Evaluating Disentangled Representations

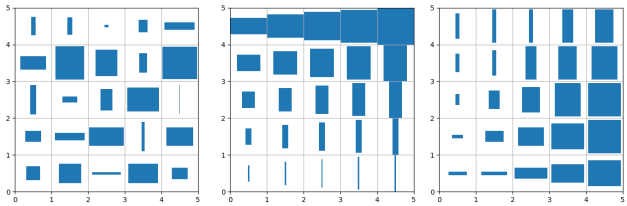
Anna Sepliarskaia [1] **Julia Kiseleva** [2] **Maarten de Rijke** [1]

Abstract

There is no generally agreed upon definition of disentangled representation. Intuitively, a disentangled representation captures and separates a few factors of variation that generate the data. Disentangled representations are useful for many tasks such as reinforcement learning, transfer learning, and zero-shot learning. To evaluate disentangled representations several metrics have been proposed. However, theoretical guarantees for existing metrics of disentanglement are still missing, and in some applications, existing metrics do not have a consistent correlation with the outcomes of a qualitative study of the disentanglement of the learned representations. In this paper, we analyze metrics of disentanglement and their properties. Specifically, we analyze whether available metrics satisfy two desirable properties: (1) assign a high score to representations that are disentangled according to the definition; and (2) assign a low score to representations that are entangled according to the definition. We show that most of the current metrics do not satisfy at least one of these properties. Consequently, we propose a new definition for a metric of disentanglement that satisfies both of the properties.

1. Introduction

Algorithms for learning representations are crucial for a variety of machine learning tasks, including image classification (Vincent et al., 2008; Hinton & Salakhutdinov, 2006) and image generation (Goodfellow et al., 2014; Makhzani et al., 2015). One type of representation learning algorithm is designed to create a disentangled representation. While there is no standardized definition of a disentangled representation, the key intuition is that a disentangled representation should capture and separate the generative factors (Bengio et al., 2013; Higgins et al., 2018). In this paper, we assume that the *generative factors* of the dataset are interpretable factors that describe every sample from the dataset.



(a) Example of a dataset containing rectangles with width w and height h from the uniform distribution $U(0, 1)$

(b) Change in the elements of the dataset caused by a change in the disentangled latent representation. Horizontally, the change is caused by a change in the first hidden factor; vertically, the change is caused by another hidden factor.

(c) Change in the elements of the dataset caused by a change in the entangled latent representation. Horizontally, the change is caused by a change in the first hidden factor; vertically, the change is caused by another hidden factor.

Figure 1. Example of a dataset and disentangled and entangled representations.

Consider, for example, Fig. 1a, where we show a dataset containing rectangles of different shapes. There are two generative factors for this dataset: the length and width of the rectangles. In the disentangled latent representation of this dataset we can choose two latent factors. One of these factors is an invertible function of the length of the rectangles. Another is an invertible function of the width of the rectangles. In such a representation a change of one latent factor leads to a change only in one generative factor (see Fig. 1b). In an entangled representation a change in one latent factor can lead to a change in the length and width of the rectangles (see Fig. 1c).

Learning a disentangled representation is an important step towards better representation learning because a disentangled representation contains information about elements in a dataset in an interpretable and compact structure (Bengio et al., 2013; Higgins et al., 2018). Interpretability of a representation helps in tasks where users interact with a system, as they understand how it works and can provide

[*]Equal contribution [1]University of Amsterdam, Amsterdam, The Netherlands [2]Microsoft Research AI, Seattle, WA, USA. Correspondence to: Anna Sepliarskaia <a.sepliarskaia@uva.nl>, Julia Kiseleva <Julia.kiseleva@microsoft.com>, Maarten de Rijke <derijke@uva.nl>.

informative feedback. Moreover, learning a disentangled representation helps for tasks where state-of-the-art machine learning-based approaches still struggle but where humans excel. Such scenarios include learning with knowledge transfer (Tommasi et al., 2010; Huang & Frank Wang, 2013; Pan et al., 2010), zero-shot inference (Lampert et al., 2009; Romera-Paredes & Torr, 2015) and supervised learning (Szegedy et al., 2013; Nguyen et al., 2015).

Therefore, the development of an algorithm that learns disentangled representations has become an active area of research (Detlefsen & Hauberg, 2019; Dezfouli et al., 2019; Lorenz et al., 2019). To evaluate these algorithms several metrics of disentanglement have been proposed. Such metrics are being used to help select representation learning algorithms that create representations with the highest degree of disentanglement. However, it has been shown that scores of existing metrics of disentanglement need not correlate with the outcomes of a qualitative study of the disentanglement of learned representations (Abdi et al., 2019); moreover, it is not clear which metric should be preferred.

Important steps have been taken towards a formal evaluation of disentangled representations. For example, metrics of disentanglement have been compared through an experimental study on several datasets (Locatello et al., 2018). A framework for the evaluation of disentangled representations (Eastwood & Williams, 2018) has been put forward. And a formal definition of disentangled representations using group theory has been proposed (Higgins et al., 2018). We continue this line of research by providing an analysis of theoretical properties of disentanglement metrics.

Our key contributions in this paper are:

- We review existing metrics of disentanglement and discuss their fundamental properties.
- We propose a new metric of disentanglement with theoretical guarantees, and establish its fundamental properties.
- We provide an experimental comparison of the newly proposed metric with existing metrics.[1]

2. Background and Notation

2.1. Representation learning

There are different types of representation learning algorithm, but usually, an algorithm for learning disentangled representations consists of two parts: an encoder and a decoder. An *encoder* is a function:

$$f_e : \mathbb{R}^d \to \mathbb{R}^N,\ \mathbf{c} = f_e(\mathbf{x}), \tag{1}$$

where $\mathbf{c}$ is a latent representation of the data sample $\mathbf{x}$. Typically, the dimension of the latent representation is much smaller than the dimension of the data. A *decoder* is a function:

$$f_d : \mathbb{R}^N \to \mathbb{R}^d,\ f_d(f_e(\mathbf{x})) \sim \mathbf{x}, \tag{2}$$

where $f_d(f_e(\mathbf{x}))$ should be close to $\mathbf{x}$. Thus, the latent representation should contain almost all the information that is contained in the original data.

2.2. Ground truth generative factors

We assume that a dataset was generated using a generative process that generates *generative factors*. We define the *generative factors* of a dataset in the following way:

Definition 1. The *generative factors* of a dataset are interpretable factors that describe the difference between any two samples from X.

Consider, for example, a dataset containing rectangles of different shapes presented, as illustrated in Fig. 1a. The generative factors for this dataset are the length and width of the rectangles.

We further define the *ground truth* generative factors:

Definition 2. The *ground truth* generative factors are the values of generative factors for a given collection.

This means that we assume that for each sample $\mathbf{x} \in X$ of the dataset, the values of the generative factors $\mathbf{z} \in \mathbb{R}^K$ are known during the evaluation.

3. Metrics of Disentanglement of Representations

The main purpose of this paper is to analyze the existing metrics of disentangled representations, which is done in this section. Though there is no universally accepted definition of disentanglement, most metrics are based on the definition proposed in (Bengio et al., 2013) and reflect characteristics of a disentangled representation in accordance with this definition. In particular, there are two main characteristics of disentangled representations and existing metrics can be divided into two groups, depending on which characteristic they reflect. We analyze these two groups of metrics below. In particular, we analyze if existing metrics satisfy the following properties:

Property 1. *A metric gives a high score to all representations that satisfy the characteristic that the metric reflects.*

Property 2. *A metric gives a low score for all representations that do not satisfy the characteristic that the metric reflects.*

At the end of this section, we discuss the difference between the two characteristics of disentangled representations.

[1] A comparison of methods for learning a disentangled representation is beyond the scope of this paper.

3.1. BetaVAE, FactorVAE and DCI

In this subsection, we analyze metrics that reflect the following characteristic of disentangled representations.

Characteristic 1. In a *disentangled representation* a change in one latent dimension corresponds to a change in one generative factor while being relatively invariant to changes in other generative factors.

3.1.1. Definition of BetaVAE

The algorithm that calculates *BetaVAE* (Higgins et al., 2017) consists of the following steps:

1. Choose a generative factor z_k.
2. Generate a batch of pairs of vectors for which the value of z_k within the pair is equal, while other generative factors are chosen randomly:
$$(\mathbf{p}_1 = \langle z_{1,1}, \dots, z_{1,K}\rangle, \mathbf{p}_2 = \langle z_{2,1}, \dots, z_{2,K}\rangle), \\ z_{1,k} = z_{2,k}$$
3. Calculate the latent code of the generated pairs: $(\mathbf{c}_1 = f_e(g(\mathbf{p}_1)), \mathbf{c}_2 = f_e(g(\mathbf{p}_2)))$
4. Calculate the absolute value of the pairwise differences of these representations:
$$\mathbf{e} = \langle \left|c_{1,1} - c_{2,1}\right|, \dots, \left|c_{1,N} - c_{2,N}\right| \rangle$$
5. The mean of these differences across the examples in the batch gives one training point for the linear regressor that predicts which generative factor was fixed.
6. BetaVAE is the accuracy of the linear regressor.

3.1.2. Definition of FactorVAE

The idea behind FactorVAE (Kim & Mnih, 2018) is very similar to BetaVAE. The main difference between them concerns how a batch of examples is generated to obtain a variation of latent variables when one generative factor is fixed. Another difference is the classifier that predicts which generative factor was fixed using the variation of latent variables. *FactorVAE* can be calculated by performing the following steps:

1. Choose a generative factor z_k.
2. Generate a batch of vectors for which the value of z_k within the batch is fixed, while other generative factors are chosen randomly.
3. Calculate latent codes of vectors from one batch.
4. Normalize each dimension in the latent representation by its empirical standard deviation over the full data.
5. Take the empirical variance in each dimension of these normalized representations.
6. The index of the dimension with the lowest variance and the target index k provides one training point for the classifier.
7. FactorVAE is the accuracy of the classifier.

3.1.3. DCI: Disentanglement, Completeness and Informativeness

Eastwood & Williams (2018) propose to use a metric of disentangled representations, which we call DCI, that is calculated as follows:

1. First, the *informativeness* between c_i and z_j is calculated. To determine the informativeness between c_i and z_j, Eastwood & Williams (2018) suggest training K regressors. Each regressor f_j predicts z_j given $\mathbf{c}$ ($\hat{z}_j = f_j(\mathbf{c})$) and can provide an importance score $P_{i,j}$ for each c_i. The normalized importance score obtained by regressor f_j for variable c_i is used as the informativeness between c_i and z_j:
$$I_{i,j} = \frac{P_{i,j}}{\sum_{k=0}^{k=K} P_{i,k}}.$$
2. For each latent variable its score of disentanglement is calculated as follows:
$$H_K(I_i) = 1 + \sum_{k=1}^{K} I_{i,k} \log_K I_{i,k}.$$
3. The weighted sum of the obtained scores of disentanglement for the latent variables is DCI:
$$\text{DCI}(\mathbf{c}, \mathbf{z}) = \sum_i \left(\rho_i \cdot H_K(I_i)\right), \tag{3}$$
where $\rho_i = \sum_j P_{i,j} / \sum_{ij} P_{i,j}$.

3.1.4. Analysis of whether metrics satisfy the property 1

Fact 1. BetaVAE *and* FactorVAE *do not satisfy Property 1.*

Proof. In a representation that satisfies Characteristic 1, there may be several latent factors that correspond to changes in the same generative factor. Consequently, these latent factors have variation 0 when the corresponding generative factor is fixed. That is why the classifier cannot distinguish between these latent factors and its accuracy is less than 1. Consequently, the BetaVAE and FactorVAE metrics return scores of less than 1 for a perfectly disentangled representation. □

Fact 2. *DCI does not satisfy Property 1.*

Proof. We argue that using entropy as a score of disentanglement of one latent variable is not correct. Indeed, a score of disentanglement of c_i should be high when c_i reflects one generative factor well, while it reflects other generative factors equally poorly. However, since the distribution may be close to uniform for these generative factors, the entropy is large. Let us provide an example that is built on this

observation. Suppose there are 11 generative factors, and 11 is the dimension of the latent representation. Each latent factor c_i captures primarily a generative factor z_i:

$$I_{i,i} = 0.8,\ I_{i,k} = 0.02,\ k \neq i.$$

Then, the DCI score is 0.6, so the DCI assigns a small score to a representation that satisfies Characteristic 1. □

3.1.5. Analysis of whether metrics satisfy the property 2

Fact 3. BetaVAE *does not satisfy Property 2.*

Proof. As a proof, we give a counterexample. Let us consider all training points for a linear classifier with a fixed label. The classifier can learn to map some regularity in the values of features to the right class. However, there do not exist any constraints on this regularity. The classifier can learn to map samples with a value of 0 of some feature to the correct class. But the classifier can also learn to map samples with other patterns in the feature values to the correct class. Given this intuition, let us consider the following example. Suppose there are 3 generative factors from a uniform distribution and the dimension of the latent representation is 3. Assume that the latent variables are equal to the generative factors with the following probabilities:

$$p_1 = (0.5, 0.5, 0),\ p_2 = (0, 0.5, 0.5),\ p_3 = (0.5, 0, 0.5).$$

We generate 10,000 training points with a batch size of 128. The accuracy of the linear classifier is equal to 0.9967 in this case, but the latent representation does not satisfy Characteristic 1. This shows that BetaVAE does not satisfy Property 2. □

Fact 4. FactorVAE *does not satisfy Property 2.*

Proof. First, let us analyze the algorithm that calculates the FactorVAE score. Suppose that for each generative factor z_j there is a latent variable c_{i_j} that correlates with z_j more than other variables in the latent code $\mathbf{c}$. In this case, when z_j is fixed and the batch size is large enough, the variation in the latent factor c_{i_j} will be smaller than the variation in other latent factors. Consequently, the classifier will have high accuracy. Given this intuition, let us consider the following example. Suppose there are 3 generative factors from a Gaussian distribution with $\mu = 0, \sigma = 1$, and each latent variable is a weighted sum of the generative factors:

$$c_1 = 0.5 \cdot z_1 + 0.4 \cdot z_2 + 0.5 \cdot z_3$$
$$c_2 = 0.4 \cdot z_1 + 0.5 \cdot z_2 + 0.5 \cdot z_3$$
$$c_3 = 0.4 \cdot z_1 + 0.4 \cdot z_2 + 0.6 \cdot z_3.$$

We generate 10,000 training points with a batch size of 128. The FactVAE disentanglement score is equal to 1 in this case, but the representation does not satisfy Characteristic 1. This shows that FactoVAE does not satisfy Property 2. □

Fact 5. *DCI does not satisfy Property 2.*

Proof. We give a counterexample, which is built on the fact that the weighted sum in Eq. 3 can be large if only one latent variable is disentangled, while the other latent variables are entangled and do not capture any information about generative factors. Suppose there are 2 generative factors and the dimension of the latent representation is 2, and the matrix of informativeness is the following:

$$P_{0,0} = 1, P_{0,1} = 0, P_{1,1} = 0.09, P_{1,0} = 0.01.$$

In this case, the DCI score is 0.957. This counterexample shows that the DCI score can be close to 1 for the representation does not satisfy Characteristic 1. □

3.2. SAP and MIG metrics

In this subsection, we analyze metrics that reflect the following characteristic of disentangled representations.

Characteristic 2. In a *disentangled representation* a change in a single generative factor leads to a change in a single factor in the learned representation.[2]

3.2.1. SAP score: Separated Attribute Predictability

Kumar et al. (2017) provide a metric of disentanglement that is calculated as follows:

1. Compute a *matrix of informativeness* $I_{i,j}$, in which the ij-th entry is the linear regression or classification score of predicting the j-th generative factor using only the i-th variable in the latent representation.
2. For each column in the matrix of informativeness $I_{i,j}$, which corresponds to a generative factor, calculate the difference between the top two entries (corresponding to the top two most predictive latent factors). The average of these differences is the final score, which is called the SAP:

$$\text{SAP}(\mathbf{c}, \mathbf{z}) = \frac{1}{K} \sum_k \left(I_{i_k,k} - \max_{l \neq i_k} I_{l,k} \right),$$

where $i_k = \arg\max_i I_{i,k}$.

3.2.2. MIG: Mutual Information Gap

Chen et al. (2018) propose a disentanglement metric, Mutual Information Gap (MIG), that uses mutual information between the j-th generative factor and the i-th latent variable

[2] This property of representations is also called *completeness* (Eastwood & Williams, 2018).

as a notion of informativeness between them. The *mutual information* between two variables c and z is defined as

$$I(c; z) = H(z) - H(z|c),$$

where $H(z)$ is the entropy of the variable z. Mutual information measures how much knowing one variable reduces uncertainty about the other. A useful property of mutual information is that it is always non-negative $I(c; z) > 0$. Moreover, $I(c; z)$ is equal to 0 if and only if c and z are independent. Also, mutual information achieves its maximum if there exists an invertible relationship between c and z. The following algorithm calculates the MIG score:

1. Compute a *matrix of informativeness* $I_{i,j}$, in which the ij-th entry is the mutual information between the j-th generative factor and the i-th latent variable.
2. For each column of the score matrix $I_{i,j}$, which corresponds to a generative factor, calculate the difference between the top two entries, and normalize it by dividing by the entropy of the corresponding generative factor. The average of these normalized differences is the MIG score:

$$\text{MIG}(\mathbf{c}, \mathbf{z}) = \frac{1}{K} \sum_k \frac{I_{i_k,k} - \max_{l \neq i_k} I_{l,k}}{H(z_k)},$$

where $i_k = \arg\max_i I_{i,k}$.

3.2.3. Analysis of whether metrics satisfy the property 1

Fact 6. *SAP does not satisfy Property 1.*

Proof. We claim that it is incorrect to use the R^2 score of linear regression as informativeness between latent variables and generative factors. Indeed, a linear regression cannot capture non-linear dependencies. Thus, informativeness, which is calculated using the R^2 score of a linear regression, may be low if each generative factor is a non-linear function of some latent variable. Let us give an example that is built on this observation. Suppose there are 2 generative factors from the uniform distribution $U([-1, 1])$ and the dimension of the latent representation is 2. Let us assume the latent variables are obtained from the generative factors according to the following equations:

$$c_1 = z_1^{15},\ c_2 = z_2^{15}.$$

For this representation, we generate 10,000 examples and obtain the SAP score equal to 0.32. It proves that SAP can assign a low score to a representation that satisfies Characteristic 2. □

Fact 7. *MIG satisfies Property 1.*

Proof. Indeed, in a disentangled representation each generative factor is primarily captured in only one latent dimension. This means that for each generative factor z_j, there is exactly one latent factor c_{i_j} for which z_j is a function of c_{i_j}: $z_j \sim f(c_{i_j})$. Therefore,

$$I_{i_j,j} = H(z_j) - H(z_j|c_{i,j}) \sim H(z_j),$$

whereas for other latent variables $I_{k,j} = I(c_k, z_j) \sim 0$. Consequently, according to MIG, the score of disentanglement of each generation factor is close to 1:

$$\frac{I_{i_j,j} - \max_{k \neq i_j} I_{k,j}}{H(z_j)} \sim 1. \tag{4}$$

Therefore, the average of these scores is also close to 1. This shows that MIG always assigns a high score to a representation that satisfies Characteristic 2. □

3.2.4. Analysis of whether metrics satisfy the property 2

Fact 8. *SAP does not satisfy Property 2.*

Proof. A high SAP score indicates that the majority of generative factors is captured linearly in only one latent dimension. However, the SAP metric does not penalize the existence of several latent factors that capture the same generative factor non-linearly. Let us consider the following example. Suppose there are 2 generative factors from the uniform distribution $U([-1, 1])$, and the dimension of the latent representation is 3. Let us assume that the latent factors are obtained from the generative factors according to the following equations:

$$c_1 = z_1,\ c_2 = z_1^{25} + z_2^{25},\ c_3 = z_2.$$

For this latent representation, a change in each generative factor leads to a change in several latent factors, but the SAP score is equal to 0.98. This shows that the SAP score can be close to 1 for a latent representation that does not satisfy Characteristic 1. □

Fact 9. *MIG satisfies Property 2.*

Proof. A high MIG score indicates that the majority of generative factors is captured in only one latent dimension. Consequently, a change in one of the generative factors entails a change primarily in only one latent dimension. □

A summary of the results of our analysis is given in Table 1.

Table 1. Summary of facts about proposed metrics of disentangled representations.

Metric	Satisfies Property 1	Satisfies Property 2
BetaVAE	No	No
FactorVAE	No	No
DCI	No	No
SAP	No	No
MIG	Yes	Yes

3.3. Difference between Characteristics 1 and 2

The Characteristics 1 and 2 of a disentangled representation have important differences. Indeed, a representation in which several latent factors capture one common generative factor satisfies a Characteristic 1, but not a Characteristic 2. On the other hand, a representation in which a latent variable captures multiple generative factors while there are no other latent variables that capture these generative factors does not satisfy Characteristic 1, but satisfies Characteristic. 2.

Consider, for example, the following latent representation of dimension 4 of the dataset containing rectangles of different shapes shown in Fig. 1a:

$$z_1 = x,\ z_2 = x^2,\ z_3 = y,\ z_4 = y^3$$

where x is the length of a rectangle, while y is the width of a rectangle. It satisfies Characteristic 1, but not a Characteristic 2. Conversely, any one-dimensional latent representation of the same dataset would satisfy Characteristic 2, but not necessarily Characteristic 1.

4. A New Metric of Disentanglement, DCIMIG

The previous metrics were designed to reflect only one out of two characteristics of disentangled representations. We believe that a metric should reflect both of them: Characteristics 1 and 2. Moreover, following (Eastwood & Williams, 2018) we think that the metric should also reflect the *informativeness* of a representation.

Formally, this means that we believe that a *disentangled representation* satisfies the following characteristic.

Characteristic 3. In a disentangled representation, we can choose a subset of latent variables: $c' = \{c_{i_1}, \dots, c_{i_K}\}$, that satisfy Characteristic 1 and Characteristic 2. Moreover, a disentangled representation should contain nearly all information about generative factors, i.e., it should have a high degree of *informativeness* (Eastwood & Williams, 2018).

With this in mind, we propose a new metric of disentanglement of representation, called DCIMIG. The previously introduced MIG metric is a good starting point because it is the only metric that satisfies Properties 1 and 2. However, the MIG metric was not created to reflect Characteristic 1 and *informativeness*. As a consequence, the MIG metric does not penalize latent representations in which latent factors capture several generative factors; and the MIG metric does not capture the informativeness of latent representations as it equally penalizes latent representations for not capturing informative generative factors and for not capturing non-informative generative factors. That is why we propose a new metric, *DCIMIG*, that captures the Disentanglement, Completeness (see footnote 2) and Informativeness of a representation using the Mutual Information Gap.

4.1. Definition of DCIMIG

Following MIG, we create a matrix of informativeness $I_{i,j}$, in which the ij-th entry is the mutual information between the j-th generative factor and the i-th variable in the latent representation. The following steps for calculating DCIMIG differ from the steps suggested in MIG:

1. For each latent variable c_i, find the generative factor z_{j_i} that it reflects the most: $j_i = \arg\max_j I_{i,j}$.
2. Calculate the disentanglement for each latent variable: $D_i = I_{i,j_i} - \max_{k \neq j_i} I_{i,k}$.
3. For each generative factor z_j, find the most disentangled latent factor c_{k_j}, that reflects z_j: $k_j = \arg\max_{l \in \mathbb{I}_j} D_l$, where $\mathbb{I}_j = \{i : z_{j_i} = z_j\}$.
4. For each generative factor z_j, calculate the disentanglement score D_j^z, which is equal to D_{k_j} if there is at least one latent factor, that captures z_j, otherwise, it is 0.
5. Finally, the *disentanglement score* of a latent representation according to DCIMIG is the normalized sum of D_j^z:

$$\text{DCIMIG}(\mathbf{c}, \mathbf{z}) = \frac{\sum_{j=1}^{K} D_j^z}{\sum_{j=1}^{K} H(z_j)},$$

where $H(z_j)$ is the entropy of z_j.

4.2. Analysis of whether DCIMIG satisfies Property 1

Fact 10. *DCIMIG satisfies Property 1.*

Proof. Indeed, in a representation that satisfies Characteristic 3, there is a subset c' of latent variables, in which each latent variable is sensitive to changes in one generative factor only. Moreover, for each generative factor z_j there is only one latent variable $c_{i_j} \in c'$ that captures the changes in z_j. Consequently, c_{i_j} is a function of z_j: $c_{i_j} = f_j(z_j)$, while the other latent factors are invariant to changes in z_j. This means that, $D_{i_j} = I_{i_j,j} - \max_{k \neq j} I_{i_j,k} = I_{i_j,j}$. Also, the disentangled representation should have a high degree of *informativeness*. Consequently, the latent variables in c' should capture all the information contained in z_j. But only $c_{j,i}$ contains some information about z_j. Therefore,

$I_{i_j,j} = H(z_j)$, and $D_j^z = H(z_j)$. Consequently, DCIMIG is equal to 1 in this case. □

4.3. Analysis of whether DCIMIG satisfies Property 2

Fact 11. *DCIMIG satisfies Property 2.*

Proof. When a representation does not satisfy Characteristic 3 for the majority of informative generative factors z', we cannot find a factor in the latent representation that reflects only this factor. There are 2 cases for the generative factors from z'. In the first case, there is no latent factor that captures the generative factor $z_j \in z'$. In that case, D_j^z is equal to 0. The second case is characterized by the fact that there is a latent factor that captures a generative factor, but this latent factor also captures other generative factors. In that case the disentangled score of this latent factor D_{i_j} is small, and consequently, D_j^z is small. □

4.4. Difference between DCIMIG, DCI and MIG

While DCIMIG is similar to the DCI and MIG metrics it has important differences from them. First, DCI does not penalize representations if there is a generative factor that is not captured by any latent variable; it penalizes representations that contain both disentangled latent variables and entangled ones. DCIMIG penalizes representations that do not capture some generative factors; the DCIMIG score is high if there is a subset of disentangled latent variables that captures all generative factors; DCIMIG does not penalize representations if there are other latent variables that are entangled. Second, MIG does not penalize representations if a latent variable captures several generative factors; if a representation only captures a subset of all the generative factors, then MIG assigns the same score no matter which subset is captured, whether the generative factors that have been captured have high or low entropy. In contrast, DCIMIG does distinguish between representations that capture different subsets of generative factors.

5. Experiments

In this section we describe our experiments. We compare the proposed metric, DCIMIG, with two conceptually similar metrics: MIG and DCI. To this end, we consider two experimental conditions. Due to space constraints, we only present the experimental setup and implications of the experiments; a detailed description of the results is given in Appendix A.

5.1. Spearman rank correlation between metrics

Following (Locatello et al., 2018), we explore how the metrics agree. We provide tables with correlation scores between the metrics in Appendix A.1.

Results. Not surprisingly, we observe that DCIMIG strongly correlates with MIG and DCI on the dSprites (Higgins et al., 2017) and Cars3D (Reed et al., 2015) datasets. However, these datasets are artificial datasets, on which all metrics are correlated and the MIG and DCI metrics are strongly correlated. Conceptually , DCIMIG is between these two metrics, and consequently, it also strongly correlates with them on these artificial datasets.

5.2. Different behavior of DCIMIG, MIG and DCI

In this section we examine the difference between DCIMIG and MIG, and DCIMIG and DCI. In particular, we show examples where DCIMIG and MIG, and DCIMIG and DCI, do not agree on which of the two representations is the most disentangled representation. This experiment provides a better understanding of the differences between the metrics and what exactly the metrics penalize. In Appendix A.2 we explain the behavior of the metrics by exploring the matrices of informativeness of representations. Below we give the details of the experiment and its results.

5.2.1. Comparison of DCIMIG and MIG

In this section, we provide an example of two representations for which DCIMIG and MIG disagree over which of the two representations is the more disentangled representation, and we provide the outcomes of the comparison of the representations. In particular, we compare two representations obtained by the β-TCVAE model (Chen et al., 2018) and the AnnealedVAE model (Burgess et al., 2018), trained on the Cars3D dataset. In (Locatello et al., 2018) they are numbered 10402 and 10587, respectively; their matrices of informativeness are given in the Appendix, in Figures 3a and 3b correspondingly.

Implications. MIG provides high scores to compact latent representations, while DCIMIG does not penalize representations in which several latent factors capture the same generative factor.

5.2.2. Comparison of DCIMIG and DCI

In this section, we provide an example of two representations for which DCIMIG and DCI disagree over which of the two representations is the more disentangled representation; and provide the outcomes of the comparison of the representations. In particular, we compare the representations obtained by two models: the two β-TCVAE models trained on the Cars3D dataset; within (Locatello et al., 2018) they are numbered 10429 and 10264. Their matrices of informativeness are given in the Appendix, in Figures 3a and 3b correspondingly.

Implications. DCI provides high scores to the latent representations in which all the latent factors are disentangled,

independently of whether the latent factors capture different generative factors. While DCIMIG provides high scores to representations that contain for each generative factor at least one latent disentangle latent dimension, which reflects this factor and it is indifferent as to whether the remaining latent factors are entangled or disentangled.

6. Related Work

This paper is relevant to two research directions: the formulation of a notion of disentangled representation and the analysis of differences between proposed metrics of disentangled representations.

A definition of disentangled representation is presented by Higgins et al. (2018), who propose to call a representation disentangled if it is consistent with transformations that characterized the dataset. In particular, Higgins et al. (2018) suggested that transformations that change only some properties of elements in the dataset, while leaving other properties unchanged, give the structure of a dataset. Desirable properties of a disentanglement metric are formulated by Eastwood & Williams (2018); they are *disentanglement*, *completeness*, and *informativeness*. Eastwood & Williams (2018) claim that a good representation should satisfy all of these properties, namely (1) if a representation is good, then change in one latent factor should lead to change in one generative factor, (2) a change in one generative factor should lead to a change in one latent factor, and (3) a latent representation should contain all information about the generative factors. Therefore, Eastwood & Williams (2018) propose three metrics to satisfy each of the properties listed. However, the proposed metrics were not analyzed — a gap that we fill.

Several papers analyze the differences between metrics of disentanglement through experimental studies (Locatello et al., 2018; Chen et al., 2018). For example, Locatello et al. (2018) train 12,000 models that cover the most prominent methods and evaluate these models using existing metrics of disentanglement. The study shows that the metrics are correlated, but the degree of correlation depends on the dataset. It is important to note that their experimental results are consistent with our theoretical findings: the BetaVAE (Higgins et al., 2017) and FactorVAE (Kim & Mnih, 2018) metrics are strongly correlated with each other; and the SAP (Kumar et al., 2017), MIG (Chen et al., 2018), DCI (Eastwood & Williams, 2018) scores are also strongly correlated. Locatello et al. (2018) take an important step towards the evaluation of methods to create disentangled representations, however, the properties of the metrics are not analyzed theoretically. Chen et al. (2018) take a step in this direction, but only analyze the BetaVAE, FactorVAE and MIG metrics. Chen et al. (2018) compare metrics by analyzing robustness to the choice of the hyperparameters during experiments. The experimental findings are quite similar to ours: BetaVAE is a very optimistic metric and assigns high scores to entangled representations.

To summarize, the key distinctions of our work compared to previous efforts are: (1) a broad coverage, in-depth analysis of previously proposed metrics of disentanglement, and (2) a proposal of a single metric of disentanglement that reflects all properties of previously proposed ones and has theoretical guarantees.

7. Conclusion

In recent years, several models have been developed to obtain disentangled representations (Yu & Grauman, 2017; Hu et al., 2017; Denton et al., 2017; Kim & Mnih, 2018). Currently, there are five metrics that are commonly used to evaluate the models: *BetaVAE* (Higgins et al., 2017), FactorVAE (Kim & Mnih, 2018), DCI (Eastwood & Williams, 2018), SAP (Kumar et al., 2017) and MIG (Chen et al., 2018). Interestingly, all of these metrics are based upon the definition of disentangled representation proposed in (Bengio et al., 2013). However, three of the metrics were designed to reflect Characteristic 1 of disentangled representations, while two were designed to reflect Characteristic 2. The primary goal of this paper has been to provide an analysis of the existing metrics of disentangled representations. We theoretically analyze how well the proposed metrics reflect the characteristics of disentangled representations that they are intended to reflect. In particular, we analyzed each of the existing metrics of disentanglement by two properties: whether a metric is close to 1 when a representation satisfies the characteristic that the metric reflects and whether the metric is close to 0 when a representation does not satisfy the characteristic. Surprisingly, we found that most of the existing metrics does not satisfy these basic properties.

The importance of developing a reliable metric of disentanglement has been clearly stated by Kim & Mnih (2018); Abdi et al. (2019). A key contribution of this paper is a new metric of disentangled representation, called DCIMIG. First, we formalize the desired characteristics, which, in our opinion, should reflect the metrics, and then prove that DCIMIG reflects them properly. In particular, DCIMIG captures the Disentanglement, Completeness and Informativeness of a representation using the Mutual Information Gap.

In future work, we plan to extend DCIMIG to the case where some generative factors form a subspace, and a disentangled representation should align with these subspaces instead of single generative factors.

References

Abdi, A. H., Abolmaesumi, P., and Fels, S. A preliminary study of disentanglement with insights on the inadequacy of metrics. *arXiv preprint arXiv:1911.11791*, 2019.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8):1798–1828, 2013.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Chen, T. Q., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

Denton, E. L. et al. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pp. 4414–4423, 2017.

Detlefsen, N. S. and Hauberg, S. Explicit disentanglement of appearance and perspective in generative models. *arXiv preprint arXiv:1906.11881*, 2019.

Dezfouli, A., Ashtiani, H., Ghattas, O., Nock, R., Dayan, P., and Ong, C. S. Disentangled behavioral representations. *bioRxiv*, pp. 658252, 2019.

Eastwood, C. and Williams, C. K. A framework for the quantitative evaluation of disentangled representations. In *ICLR*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.

Hu, Q., Szabó, A., Portenier, T., Zwicker, M., and Favaro, P. Disentangling factors of variation by mixing them. *arXiv preprint arXiv:1711.07410*, 2017.

Huang, D.-A. and Frank Wang, Y.-C. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2496–2503, 2013.

Kim, H. and Mnih, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pp. 951–958, 2009.

Locatello, F., Bauer, S., Lucic, M., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

Lorenz, D., Bereska, L., Milbich, T., and Ommer, B. Unsupervised part-based disentangling of object shape and appearance. *arXiv preprint arXiv:1903.06946*, 2019.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Pan, S. J., Yang, Q., et al. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Reed, S. E., Zhang, Y., Zhang, Y., and Lee, H. Deep visual analogy-making. In *Advances in neural information processing systems*, pp. 1252–1260, 2015.

Romera-Paredes, B. and Torr, P. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pp. 2152–2161, 2015.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tommasi, T., Orabona, F., and Caputo, B. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pp. 3081–3088, 2010.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103. ACM, 2008.

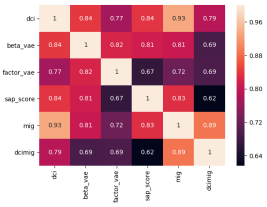
Yu, A. and Grauman, K. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5571–5580. IEEE, 2017.

Supplement to "Evaluating Disentangled Representations"

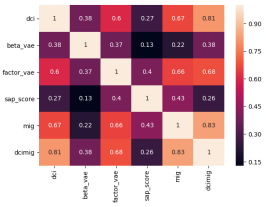
A. Experiments

A.1. Spearman rank correlation between metrics

We expand the tables given in (Locatello et al., 2018), which show the correlation of Spearman ranks between different metrics, by adding DCIMIG. We show the results for two datasets: dSprites (Higgins et al., 2017) and Cars3D (Reed et al., 2015), in Figure 2.



(a) Rank correlation of different metrics on the dSprites dataset.



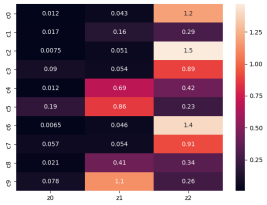
(b) Rank correlation of different metrics on the Cars3D dataset.

Figure 2. Rank correlation of different metrics on two datasets. Overall, all metrics are strongly correlated.

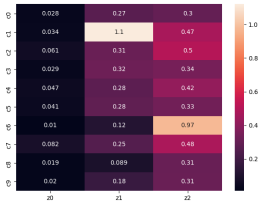
A.2. Different behavior of DCIMIG, MIG and DCI

A.2.1. Comparison of DCIMIG and MIG

The representation with the matrix of informativeness given in Figure 3b achieves a higher MIG score than the representation with the matrix of informativeness given in Figure 3a. This behavior of MIG can be explained by the fact that in the matrix of informativeness shown in Figure 3b only one latent factor, namely c_1, captures z_1, and only one latent factor, namely c_6 captures z_2. On the other hand, in the matrix of informativeness shown in Figure 3a, there are several latent factors, namely c_9, c_5, that capture z_1, and also there are several hidden factors that capture z_2.



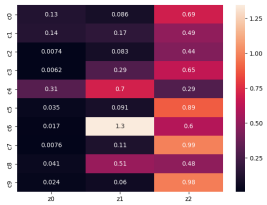
(a) Matrix of Informativeness, with MIG $= 0.05$, DCIMIG $= 0.233$.



(b) Matrix of Informativeness, with MIG $= 0.12$, DCIMIG $= 0.15$.

Figure 3. Matrices of informativeness of two representations, for which MIG and DCIMIG, do not agree which of the two is the most disentangled one.

(a) Matrix of Informativeness, with DCI $= 0.38$, DCIMIG $= 0.28$.



(b) Matrix of Informativeness, with DCI $= 0.42$, DCIMIG $= 0.17$.

Figure 4. Matrices of informativeness of two representations, for which DCI and DCIMIG, do not agree which of the two is the most disentangled one.

Now let us explain why DCIMIG gives lower scores for the representation with the matrix of informativeness shown in Figure 3b than for a representation with the matrix of informativeness shown in Figure 3a. DCIMIG selects c_1 to reflect z_1 and c_6 to reflect z_2 for the representation with the matrix of informativeness shown in Figure 3b. DCIMIG selects c_9 to reflect z_1 and c_2 to reflect z_2 for the representation with the matrix of informativeness shown in Figure 3a. But c_1 from Figure 3b is less disentangled than c_9 from 3a: c_1 from Figure 3b captures both z_1 and z_2. In addition, c_6 from Figure 3b is less disentangled than c_2 from Figure 3a. This explains why DCIMIG selects the representation with the matrix of informativeness specified in Figure 3a as the more disentangled representation.

A.2.2. Comparison of DCIMIG and DCI

The representation with the matrix of informativeness given in Figure 4b achieves a higher DCI score than the representation with the matrix of informativeness given in Figure 4a. This behavior of DCI can be explained by the fact that in the matrix of informativeness shown in Figure 4b there are only two entangled latent factors, namely c_1, c_8, while in Figure 4a four latent factors are entangled (c_0, c_5, c_7, c_8). It is worth noting that in the representation with the matrix of informativeness shown in Figure 4b, there are many latent factors that capture the same generative factors.

Now let us explain why DCIMIG gives lower scores for the representation with the matrix of informativeness shown in Figure 4b than for the representation with the matrix of informativeness given in Figure 4a. For the representation with the matrix of informativeness given in Figure 4b, DCIMIG selects c_6 to reflect z_1 and c_9 to reflect z_2. For the representation with the matrix of informativeness given in Figure 4a, DCIMIG selects c_9 to reflect z_1 and c_4 to reflect z_2. But c_6 from the Figure 4b is less disentangled than c_9 from Figure 4a, c_9 from Figure 4b is less disentangled than c_4 from Figure 4a. This explains why DCIMIG selects the representation with the matrix of informativeness specified

in Figure 4a as the more disentangled representation.