Information Theory in Density Destructors

J. Emmanuel Johnson [1] **Valero Laparra** [1] **Raul Santos-Rodriguez** [2] **Gustau Camps-Valls** [1] **Jesus Malo** [1]

Abstract

Density destructors are differentiable and invertible transforms that map multivariate PDFs of arbitrary structure (low entropy) into non-structured PDFs (maximum entropy). Multivariate Gaussianization and multivariate equalization are specific examples of this family, which break down the complexity of the original PDF through a set of elementary transforms that progressively remove the structure of the data.

We demonstrate how this property of density destructive flows is connected to classical information theory, and how density destructors can be used to get more accurate estimates of information theoretic quantities. Experiments with total correlation and mutual information in multivariate sets illustrate the ability of density destructors compared to competing methods. These results suggest that information theoretic measures may be an alternative optimization criteria when learning density destructive flows.

1. Introduction

Estimating the probability density function (PDF) plays a central role in many machine learning problems like regression, classification, or data representation. However, the problem of PDF estimation is notoriously difficult when considering moderate and high dimensional data. In the deep learning community three families of methods are responsible for the majority of the progress in PDF estimation: Variational AutoEncoders (VAEs) (Kingma & Welling, 2014), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Invertible Flows (IFs) (Rezende & Mohamed, 2015). Each family tackles the PDF estimation from a slightly different *algorithmic* perspective, but they share many *conceptual* properties. They look for two main components: the first looks for a function $\mathbf{G}_\theta(\cdot)$ that maps samples from a known latent space $\mathcal{Z}$ to the observed space $\mathcal{X}$. The second component aims to find a function $\mathbf{D}_\theta(\cdot)$ that maps data from our observed space $\mathcal{X}$ to some latent space $\mathcal{Z}$. Both individual components can be seen, and will be hereafter referred to, as density *generators* and density *destructors* (Inouye & Ravikumar, 2018). The **generative** transformation can be written as:

$$\mathbf{z} \xrightarrow{\mathbf{G}_\theta} \hat{\mathbf{x}} \tag{1}$$

where $\mathbf{z}$ comes from our latent space distribution $\mathcal{P}_\mathbf{z}$, θ are the parameters of the generative transformation $\mathbf{G}$, and $\hat{\mathbf{x}}$ is the approximated data that follows the distribution $\hat{\mathcal{P}}_\mathbf{x}$. This component is found in all the frameworks mentioned above: the generator in GANs, the decoder portion of VAEs, and the invertible function $\mathbf{f}(\cdot)$ in IFs. Obtaining this component is difficult as the hypothesis space for $\mathbf{G}_\theta$ is large as we do not know the actual PDF of the data, $\mathcal{P}_\mathbf{x}$. Thus it is difficult to create appropriate cost functions and clever learning schemes are required to obtain $\mathbf{G}$; i.e. the adversarial formulation in GANs, the encoder-decoder relationship in VAEs, or imposing the invertibility of $\mathbf{f}$ in IFs.

Alternatively, one could look at the problem in the reverse order as a **destructive** transformation:

$$\mathbf{x} \xrightarrow{\mathbf{D}_\theta} \hat{\mathbf{z}} \tag{2}$$

where $\mathbf{x}$ comes from the true data distribution $\mathcal{P}_\mathbf{x}$, θ are the parameters of the destructive transformation $\mathbf{D}$, and $\hat{\mathbf{z}}$ follows the approximated base density $\hat{\mathcal{P}}_\mathbf{z}$. This term does not exist in the classical GAN formulation but several new versions have tried to overcome it (Chen et al., 2016; Makhzani et al., 2016; Zhu et al., 2017). In VAEs, this destructor is a non-invertible function (the encoder) where we need to pair it with the decoder $\mathbf{G}$ for learning. In IFs several methods attempt to learn an invertible function $\mathbf{f}$ through inference, mapping the data from $\mathcal{X}$ to a latent space $\mathcal{Z}$, (Dinh et al., 2017; Laparra et al., 2011; Ballé et al., 2016). Given an invertible transform, $\mathbf{D}$, the relation between our data distribution $\mathcal{P}_\mathbf{x}$ and the $\mathcal{P}_\mathbf{z}$ can be calculated through the standard change of variables used in most IFs (Rezende & Mohamed, 2015):

$$\mathcal{P}_\mathbf{x}(\mathbf{x}) = \mathcal{P}_\mathbf{z}(\hat{\mathbf{z}}) \left|\nabla_\mathbf{x} \mathbf{D}(\mathbf{x})\right| \tag{3}$$

[0]This work was funded by MINECO: DPI2017-89867.

[1]Universitat de Valencia, Valencia, Spain [2]Bristol, United Kingdom. Correspondence to: J. Emmanuel Johnson <juan.johnson@uv.es>.

First workshop on *Invertible Neural Networks and Normalizing Flows* (ICML 2019), Long Beach, CA, USA

where $\hat{\mathbf{z}} = \mathbf{D}(\mathbf{x})$, $\mathcal{P}_{\mathbf{z}}$ is the base distribution, and $|\nabla_{\mathbf{x}}(\cdot)|$ is the determinant of the Jacobian of our density destructor $\mathbf{D}$.

The *destructive* perspective gives us some advantages, as we can define a latent distribution $\mathcal{P}_{\mathbf{z}}$ with nice enough properties that we can measure how well we approach it. For example, assuming the base density is uniform, $\mathcal{P}_{\mathbf{z}} \sim \mathcal{U}$, the change-of-variables formula (Eq. 3) just results in the calculation of the exact likelihood of the data because $\mathcal{P}_{\mathbf{z}}(\hat{\mathbf{z}})$ is equal to one (Inouye & Ravikumar, 2018). Alternatively, we can assume that the base PDF is Gaussian and use standard non-Gaussianity measures to assess the distance to the goal (Laparra et al., 2011; Ballé et al., 2016). In other cases, sensible cost functions such as the Kullback-Leibler Divergence ($\mathrm{D_{KL}}$) could be used to measure the similarity between the approximated $\hat{\mathcal{P}}_{\hat{\mathbf{z}}}$ and the true $\mathcal{P}_z$ we choose.

2. Proposal

The literature on Invertible Flows (Rezende & Mohamed, 2015; Inouye & Ravikumar, 2018) does not link these transforms with classical information theory. In this work we establish this connection by using two properties of density destructive flows: 1) destructors effectively reduce the data structure so that the output may have trivial entropy/redundancy, and 2) destructors are smooth routes to the target PDF, so their Jacobian can always be computed, which allows us to obtain information measures that ultimately depend on $\nabla_{\mathbf{x}}\mathbf{D}$. Quantifying data structure and the relations between features is at the core of machine learning. Information theoretic magnitudes describe data complexity with few or no assumptions (Timme & Lapish, 2018). Unfortunately, computation of these magnitudes from their definition is not straightforward because they involve multivariate PDF estimation. In this work we show how key quantities that describe redundancy, as the Total Correlation, T, (Watanabe, 1960; Studený & Vejnarová, 1998), and the Mutual Information, I (Cover & Thomas, 2006), naturally appear in the the density destructor framework. Moreover, we will show how, under some conditions, they can be reduced to (easier) univariate operations. Finally, the experiments demonstrate that information theoretic magnitudes may be effective learning criteria for destructive flows, and that estimates of redundancy are obtained via density destructors.

3. Information Theory in Density Destructors

In deep learning, redundancy measures are relevant since they have been linked to the *information bottleneck principle* (Tishby & Zaslavsky, 2015) whereby artificial networks can be classified according to the mutual information between layers. Redundancy reduction is also a relevant self-organization principle in natural neural networks (Barlow, 2001; Malo & Laparra, 2010), and it is also key in unsupervised learning (Hyvärinen et al., 2001). However, these measures are notoriously difficult to compute in high dimensional data.

Fortunately, the ability of density destructors to remove structure makes them appropriate to measure redundancy, as well as to derive convergence rates to the base distribution $\mathcal{P}_z$ in information terms.

3.1. Loss Function in density destructors

The loss function should measure how close the data is to the latent space, $\hat{\mathbf{z}}$, and follows the base distribution $\mathcal{P}_{\mathbf{z}}$. In the latent space we have an advantage because we can choose the target distribution, and typically we choose a distribution such that we have an analytic expression (e.g. Uniform or Gaussian). A usual criterion is to minimize the $\mathrm{D_{KL}}$ divergence between the distribution of the transformed data $\hat{\mathcal{P}}_{\mathbf{z}}$ and our target $\mathcal{P}_{\mathbf{z}}$ such that:

$$J(\hat{\mathbf{z}}) = \mathrm{D_{KL}}\left(\hat{\mathcal{P}}_{\hat{\mathbf{z}}} || \mathcal{P}_{\mathbf{z}}\right) \tag{4}$$

If the target distribution is separable (just a product of marginals), as usually assumed in destructive flows, we can decompose the above expression as:

$$J(\hat{\mathbf{z}}) = \underbrace{T(\hat{\mathbf{z}})}_{\text{Total Corr.}} + \underbrace{J_m(\hat{\mathbf{z}})}_{\text{Marginal KLDs}} \tag{5}$$

using the Pythagorean theorem for $\mathrm{D_{KL}}$ (Cardoso, 2003). While the marginal KLDs can be easily reduced by a simple equalization function, in general, the Total Correlation term, T, is difficult to compute from its definition since it involves integration of unknown multivariate $\hat{\mathcal{P}}_{\mathbf{z}}$. However, in order to use the divergence as an optimization criterion, we do not need to compute the value itself; we just have to minimize it; equivalently, we can enforce the difference of T to be maximum before and after the destructor transformation, which is easy to compute as (Studený & Vejnarová, 1998):

$$\Delta T(\mathbf{x}, \hat{\mathbf{z}}) = \sum_{d=1}^{D} \left(H(\hat{\mathbf{z}}_d) - H(\mathbf{x}_d)\right) - \mathbb{E}_{\mathcal{P}_{\mathbf{x}}}[\log |\nabla_{\mathbf{x}}\mathbf{D}(\mathbf{x})|] \tag{6}$$

The first term of the equation is easy to compute since it only involves operations on univariate distributions. The second term is the expected value of the logarithm of the determinant of the Jacobian of the transformation. In the density destructors framework, this transformation is enforced to be smooth and differentiable. Therefore we can compute the second term by evaluating the Jacobian over the training data using automatic differentiation tools.

3.2. Estimating information theoretic measures

In this section, we show how to compute the information theoretic measures T and I (mutual information) follow-

ing the loss function of density destructors. Similar procedures could be used to compute other useful information quantities, such as D_{KL}, entropy, and negentropy (non-gaussianity).

Total Correlation. T is the information shared among the dimensions of a multidimensional random variable (Watanabe, 1960; Studený & Vejnarová, 1998). We are going to show how by applying a density destructor over $\mathbf{x}$ we can compute $T(\mathbf{x})$ easily as the difference of T between the input and the output, $\Delta T(\mathbf{x}, \hat{\mathbf{z}})$. Assuming the density destructor model has reached convergence, the T in the latent space can be computed easily since we know the distribution. Therefore the T of the original data will be the difference in T in $\mathcal{X}$ plus the T in the latent space $\mathcal{Z}$, i.e. $T(\mathbf{x}) = \Delta T(\mathbf{x}, \hat{\mathbf{z}}) + T(\hat{\mathbf{z}})$. If the chosen distribution for the latent space is uniform or the Gaussian (as it is customary) then the T in the latent space is zero, $T(\hat{\mathbf{z}}) = 0$. Thus, T of the original data is simply $T(\mathbf{x}) = \Delta T(\mathbf{x}, \hat{\mathbf{z}})$, which could be computed using Eq. 6. However, note that the expectation over the data set in Eq. 6 may require many samples and is time consuming.

This inconvenience is solved by the specific density destructor based on Gaussianization proposed in (Laparra et al., 2011). In that case the original PDF is deconstructed through a series of L layers implementing a series of marginal Gaussianization transforms and rotations. Note that both operations in each layer are easy to compute (just a set of univariate sigmoids followed by any orthogonal matrix), and they are straightforward to derive and invert. In (Laparra et al., 2011) we show the convergence of this procedure to the Gaussian target, but more importantly for the current discussion on T, the redundancy of the input is just the sum of the ΔT in each layer:

$$T(\mathbf{x}) = \sum_{i=1}^{L} \Delta T(\mathbf{x}^i) = \sum_{i=1}^{L} J_m(\mathbf{x}^{i+1}) \tag{7}$$

which, as opposed to eq. 6, does not involve any averaging over the whole dataset, and only requires straightforward univariate operations.

Mutual Information. I is the amount of information shared by two datasets $\mathbf{x}$ and $\mathbf{y}$ (Cover & Thomas, 2006). In the density destructor framework, where T is easy to compute (in general through eq. 6, or in Gaussianization through the simpler eq. 7), I can be computed using *three* density destructors as:

$$I(\mathbf{x}, \mathbf{y}) = T([\mathbf{D}_{\mathbf{x}}(\mathbf{x}), \mathbf{D}_{\mathbf{y}}(\mathbf{y})]). \tag{8}$$

where we apply an independent density destructor to each dataset, and then we compute the T for the concatenated variable $[\mathbf{D}_{\mathbf{x}}(\mathbf{x}), \mathbf{D}_{\mathbf{y}}(\mathbf{y})]$ through an extra destructor.

This procedure is possible because I does not change under invertible transformations (as the density destructors) applied separately to each dataset (Cover & Thomas, 2006). Therefore, $I(\mathbf{x}, \mathbf{y}) = I(\mathbf{D}_{\mathbf{x}}(\mathbf{x}), \mathbf{D}_{\mathbf{y}}(\mathbf{y}))$. Since we removed T within each individual dataset by applying individual density destructors, the only redundant information that remains in the concatenated vectors is the one shared by both datasets, then $I(\mathbf{D}_{\mathbf{x}}(\mathbf{x}), \mathbf{D}_{\mathbf{y}}(\mathbf{y})) = T([\mathbf{D}_{\mathbf{x}}(\mathbf{x}), \mathbf{D}_{\mathbf{y}}(\mathbf{y})])$. See appendix for more elaborate proof.

4. Experiments

For all of our experiments , we assume that the latent space is a Gaussian and our algorithm of choice is the Rotation-Based Iterative Gaussianization (RBIG) [1] (Laparra et al., 2011), which finds a sequence of two steps transformations: univariate Gaussianization procedures coupled with a rotation (e.g. independent components analysis, principal components analysis -PCA- or even random rotations). The two operations (marginal gaussianization and rotation) constitute one layer. We chose PCA for the rotation step in the experiments. We use T as an optimization criterion to train the model, and the stopping criterion proposed in (Laparra et al., 2011). Experiments show that this destructive flow estimates T and I effectively compared to other competing algorithms that can be found in the ITE-Toolbox (Szabó, 2014).

4.1. Toy Example: Concentric Circles

We emulated the concentric circles toy example found in (Inouye & Ravikumar, 2018), where a multitude of different density destructors that assume a uniform base distribution were used, i.e. a canonical density destructor. The full process can be broken into two parts: 1) minimize the total correlation assuming a Gaussian distribution using RBIG 2) followed by a histogram CDF transformation to project the data into unit hypercube space. The results shown in fig. 1 demonstrate that RBIG is a worthy candidate, and achieves similar results to those in (Inouye & Ravikumar, 2018), both in terms of approximating the data distribution $\mathcal{X}$ (fig: 1 (a-b)) and of generating samples from the true base distribution $\mathbf{z}$ (fig: 1 (d-e)). We also show the quality of the data inversion in the approximated base density $\hat{z}$ (fig: 1 (b-c)). Figure 1(f) shows the ΔT as the cumulative sum between each layer. Results clearly show that we have reached convergence after removing *all* redundant information.

[1] Please go to the RBIG algorithm homepage for a working implementation along with demonstrations of the IT measures: http://isp.uv.es/rbig.html

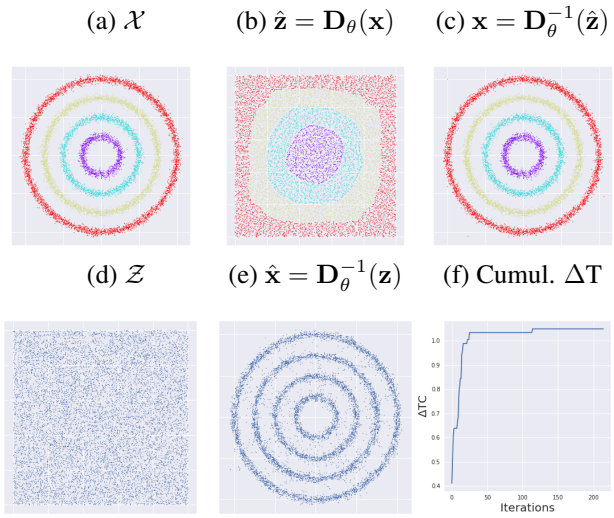

Figure 1. Density estimation of concentric circles using RBIG: (a) original data distribution $\mathcal{X}$, (b) approximated base distribution as a unit cube, (c) the inverse of the destructor $\mathbf{x} = \mathbf{D}_\theta^{-1}(\hat{z})$, (d) samples generated from a uniform distribution $\mathcal{Z}$, (e) the inverse transformation of $\mathbf{z}$ to $\hat{\mathbf{x}}$, and (f) the Cumulative sum of ΔT over the layers (or number of iterations).

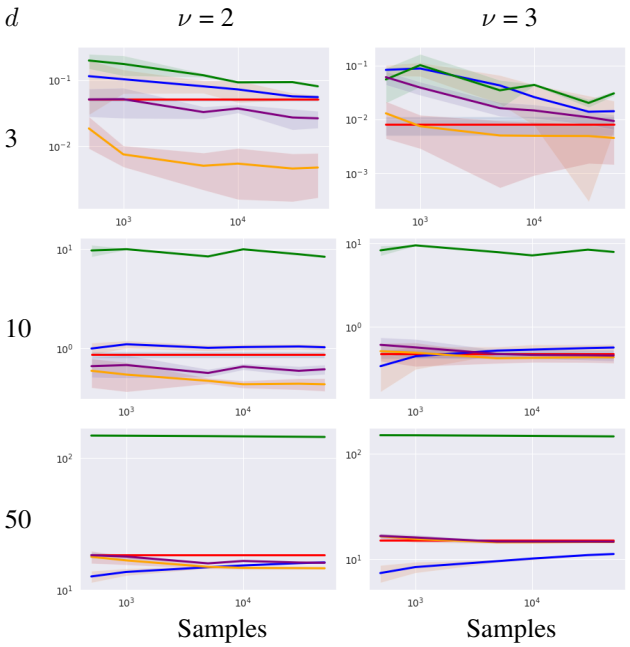

Figure 2. Estimation of T for data drawn from d-dimensional t-Student PDFs with different values of $\nu = 3, 5$ and different number of dimensions $d = 3, 10, 50$ respectively. The mean and standard deviation of the results are given for five trials with samples ranging from 500 to 50,000. **Legend**: Analytical (red), RBIG (purple), expF (orange), and vME (green).

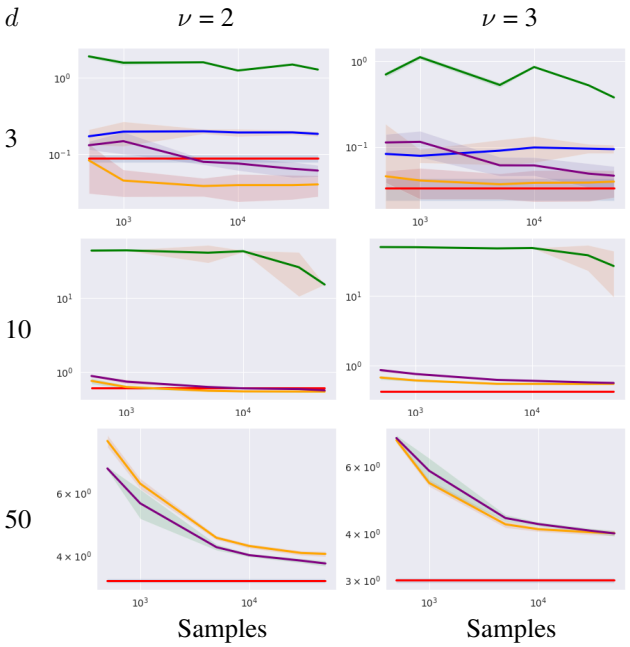

Figure 3. Estimation of I for data drawn from d-dimensional t-Student PDFs with different values of $\nu = 3, 5$ and different number of dimensions $d = 3, 10, 50$ respectively. The mean and standard deviation of the results are given for five trials with samples ranging from 500 to 50,000. **Legend**: Analytical (red), RBIG (purple), expF (orange), and vME (green).

4.2. Total Correlation and Mutual Information

We used the RBIG destructive flow to measure the T and I found within data drawn from multivariate t-Student distributions. Our redundancy estimates are compared with the values found using the k-Nearest Neighbor (kNN) (Goria et al., 2005), the maximum likelihood expectation with the analytical value of the exponential family (expF) (Nielsen & Nock, 2010), and the von Mises Expansion (vME) (Kandasamy et al., 2015) in the implementations given in the ITE-Toolbox (Szabó, 2014). A comparison in Fig. 2 and Fig. 3 are done in terms of distance to the analytical values for T and I in the t-Student (Guerrero-Cusumano, 1998). Any algorithms omitted from the plots resulted in negative values for the respective IT measures. Results for the RBIG destructive flow (in purple) are always the best or close to the best, showing that it is a robust method to compute multivariate information theoretic measures.

5. Conclusion

We connected the density destructors framework introduced in (Inouye & Ravikumar, 2018) with classical information theory. This connection allows the use of Total Correlation as learning criterion for destructive flows and to compute non-trivial information theoretic quantities via density destructors. We chose a particular density destructive flow for multivariate Gaussianization and reported empirical evidence of performance in simulated examples.

References

Ballé, J., Laparra, V., and Simoncelli, E. P. Density modeling of images using a generalized normalization transformation. *ICLR*, abs/1511.06281, 2016.

Barlow, H. Redundancy reduction revisited. *Network: Comp. Neur. Syst.*, 12(3):241–253, 2001.

Cardoso, J.-F. Dependence, correlation and gaussianity in independent component analysis. *Journal of Machine Learning Research*, 4:1177–1203, 2003.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS 29*, pp. 2172–2180. 2016.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory, 2nd Edition*. Wiley, 2006.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2017.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *NIPS*, 2014.

Goria, M., Leonenko, N., Mergel, V., and Inverardi, P. A new class of random vector entropy estimators and its applications in testing statistical hypotheses. *J. Nonparame. Stat.*, 17(3):277–297, 2005.

Guerrero-Cusumano, J. L. Measures of dependence for the multivariate t distribution. *Comm. Stat. - Theory and Methods*, 27(12):2985–3006, 1998.

Hyvärinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis*. Wiley, NY, USA, 2001.

Inouye, D. I. and Ravikumar, P. Deep density destructors. In *ICML*, 2018.

Kandasamy, K., Krishnamurthy, A., Póczos, B., Wasserman, L. A., and Robins, J. M. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *NIPS*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Laparra, V., Camps-Valls, G., and Malo, J. Iterative gaussianization: From ica to random rotations. *IEEE Transactions on Neural Networks*, 22:537–549, 2011.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. In *ICLR*, 2016.

Malo, J. and Laparra, V. Psychophysically tuned divisive normalization factorizes the PDF of natural images. *Neural computation*, 22(12):3179–3206, 2010.

Nielsen, F. and Nock, R. Entropies and cross-entropies of exponential families. *IEEE ICIP*, pp. 3621–3624, 2010.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *ICML*, 2015.

Studený, M. and Vejnarová, J. The multiinformation function as a tool for measuring stochastic dependence. In *Proc. NATO Adv. Study Inst. Learn. Graph. Models*, pp. 261–297. Kluwer, 1998.

Szabó, Z. Information theoretical estimators toolbox. *Journal of Machine Learning Research*, 15:283–287, 2014.

Timme, N. M. and Lapish, C. C. A tutorial for information theory in neuroscience. In *eNeuro*, 2018.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. *IEEE Inf. Theory Workshop*, pp. 1–5, 2015.

Watanabe, M. S. Information theoretical analysis of multivariate correlation. *IBM J. Res. Develop.*, 4:66–82, 1960.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE ICCV*, 2017.

6. Appendix: Mutual Information from density destructors

Recall the definitions for I and T as:

$$\begin{aligned} I(\mathbf{x},\mathbf{y}) &= H(\mathbf{x}) + H(\mathbf{y}) - H([\mathbf{x},\mathbf{y}]) \\ T(\mathbf{x}) &= \sum_{d=1}^{D} H(\mathbf{x}_d) - H(\mathbf{x}) \end{aligned}$$

If we apply two separate density destructor transforms on $\mathbf{x} \in \mathbb{R}^{D_x}$ and $\mathbf{y} \in \mathbb{R}^{D_x}$, we achieve the new datasets $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ respectively, where $\sum_{d=1}^{D_x} H(\hat{\mathbf{x}}_d) = H(\hat{\mathbf{x}})$ and $\sum_{d=1}^{D_y} H(\hat{\mathbf{y}}_d) = H(\hat{\mathbf{y}})$. We can rewrite the mutual information in terms of the transformed versions like so:

$$I(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \sum_{d=1}^{D_x} H(\hat{\mathbf{x}}_d) + \sum_{d=1}^{D_y} H(\hat{\mathbf{y}}_d) - H([\hat{\mathbf{x}},\hat{\mathbf{y}}])$$

For convenience lets assume that we stack $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ into a single vector $\hat{\mathbf{v}} = [\hat{\mathbf{x}},\hat{\mathbf{y}}]$, then we can combine the summations for the marginals into a single term that runs through all the dimensions of $\hat{\mathbf{v}}$:

$$I(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \sum_{d=1}^{D_x+D_y} H(\hat{\mathbf{v}}_d) - H(\mathbf{v})$$

And then applying the definition of total correlation:

$$I(\hat{\mathbf{x}},\hat{\mathbf{y}}) = T(\mathbf{v}) = T([\hat{\mathbf{x}},\hat{\mathbf{y}}]),$$

leading to eq. 8.

So we see that the mutual information for two destructed variables is the same as the total correlation of the two destructed variables stacked into a single vector. The mutual information is invariant under smooth, invertible transformations, as is the case for any density destructors applied to $\mathbf{x}$ and $\mathbf{y}$. The role of these initial destructors is removing redundant information between the different variables within each dataset. Once we did that, the remaining redundancy (in the stacked vector, which will be computed by the third destructor) is the information shared by the original variables. If this third destructor is chosen to be the Rotation-Based Iterative Gaussianization (Laparra et al., 2011), we have an easy way, eq.7, to calculate the mutual information between two multivariate variables of arbitrary dimension.