]}. \tag{8}$$

where the expectations over a mini-batch lead to a biased gradient estimate [22]. The bias is reduced by the MINE gradient estimator, which replaces the estimate in the denominator of the second term in (8) by exponential average across mini-batches [22].

Since the MINE estimate also includes the partition function, the variance increases rapidly with respect to the MI [10, 14]. To alleviate this, the Smoothed Mutual Information Lower-bound Estimator (SMILE) is proposed by limiting the variance of the partition function using the clipping function $clip(a, \tau) = \max(\min(e^a, e^\tau), e^{-\tau})$ for some $\tau \geq 0$ [14],

$$I_{\text{SMILE}}(X, Y) = \mathbb{E}_{p(x,y)}[f_\theta(X, Y)] - \log \mathbb{E}_{p(x)p(y)}[clip(f_\theta(X, Y), \tau)]. \tag{9}$$

Specifically, I_{DV} and I_{NWJ} are appealing since they become tight with the optimal critic. However, they exhibit high variance due to their reliance on the high variance partition function estimator [13]. I_{NCE} and I_{SMILE} aim to reduce the variance at the cost of increasing bias. The hyperparameter τ of I_{SMILE} can be adjusted for the bias-variance trade-off [14], and for this work, we fixed $\tau = 1$.

In summary, the variational estimators aim to maximize the lower bound of mutual information with respect to the critic function implemented as a neural network. The loss function to minimize is defined as the negative lower bounds of (5), (6), (7) and (9), where the expectation is implemented by the sample mean under mini-batches. Gradient descents are employed to fit the parameters of the neural networks, and the estimates of the MI lower bound are smoothed across the mini-batches considering the small sample size of the mini-batch. The loss function during training the neural network is directly used for the estimator.

2.2 Estimators based on flow-based generative model

Given a set of data instances without labels, generative models capture the data distribution by fitting the parameters to maximize the data likelihood. Neural networks are used to fit the parameters, and the data log-likelihoods $\log p_\phi(x, y)$, $\log p_\psi(x)$ and $\log p_\xi(y)$ of samples from a mini-batch are evaluated to estimate MI as follows, [14]

$$I_{\text{GEN}}(X, Y) = \mathbb{E}_{p(x,y)}\left[\log p_\phi(X, Y)\right] - \mathbb{E}_{p(x)}\left[\log p_\psi(X)\right] - \mathbb{E}_{p(y)}\left[\log p_\xi(Y)\right] \tag{10}$$

where the expectations are implemented by sample means. Again, exponential smoothing of the MI estimates across the mini-batches is performed to obtain a more reliable estimator.

A restricted Boltzmann machine (RBM), described by a probabilistic undirected graph, is a maximum likelihood model that learns a probability distribution over its data [23]. Although greedy multi-layer training theoretically enhances the variational bounds of the log-likelihood, achieving the theoretical bounds may necessitate a prolonged Gibbs sampling, leading to an approximation such as contrastive divergence [23]. Additionally, the computation of the log-likelihood involves an intractable partition function [24].

In a directed graphical model, efficient approximate inference can be achieved by sharing variational parameters across data instances, a strategy called *amortized inference* [25]. In particular, variational autoencoder (VAE) leverages the efficiency of amortized inference while counteracting sample noise by the reparametrization trick. However, the

evaluating log-likelihood of a data point requires Monte-Carlo estimation using random samples of latent variables from the inference model [26].

Such approximations can be avoided by adopting a flow-based generative model instead, by leveraging the change of variable law of probabilities to transform a simple distribution into a complex one [27]. A flow-based model using real-valued non-volume preserving transformations (real NVP) leads to an unsupervised learning algorithm, that directly evaluates and minimizes the negative log-likelihood function [15]. This is advantageous for MI estimation using (10), which requires the computation of log-likelihood with respect to data samples in a mini-barch.

Given an observed data x, a latent variable z with simple prior distribution p_Z, and a bijection $f : X \to Z$, the log-likelihood is given by the change of variable formula [15],

$$\log\left(p_X(x)\right) = \log\left(p_Z(z)\right) + \log\left(\left|\det\left(\frac{\partial f}{\partial x^T}\right)\right|\right). \tag{11}$$

A flexible bijective model can be built by composing simple bijections, $f = f_1 \circ f_2 \circ \cdots \circ f_K$.

Let $x^T = [x_1^T, x_2^T]$ and $z^T = [z_1^T, z_2^T]$, with $x_1, x_2, z_1, z_2 \in \mathbb{R}^{n\times 1}$. Then, the real NVP transformations are composed of simple bijective functions, where each function is referred to as an affine coupling layer modeled as follows, [15]

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} z_1 \\ e^{s_{\phi_1}(z_1)} \odot z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ t_{\phi_2}(z_1) \end{bmatrix} \tag{12}$$

where $\odot$ is the Hadamard product. s_{ϕ_1} and t_{ϕ_2} stand for scale and translation, which can be arbitrarily complex by employing deep neural networks. The inverse of the bijective function is $x_1 = z_1, x_2 = e^{-s_{\phi_1}(z_1)} \odot \left(z_2 - t_{\phi_2}(z_1)\right)$, and the Jacobian is $\exp\left(\mathbf{1}^T s_{\phi_1}(z_1)\right)$, where $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is vector of ones. The inverse and Jacobian remain numerically efficient. Since each later keeps the first half of the vector z_1 unchanged, a permutation is performed after every layer [15].

While the variational approaches in Sect. 2.1 optimize single critic $f_\theta(x, y)$ which gives rise to a partition function, the flow-based generative model optimizes parameters of $\log p_\phi(x, y)$, $\log p_\psi(x)$, and $\log p_\xi(y)$ separately. In particular, the flow-based estimator requires samples from the joint data distribution $p(x, y)$ only, while variational estimators further require samples from the product of marginals $p(x)p(y)$, except I_{NCE}, to evaluate the second terms on the right-hand sides of (5), (6) and (9) [14].

3 Experiments

In this section, we present experimental results quantifying the degree of the statistical relationship between a pair of random signal vectors by MI estimators. Due to the equivalence of time and frequency domain representations of a signal, MI can be equivalently estimated with spectral features.

The advantage of using the spectral features is that the spectral coefficients tend to be mutually uncorrelated with an increasing analysis window length for a stationary signal. Additionally, they tend to be asymptotically complex Gaussian due to the central limit theorem [28]. Thus, utilizing spectral features alleviates the burden of capturing complicated inter-dependency within the signal vector.

Based on asymptotic theory, the signal vectors are assumed to consist of elements that are exponentially distributed and element-wise correlated. Then, the joint distribution between the signal vectors reduces to the product of bivariate distributions,

$$p(x,y) = \prod_{i=1}^{D} p(x_i, y_i), \;\; x_i \geq 0, \; y_i \geq 0 \tag{13}$$

with

$$p(x_i, y_i) = \frac{1}{4\sigma^4(1-r)} \exp\left(-\frac{x_i + y_i}{2\sigma^2(1-r)}\right) I_0\left(\frac{\sqrt{r x_i y_i}}{\sigma^2(1-r)}\right) \tag{14}$$

in which I_0 denotes the modified Bessel function of the first kind, r denotes the correlation coefficient between the pair of the exponential random variables (X_i, Y_i) and $2\sigma^2$ is equal to the mean of the marginal exponential distribution [29].

The bivariate exponential distribution (14) is derived from transforming a pair of two-dimensional zero-mean Gaussian random vectors, $[Z_{i_1}, Z_{i_2}]^T$ and $[Z_{i_3}, Z_{i_4}]^T$, with element-wise correlation coefficient of ρ and zero correlation elsewhere (see Eq. (119) in [29]). Through the transformations of $X_i = Z_{i_1}^2 + Z_{i_2}^2$ and $Y_i = Z_{i_3}^2 + Z_{i_4}^2$, the correlation coefficient r is evaluated to be $r = \rho^2$ [29].

We calculate the *true* MI by two-dimensional numerical integration, substituting (14) into (1). Figure 1 shows the resulting MI between bivariate exponential variables versus correlation coefficient r, compared with MI between bivariate Gaussian variables versus correlation coefficient ρ, evaluated from the closed form expression of $-0.5\log(1-\rho^2)$ [16].

The element-wise bivariate exponential model may be considered an oversimplification compared to more realistic models with an underlying super-Gaussian distribution [30–32]. However, drawing samples from bivariate Gamma or Laplacian distribution based on the underlying super-Gaussian model in [30–32], given MI, is more complicated to implement. Instead, a more practical approach is to leverage the data processing inequality [1],

$$I(X,Y) = I(g_1(X), g_2(Y)), \;\; \text{for bijective } g_1 \text{ and } g_2, \tag{15}$$

where, the transformed data $g_1(X)$ and $g_2(Y)$ create an alternative model of the power spectrum features. For instance, a simple element-wise transformation of the

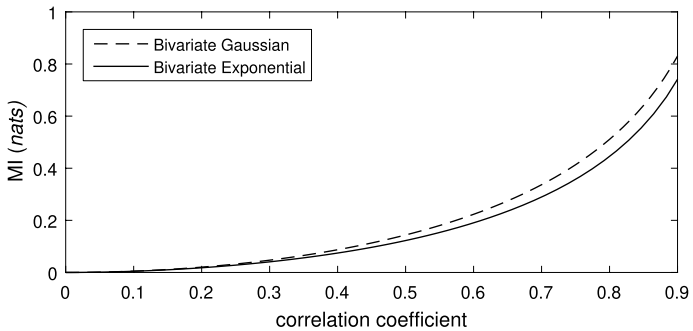

Fig. 1 Mutual information versus correlation coefficient

exponential model, i.e., $g(X) = X^{\nu}$ with $\nu > 1$, results in a Weibull distribution derived from an underlying super-Gaussian model [33].

3.1 Tasks

In the literature, the signal vectors are assumed to be drawn from a 20-dimensional Gaussian distribution with element-wise correlation [13, 14, 22]. To assess performance with higher-dimensional data, we experimented with a signal dimension of 40 and compare the result with the signal dimension of 20. Unlike previous works, each pair of elements is sampled from the bivariate exponential distribution of (13).

The true MI between the elements is varied from 0.2 to 1 *nats*, in steps of 0.2 *nats*, with $5k$ iterations conducted in each step. Pairs of correlated complex Gaussian random vectors are drawn, with a correlation coefficient ρ corresponding to the true MI. The vectors obtained by squaring the modulus of the complex Gaussian samples are fed into the neural networks. Then, the bias and variance of the estimators are analyzed and compared with each other.

Furthermore, to investigate the validity of the estimators upon data processing, we apply transformations, such as taking the element-wise logarithm, squaring, scaling, and shift rotation, which precede the input layer. The input data (x, y) is replaced with $(x, g(y))$, where g denotes the applied transformation. Taking logarithms results in commonly used log-spectra features. Squaring results in a heavy-tailed distribution which is based on a super-Gaussian model of power spectrum. Scaling implements a frequency-dependent gain of a system. For the scaling, we multiply the data vector by a diagonal matrix, whose condition number is equal to $1e + 3$. Shift rotation is a frequency shifting operation, which is related to Doppler or harmonics. For the shift rotation, we circularly rotate the data by 10 bins.

3.2 Neural network architectures

For variational methods, the critic function $f_\theta(x, y)$ is parameterized by a neural network. Assuming a separable critic $f_\theta(x, y) = h_{\theta_1}(x)^T g_{\theta_2}(y)$ where h and g are learned by two separate networks, only $2N$ forward passes are required for a batch size of N [13]. In the joint critic, x and y are concatenated and fed into the neural network. For an equal batch size of N, the combination of the input vectors leads to N^2 forward passes for the joint critic. In general, the joint critics tend to perform better than separable critics [13]. In this work, we consider the joint critic architecture only.

For all neural networks, we follow the setup in [13, 14]. We use the Adam optimizer with a learning rate of 5×10^{-4} [34], and a batch size of 128. Rectified linear unit (ReLU) activations are used for each neuron. For the variational estimators, we use two-layer perceptrons with 256 neurons per layer in the case of 20-dimensional inputs. For the flow-based generative method, each of the log-likelihoods is estimated from 6 coupling layers, with each coupling layer implemented by two-layer perceptrons with 100 neurons per layer in the case of 20-dimensional inputs [15]. The usual Gaussian priors are used for the flow-based method. For 40-dimensional inputs, we experimented with various network parameters to examine the reliability of the architecture when the dimension increases.

3.3 Experimental results

In Fig. 2, MI estimators between 20-dimensional vectors are presented, with the true MI stepping up every $5k$ iterations. The variational MI estimators exhibit large estimation errors at high MI due to the high variance of the partition function estimators or bias [10, 13, 14]. In particular, I_{NWJ} and I_{MINE} show high variance, while I_{NCE} shows high bias due to the upper bound introduced by (7). I_{SMILE} benefits from the bias-variance trade-off but also degrades at high MI.

On the other hand, the performance of the flow-based generative estimator, I_{FLOW}, is maintained even at high MI. In the experiment, the marginal distributions obtained from (14) are exponential with mean $2\sigma^2$ irrelevant to the MI, whereas the joint distribution changes according to the MI. Empirically, the sample variance of the joint entropy estimator, $-\mathbb{E}_{p(x,y)}\left[\log p_\phi(X,Y)\right]$ in (10), is not significantly influenced by increasing MI. However, the flow-based estimator is generally more biased than the variational estimators at low MI.

Figure 3 shows the MI estimators obtained by doubling the dimension of the vectors to $D = 40$. The results of Fig. 2 for $D = 20$ are duplicated in the leftmost column of Fig. 3 for comparison. The combined estimator in the last row, a heuristic algorithm combining the variational and flow-based estimator, will be explained later. The four columns, except the leftmost one, depict the MI estimators for $D = 40$. Each column is obtained by changing neural network parameters, where the number of neurons per layer and the batch size are either unchanged or doubled in line with the increased dimension.

The variational MI estimators for $D = 40$ in Fig. 3 exhibit a more severe deviation from the true value than $D = 20$, regardless of the changes in the network parameters. Particularly in the case 2 of Fig. 3, I_{NWJ} tends to negative values at high MI. In general, the architecture of the variational estimators is difficult to scale up with the data dimension. Under the current architecture, it is conceivably necessary to divide frequency bins into disjoint subbands and estimate MI separately, resorting to asymptotic statistical independence between the subbands assuming a stationary signal.

In contrast, the architecture for the flow-based estimator appears to be more reliable for $D = 40$. At high MI, the performance of the estimator for $D = 40$ is comparable to $D = 20$ when both the number of neurons per layer and the batch size increase twofold. However, at the intermediate MI region, the convergence of the estimator tends to be more unstable when the dimension increases to $D = 40$.

Figure 4 shows MI estimators between 20-dimensional input vectors transformed by element-wise logarithm, squaring, scaling, and shift rotation. Overall, the performance of I_{SMILE} at low MI and I_{FLOW} at high MI appears to be more robust against

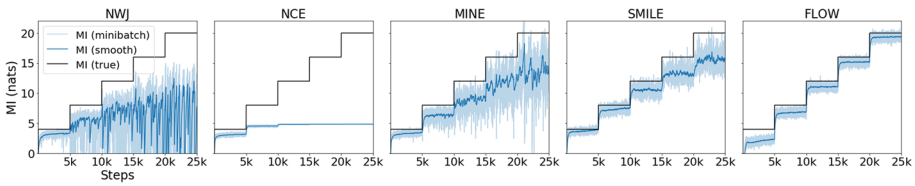

Fig. 2 MI estimators between 20-dimensional input vectors. (Light color: estimates in each iteration using a single mini-batch, dark color: estimates exponentially moving averaged across mini-batches, black color: true MI)

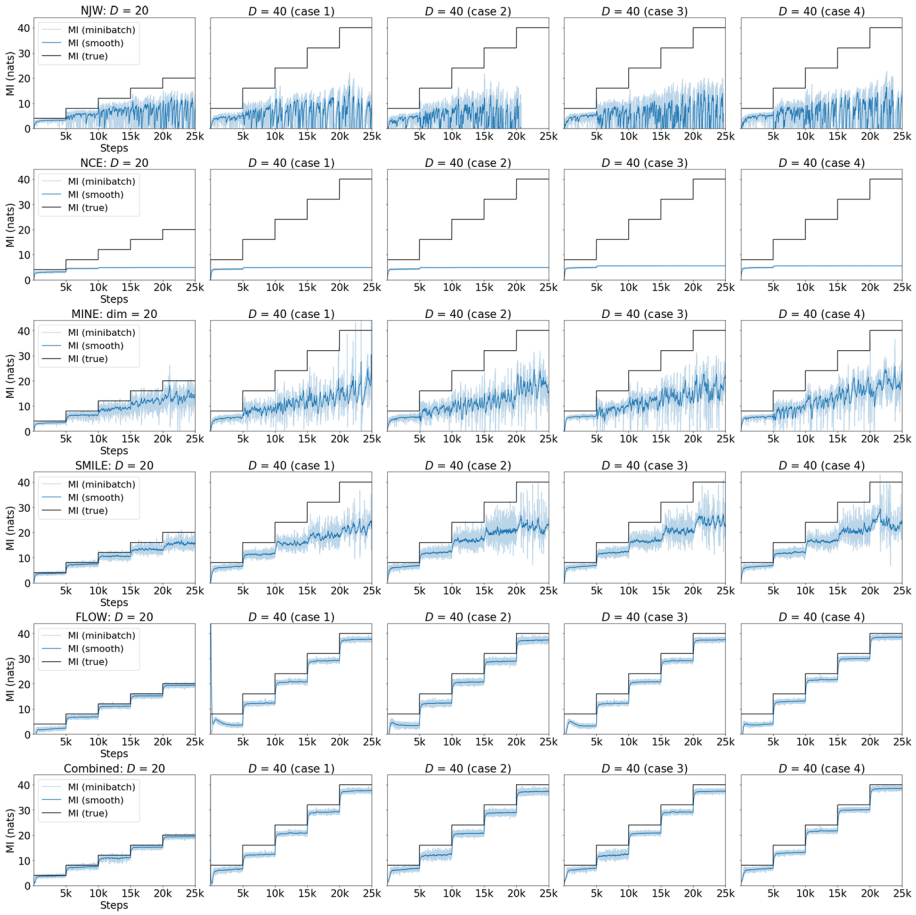

Fig. 3 MI estimators obtained by increasing the dimension D of input vectors. (Light color: estimates in each iteration from a mini-batch, dark color: estimates exponentially moving averaged across mini-batches, black color: true MI) Left column $D = 20$. Second column (case 1) $D = 40$, with the architecture of the network unchanged. Third column (case 2) $D = 40$, with the number of neurons per layer doubled. Fourth column (case 3) $D = 40$, with the batch size doubled. Last column (case 4) $D = 40$, with both the number of neurons per layer and the batch size doubled

transformations, exhibiting smaller performance degradations relative to other estimators. Except for I_{NCE}, which exhibits high bias, the bias or variance of the estimators is more negatively affected by squaring or scaling transformations than by taking the logarithm or shift rotation. Expanding the range of input data by squaring or scaling seems to have a detrimental effect on the estimators.

Since bijective transformations preserve MI, shrinking the range of the input data by applying a bijective transformation may help improve the estimator. For example, we can apply a linear transformation normalizing each frequency bin to unit variance [35], which is the inverse transformation of the scaling transformation, thus mitigating the bias induced by scaling shown in the fourth column of Fig. 4. Similarly, the heavy-tailed spectrum shown in the third column of Fig. 4 is inverse transformed by the square root function, resulting in a nonlinear transformation for input regularization. However, for this heavy-tailed white spectrum, a linear transformation such as

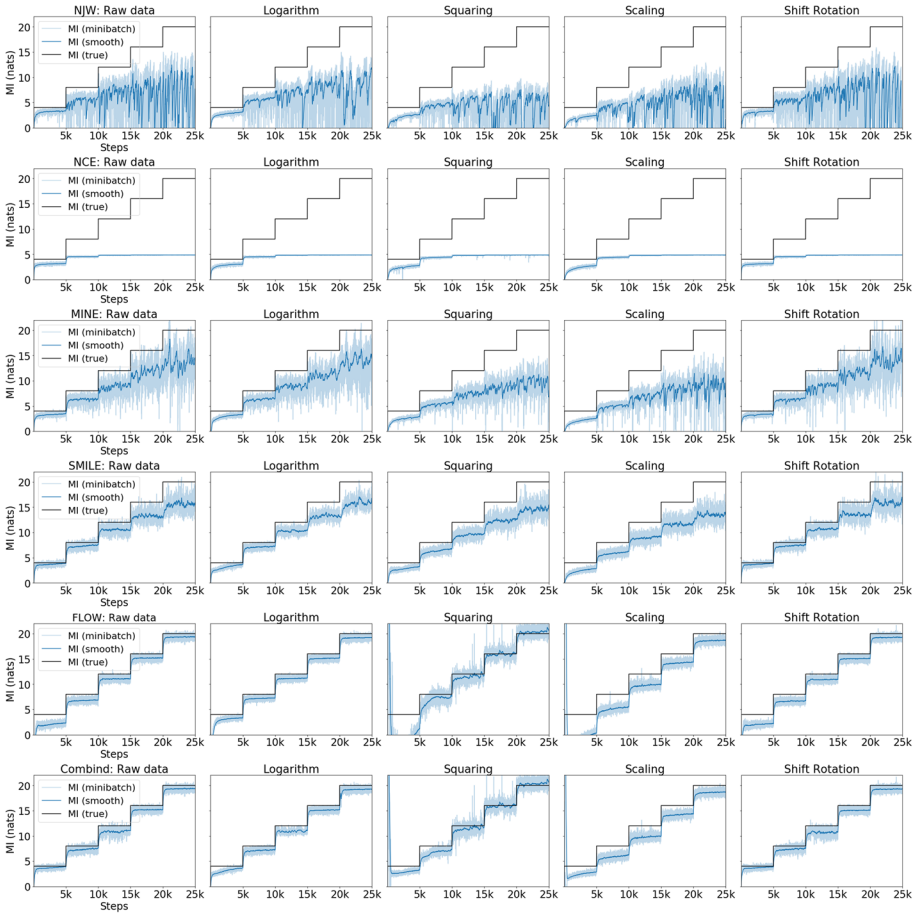

Fig. 4 MI estimators between 20-dimensional input vectors transformed by element-wise logarithm, squaring, scaling, and shift rotation. (Light color: estimates in each iteration from a mini-batch, dark color: estimates exponentially moving averaged across mini-batches, black color: true MI)

normalizing to unit variance is not successful in shrinking the data range and is thus not adequate for input regularization.

3.3.1 A heuristically combined estimator

In practice, it is worthwhile to evaluate both I_{SMILE} and I_{FLOW} and choose one by inspecting the sample mean and variance of the estimators. When the sample mean of I_{FLOW} is smaller than that of I_{SMILE}, we expect more bias from I_{FLOW}, as the variational I_{SMILE} tends to exhibit negative bias resulting from the estimates of lower bounds. On the contrary, a large sample variance of I_{SMILE} indicates that I_{SMILE} is not a reliable estimator. We observe that the sample variance of I_{FLOW} is relatively insensitive the true MI and thus is not a useful indicator of the confidence of I_{FLOW} as shown in Fig. 3.

Consequently, in cases where the sample variance of I_{SMILE} is small while the sample mean of I_{FLOW} is relatively smaller than I_{SMILE}, we presume that I_{SMILE} is more reliable. On the other hand, when the sample mean of I_{FLOW} is larger than I_{SMILE} while the sample variance of I_{SMILE} is large, we choose I_{FLOW} as a better alternative. Otherwise, when it is not evident that one estimator is preferable to the other and in case we lack further evidence, a simple average of the estimators may be used.

Algorithm 1 Heuristic algorithm for combining MI estimators

Inputs : MI estimates from M mini-batches
$\hat{I}_{\mathrm{SMILE}}(t), t = 1, 2, ...M$
$\hat{I}_{\mathrm{FLOW}}(t), t = 1, 2, ...M$
Initialize :
$\bar{I}_{\mathrm{SMILE}}(1) = \hat{I}_{\mathrm{SMILE}}(1)$, $\overline{\sigma^2}_{\mathrm{SMILE}}(1) = 0$
$\bar{I}_{\mathrm{FLOW}}(1) = \hat{I}_{\mathrm{FLOW}}(1)$
for $t = 2$ to M **do**
  $\bar{I}_{\mathrm{SMILE}}(t) = \alpha \bar{I}_{\mathrm{SMILE}}(t-1) + (1-\alpha)\hat{I}_{\mathrm{SMILE}}(t)$
  $\bar{I}_{\mathrm{FLOW}}(t) = \alpha \bar{I}_{\mathrm{FLOW}}(t-1) + (1-\alpha)\hat{I}_{\mathrm{FLOW}}(t)$
  $\overline{\sigma^2}_{\mathrm{SMILE}}(t) = \alpha(\overline{\sigma^2}_{\mathrm{SMILE}}(t-1) + (1-\alpha)(\hat{I}_{\mathrm{SMILE}}(t) - \bar{I}_{\mathrm{SMILE}}(t-1))^2)$
  if $\bar{I}_{\mathrm{FLOW}}(t) - \bar{I}_{\mathrm{SMILE}}(t) < T_1$ and $\overline{\sigma^2}_{\mathrm{SMILE}}(t) < {T_2}^2$ **then**
    $\hat{I}_{\mathrm{Comb}}(t) = \hat{I}_{\mathrm{SMILE}}(t)$
  else if $\bar{I}_{\mathrm{FLOW}}(t) - \bar{I}_{\mathrm{SMILE}}(t) \geq T_1$ and $\overline{\sigma^2}_{\mathrm{SMILE}}(t) > {T_3}^2$ **then**
    $\hat{I}_{\mathrm{Comb}}(t) = \hat{I}_{\mathrm{FLOW}}(t)$
  else
    $\hat{I}_{\mathrm{Comb}}(t) = (\hat{I}_{\mathrm{SMILE}}(t) + \hat{I}_{\mathrm{FLOW}}(t))/2$
  end if
end for
Output : $\hat{I}_{\mathrm{Comb}}(t), t = 2, ...M$

The proposed heuristic algorithm is detailed in the Algorithm 1, combining the two estimators, I_{SMILE} and I_{FLOW}, based on the sample mean and variance of the estimators. This algorithm employs a recursive form of the weighted incremental update formula from [36]. The constants used in the algorithm are $\alpha = 0.99$, $T_1 = 0.5$, $T_2 = 1$, and $T_3 = 0.5$ *nats*. The MI estimates obtained using the heuristic algorithm are shown in the last row of Figs. 3 and 4.

The algorithm successfully combines I_{SMILE} at low MI and I_{FLOW} at high MI. In the intermediate MI region, the algorithm switches between the individual estimators and the simple average of the estimators. The behavior is affected by the threshold parameters T_1, T_2, and T_3 used. In general, higher T_1 and T_2 give more confidence to I_{SMILE} and lower T_3 gives more confidence to I_{FLOW}. Simultaneously lowering T_2 and increasing T_3 in the opposite direction results in less confidence in the individual estimators and thus leads to simple averaging. The adjustment of the parameters is ineffective when both individual estimators are biased. As shown in Fig. 3, the combined estimator still suffers from bias when both individual estimators are biased in the intermediate MI region, especially for high-dimensional inputs($D = 40$). For $D = 40$, doubling both the number of neurons and the batch size reduces the bias of I_{FLOW} and of the resulting combined estimator.

Tables 1 and 2 quantify the bias and standard deviation estimated from the last $1k$ iterations at each step shown in Fig. 4. Again, except for I_{NCE}, the squaring and scaling transformations result in increased bias of the variational estimators and a significant increase in the bias of I_{FLOW} at low MI. The standard deviations of the estimates

Table 1 Bias estimates (in *nats*) with respect to data processing

Estimators	*true* MI	Raw	Logarithm	Squaring	Scaling	Shift
NWJ	4	− 0.72	− 0.93	− 1.41	− 1.60	− 0.72
	12	− 5.39	− 6.69	− 8.16	− 7.17	− 4.75
	20	− 15.39	− 10.79	− 15.00	− 22.29	− 22.89
NCE	4	− 0.84	− 0.97	− 1.26	− 1.29	− 0.86
	12	− 7.19	− 7.19	− 7.21	− 7.21	− 7.19
	20	− 15.15	− 15.15	− 15.15	− 15.15	− 15.15
MINE	4	− 0.57	− 0.73	− 1.20	− 1.50	− 0.65
	12	− 2.92	− 3.20	− 4.55	− 4.64	− 3.11
	20	− 6.12	− 5.81	− 10.67	− 11.28	− 5.72
SMILE	4	**− 0.14**	**− 0.39**	**− 0.83**	**− 1.19**	**− 0.14**
	12	− 1.51	− 1.67	− 2.34	− 2.85	− 1.32
	20	− 4.39	− 3.99	− 5.29	− 6.55	− 4.23
FLOW	4	− 1.72	− 0.70	− 3.40	− 3.83	− 1.83
	12	**− 0.92**	**− 0.81**	**− 0.26**	**− 2.07**	**− 1.06**
	20	**− 0.63**	**− 0.76**	**0.61**	**− 1.32**	**− 0.71**
Combined	4	**− 0.14**	**− 0.39**	**− 0.83**	**− 1.19**	**− 0.14**
	12	− 0.94	− 1.10	− 0.28	**− 2.07**	− 1.32
	20	**− 0.63**	**− 0.76**	**0.61**	**− 1.32**	**− 0.71**

Best results are indicated in bold

Table 2 Standard deviation estimates (in *nats*) with respect to data processing

Estimators	*true* MI	Raw	Logarithm	Squaring	Scaling	Shift
NWJ	4	0.34	0.23	0.24	0.21	**0.27**
	12	4.93	21.60	7.20	3.53	3.21
	20	17.86	8.06	5.26	29.52	29.68
NCE	4	*0.16*	*0.15*	*0.16*	*0.16*	*0.15*
	12	*0.03*	*0.03*	*0.04*	*0.04*	*0.03*
	20	*0.01*	*0.01*	*0.01*	*0.01*	*0.01*
MINE	4	**0.25**	0.24	**0.22**	**0.20**	0.31
	12	1.08	1.02	0.96	1.07	1.34
	20	2.44	2.30	2.45	2.72	2.32
SMILE	4	0.29	**0.23**	0.27	0.26	0.28
	12	0.71	0.49	**0.63**	0.71	0.64
	20	1.34	1.04	1.34	1.13	1.47
FLOW	4	0.33	0.24	0.79	0.43	0.32
	12	**0.38**	**0.32**	0.85	**0.44**	**0.37**
	20	**0.38**	**0.36**	**0.85**	**0.44**	**0.40**
Combined	4	0.29	**0.23**	0.27	0.26	0.28
	12	0.45	0.37	0.84	**0.44**	0.64
	20	**0.38**	**0.36**	**0.85**	**0.44**	**0.40**

Best results are indicated in bold, except for the standard deviation of the NCE estimator (indicated in italics), which restricts the deviation at the cost of upper bounding the estimator

are not significantly affected by the transformations. The heuristically combined estimator, taking advantage of the individual estimators, exhibits relatively smaller bias.

The estimated standard deviation of the combined estimator is also relatively small, resulting from the sample variance of individual estimators.

4 Conclusion

In this study, we evaluated several neural estimators of mutual information (MI) between random vectors, which model power spectrum features. Firstly, the variational estimators exhibited significant bias or variance in the estimated value at high MI, leading to a substantial estimation error. However, these estimators demonstrated relatively greater reliability at low MI. Conversely, the flow-based generative estimator showed reduced bias and variance at high MI but suffered from slow convergence at low MI. These empirical results indicate that each MI estimator possesses inherent limitations, emphasizing the necessity of a complementary use of these estimators to mitigate their respective drawbacks. In response to this, we propose a heuristic algorithm that combines the strengths of individual estimators.

However, the selection of parameters in the algorithm appears somewhat arbitrary and should be tailored based on the specific characteristics of the utilized data. Additionally, the algorithm still suffers from bias when both the estimators combined are biased. In particular, variational estimators tend to display more negative bias at high MI, underscoring bias a significant concern. Moreover, although the generative flow-based estimator demonstrates relatively strong performance at high MI, there is a need for a theoretical explanation of its statistical characteristics to justify its application at high MI. Beyond this, it is necessary to explore alternative generative models for MI estimation, aside from the flow-based estimator discussed in this study. Lastly, to validate the practical applicability of the neural estimators, future work should include experiments with real-world data.

Acknowledgements
Authors are indebted to Dr. J. Song and Dr. S. Ermon for the implementation of the variational estimators (github.com/ermongroup/smile-mi-estimator) and to Dr. A. S. Ashukha for the implementation of the flow-based estimator (github.com/senya-ashukha/real-nvp-pytorch). Also, authors are grateful to the editor and anonymous reviewers for their constructive comments.

Author contributions
HSK motivated the work, interpreted the experimental results and reviewed the manuscript. DS performed the experiments and analysis. DS drafted the manuscript. All the authors read and approved the final manuscript.

Funding
This work was supported by a 2-Year Research Grant of Pusan National University.

Availability of data and materials
Python scripts used for generating synthetic data are available in additional supporting files.

Declaration

Competing interests
The authors declare that they have no competing interests.

Received: 20 December 2022 Accepted: 7 December 2023
Published online: 02 January 2024

References

1. T.M. Cover, J.A. Thomas, *Elements of Information Theory* (Wiley, New Jersey, 1991)
2. R. Linsker, Self-organization in a perceptual network. Computer **21**(3), 105–117 (1988). https://doi.org/10.1109/2.36
3. R. Linsker, Local synaptic learning rules suffice to maximize mutual information in a linear network. Neural Comput. **4**(5), 691–702 (1992). https://doi.org/10.1162/neco.1992.4.5.691

4. S. Becker, Mutual information maximization: models of cortical self-organization. Netw. Comput. Neural Syst. **7**(1), 7 (1996). https://doi.org/10.1088/0954-898X/7/1/003
5. A.J. Bell, T.J. Sejknowksi, An information-maximization approach to blind separation and blind deconvolution. Neural Comput. **7**(6), 1129–1159 (1995). https://doi.org/10.1162/neco.1995.7.6.1129
6. L. Bahl, P. Brown, P. De Souza, R. Mercer, Maximum mutual information estimation of hidden Markov model parameters for speech recognition, in *Proc. IEEE ICASSP'86*, pp. 49–52 (1986). https://doi.org/10.1109/ICASSP.1986.1169179
7. R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, Y. Bengio, Learning deep representations by mutual information estimation and maximization, in *International Conference on Learning Representations* (2018)
8. M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, M. Lucic, On Mutual Information Maximization for Representation Learning, in *International Conference on Learning Representations* (2019)
9. A. Van Den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding (2018), arXiv:1807.03748
10. D. McAllester, K. Stratos, Formal limitations on the measurement of mutual information, in *International Conference on Artificial Intelligence and Statistics, PMLR*, pp. 875–884 (2020)
11. S. Gao, G. Ver Steeg, A. Galstyan, Efficient estimation of mutual information for strongly dependent variables, in *International Conference on Artificial Intelligence and Statistics, PMLR*, pp. 277–286 (2015)
12. A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information. Phys. Rev. E **69**(6), 066138 (2004). https://doi.org/10.1103/PhysRevE.69.066138
13. B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, G. Tucker, On variational bounds of mutual information, in *International Conference on Machine Learning, PMLR*, p 5171–5180 (2019)
14. J. Song, S. Ermon, Understanding the Limitations of Variational Mutual Information Estimators, in *International Conference on Learning Representations* (2019)
15. L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, in *International Conference on Learning Representations* (2017)
16. I.M. Gelfand, Calculation of the amount of information about a random function contained in another such function. Ann. Math. Soc. Transl. Ser. **2**(12), 199–246 (1959)
17. J. Taghia, R. Martin, Objective intelligibility measures based on mutual information for speech subjected to speech enhancement processing. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(1), 6–16 (2014). https://doi.org/10.1109/TASL.2013.2281574
18. J. Jensen, C.H. Taal, Speech intelligibility prediction based on mutual information. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(2), 430–440 (2014). https://doi.org/10.1109/TASLP.2013.2295914
19. S. Khademi, R.C. Hendriks, W.B. Kleijn, Intelligibility Enhancement Based on Mutual Information. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(8), 1694–1708 (2017). https://doi.org/10.1109/TASLP.2017.2714424
20. D. Barber, F. Agakov, The IM algorithm: a variational approach to information maximization. Adv. Neural. Inf. Process. Syst. **16**(320), 201 (2004)
21. X. Nguyen, M.J. Wainwright, M.I. Jordan, Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Trans. Inf. Theory **56**(11), 5847–5861 (2010). https://doi.org/10.1109/TIT.2010.2068870
22. M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, D. Hjelm Mutual Information Neural Estimation, in *International Conference on Machine Learning, PMLR*, pp. 531–540 (2018)
23. G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006). https://doi.org/10.1162/neco.2006.18.7.1527
24. R. Salakhutdinov, I. Murray, On the quantitative analysis of deep belief networks, in *International Conference on Machine Learning. PMLR*, pp. 872–879 (2008). https://doi.org/10.1145/1390156.1390266
25. S. Gershman, N. Goodman, Amortized inference in probabilistic reasoning, in *Proceedings of the Annual Meeting of the Cognitive Science Society* (2014)
26. D.P. Kingma, M. Welling, An introduction to variational autoencoders (2019), arXiv:1906.02691
27. D. Rezende, S. Mohamed, Variational inference with normalizing flows, in *International Conference on Machine Learning. PMLR*, pp. 1530–1538 (2015)
28. J. Schoukens, J. Renneboog, Modeling the noise influence on the Fourier coefficients after a discrete Fourier transform. IEEE Trans. Instrum. Meas. **IM–35**(3), 278–286 (1986). https://doi.org/10.1109/TIM.1986.6499210
29. R.K. Mallik, On multivariate Rayleigh and exponential distributions. IEEE Trans. Inf. Theory **49**(6), 1499–1515 (2003). https://doi.org/10.1109/TIT.2003.811910
30. T. Lotter, P. Vary, Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model. EURASIP J. Adv. Signal Process. **2005**(7), 1–17 (2005). https://doi.org/10.1155/ASP.2005.1110
31. R. Martin, Speech enhancement based on minimum mean-square error estimation and super Gaussian priors. IEEE Trans. Speech Audio Process. **13**(5), 845–856 (2005). https://doi.org/10.1109/TSA.2005.851927
32. I. Cohen, Speech enhancement using super-Gaussian speech models and noncausal a priori SNR estimation. Speech Commun. **47**(3), 336–350 (2005). https://doi.org/10.1016/j.specom.2005.02.011
33. N.C. Sagias, G.K. Karagiannidis, Gaussian class multivariate Weibull distributions: theory and applications in fading channels. IEEE Trans. Inf. Theory **51**(10), 3608–3619 (2005). https://doi.org/10.1109/TIT.2005.855598
34. D. P. Kingma, J. Ba, Adam: a method for stochastic optimization (2014), arXiv:1412.6980
35. C.M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, London, 1995), pp.298–300
36. D.H.D. West, Updating mean and variance estimates: an improved method. Commun. ACM **22**(9), 532–535 (1979). https://doi.org/10.1145/359146.359153

Publisher's Note